International Journal of Applied Engineering Research ISSN 0973-4562 Volume 12, Number 15 (2017) pp. 4886-4892
© Research India Publications. http://www.ripublication.com

MAC Protocol for Reducing Control Packet Overhead in Underwater Acoustic Networks

Seongjin Han[1] **and Sunmyeng Kim**[2]

[1,2] Department of Computer Software Engineering, Kumoh National Institute of Technology,
61 Daehak-ro, Gumi, Gyeongbuk 39177, South Korea.

[2]Orcid: 0000-0001-6076-6401

Abstract

Underwater acoustic networks (UANs) have become a very active research area in recent years. Compared with wireless networks, UANs are characterized by the limited bandwidth, long propagation delay and high channel dynamic in acoustic modems, which pose challenges to the design of medium access control (MAC) protocol. A reservation based MAC protocol called NR-MAC was proposed. In the NR-MAC protocol, source and destination nodes send control packets to all their neighbor nodes to make a channel reservation. The exchange of control packets is time-consuming, resulting in large overhead and low network performance. In this paper, we propose a new underwater MAC protocol, named control packet overhead reduction (CPOR), to address the above mentioned problems. The CPOR protocol reduces the time taken for control packet transmissions by sending a reservation packet for several neighbor nodes instead of each neighbour node. Performance evaluation is conducted using simulation, and confirms that the proposed protocol significantly outperforms the previous protocol in terms of number of control packets transmitted, throughput and delay.

Keyword: Control packet, MAC, Propagation delay, Overhead reduction, UAN.

INTRODUCTION

Underwater acoustic networks (UANs) 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]. And, MAC protocols for wireless networks cannot be directly applied to UANs 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]. 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 [5]. 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 [6]. 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 [7], 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 UANs. 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.

When the handshaking protocols get the channel access right through the handshaking procedure, they cause the low channel utilization due to the presence of long propagation delays in UANs, which in turn severely affects network performance.

To overcome the problem, reservation MAC protocols were proposed [8], [9], [10]. The R-MAC protocol schedules the transmission of control packets and data packets at both the source and the destination nodes to avoid data packet collisions completely [8]. When a source node has a data packet to send, it then schedules to send a reservation control packet to a destination node. The destination node schedules the transmission of acknowledgement packet for the reservation packet. And then, it sends acknowledgement packets to its neighbor nodes as well as the source node. If a neighbor node receives the acknowledgement packet, then it knows the duration of the subsequent data transmission and keeps silent during the time period. In R-MAC protocol, synchronization among nodes are required to schedule the transmitting and receiving time. The synchronization fails when the propagation delay is varied. The synchronization requirements are loosed by employing an asynchronous reservation scheme [9]. Before each transmission, control packets are used to re-arrange the schedule of reservation in each node. The R-MAC protocol also has channel resources waste problem since control packets sent by source nodes may not be received by their neighbors. In order to overcome this drawback, an improved R-MAC based MAC protocol called NR-MAC was proposed [10]. The difference with the R-MAC protocol is that a source node sends reservation control packets to its neighbor nodes as well as the destination node.

In NR-MAC protocol, source and destination nodes send reservation control packets to all their neighbor nodes to make a channel reservation. The exchange of control packets is time-consuming, resulting in large overhead and low network performance. In this paper, we propose a new underwater MAC protocol, named control packet overhead reduction (CPOR), to address the above mentioned problems. The CPOR protocol reduces the time taken for control packet transmission by sending a reservation packet for several neighbor nodes instead of each neighbor node.

The paper is organized as follows. In Section II, we provide a brief introduction to the NR-MAC protocol. We illustrate the proposed CPOR protocol in detail in Section III. In Section IV, performance studies are carried out through simulation results. Finally, we draw conclusions in Section V.

NR-MAC PROTOCOL

In this section, we summarize the NR-MAC protocol proposed in [10] and then present the motivation for our work by identifying the limitations of the NR-MAC protocol.

The NR-MAC protocol consists of three phases: latency detection, period announcement, and periodic operation. The first two phases are used to synchronize nodes in the neighborhood and the third one is for listen/sleep operations. A node in the latency detection phase detects the propagation delay to all its neighbors. In the period announcement phase, each node randomly selects its own listen/sleep time and broadcasts this time. In the periodic operation phase, nodes communicate through *xREV/REV/ACK-REV/DATA/ACK-DATA* control packet exchange. *xREV* and *REV* are reservation packets sent by a source node to its neighbor nodes and a destination node, respectively. *ACK-REV* is an acknowledgment packet sent by the destination node to its neighbor nodes and the source node. *ACK-DATA* is an acknowledgment packet for *DATA* packet.

The listen time is divided into three time slots. The first slot is reserved for receiving *ACK-REV*s, called R-slot. The second one is used to receive *xREV*s called S-slot. The third one is used to collect *REV*s and *ACK-DATA*s, called C-slot.

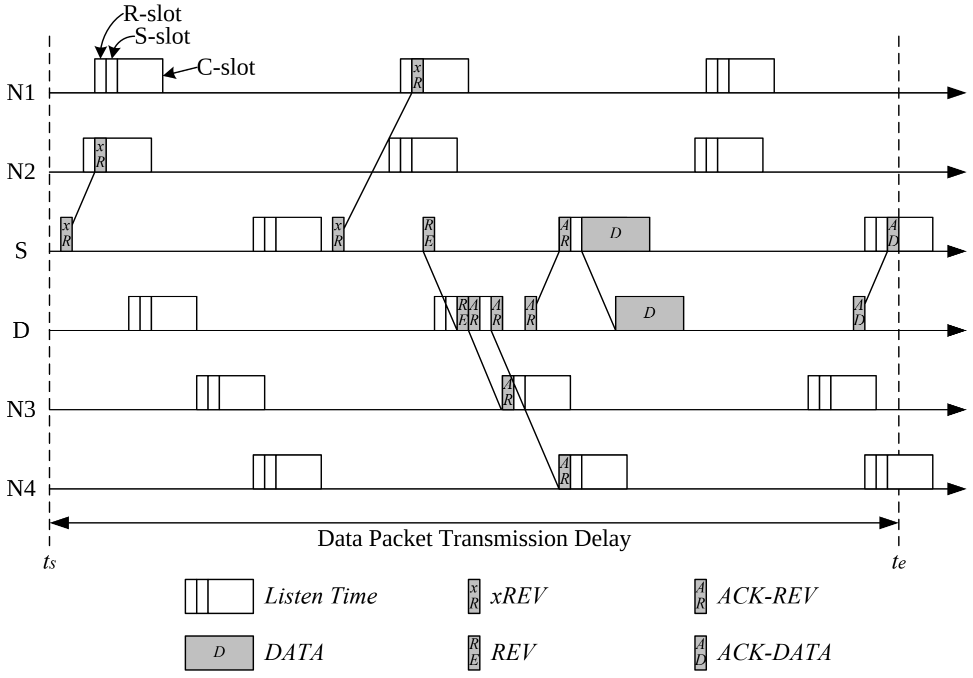

Figure 1: Example of data transmission in the NR-MAC protocol

Fig. 1 shows an example of a data transmission in the NR-MAC protocol. The source node (S) has a data packet to send at time *ts*. It sends *xREV*s to its neighbor nodess (N1 and N2) in their R-slots and a *REV* packet to the destination node (D) in its C-slot to make a channel reservation. If node D is able to arrange this reservation, it first reserves a time slot for the data transmission. Then it sends *ACK-REV*s to its neighbors (N3 and N4) in their R-slots. Upon receiving *ACK-REV*s, nodes N3 and N4 keep silent. It also sends an *ACK-REV* to node S in its R-slot. After receiving the *ACK-REV* packet, node S transmits a data packet. Finally, node D sends an *ACK-DATA* packet in C-slot of node S. The data transmission of node S ends at time *te*.

In NR-MAC protocol, source and destination nodes send reservation control packets to all their neighbour nodes to make a channel reservation. The exchange of control packets is time-consuming, resulting in large overhead and low network performance.

CPOR PROTOCOL

We describe the CPOR protocol proposed to reduce the control packet overhead and improve the network performance. The proposed CPOR protocol is similar to the NR-MAC protocol. Each node works in listen and sleep modes periodically. The durations for listen and sleep are the same for all nodes. And each node randomly selects its own listen time. Fig. 2 shows a structure of period.

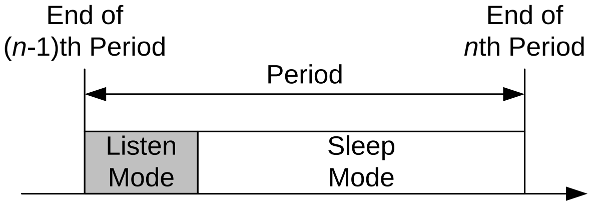

Figure 2: Structure of period

The CPOR protocol consists of three phases: latency detection, period announcement, and periodic operation. In the latency detection phase, a node detects the propagation delays to all its neighbors (see [8], [10] for more detail information).

In the period announcement phase, each node calculates listen time differences (*LTD*s) between itself and its neighbor nodes. The time difference indicates when the listen mode of a neighbor node starts based on the start time of its listen mode. In another words, if a node sends a packet at (the start time of its listen mode + its neighbor node's *LTD*), the neighbor node is in its listen mode and can receive the packet. We consider propagation delay to calculate the differences. Fig. 3 shows the definition of the *LTD*.

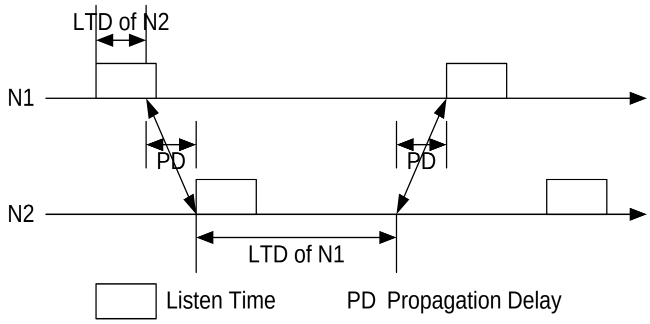

Figure 3: Definition of LTD

To calculate *LTD*s, each node randomly selects its own start time of the listen mode and broadcasts this time by sending a synchronization (SYN) packet. This packet has two fields: ID and time interval. ID is MAC address of a sender node. Time interval specifies the interval from the moment a SYN packet is sent by a sender node to the start time of its next listen mode (I_{TX}). After receiving the SYN packet, a receiver node calculates the time interval from the receiving time of the SYN packet to the start time of its next listen mode (I_{RX}). And then, the receiver node calculates *LTD* as follows:

$$LTD = (PERIOD - I_{RX} + I_{TX} - 2 \times PD)\%PERIOD \tag{1}$$

where, *PERIOD* and *PD* are a time interval of a period and a propagation delay between two nodes, respectively. *%* is a remainder operator.

Fig. 4 shows an example of *LTD*. There are two nodes (N1 and N2). Node N1 sends a SYN packet including its own ID (N1) and the time interval (I_{TX}). After receiving the SYN packet, node N2 calculates the time interval (I_{RX}). And then, it obtains the *LTD* of N1 by using (1).

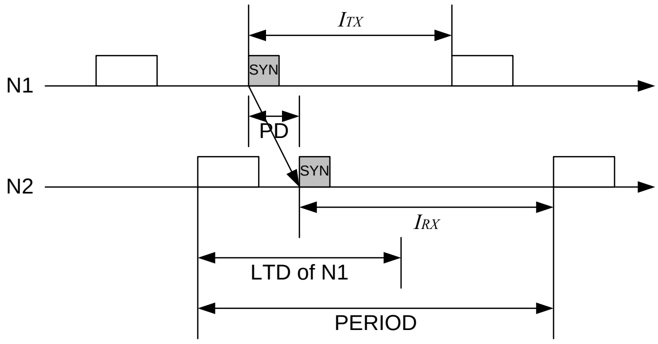

Figure 4: Example of LTD

In the periodic operation phase, nodes communicate through *xREV/REV/ACK-REV/DATA/ACK-DATA* control packet exchange. *xREV* and *REV* are reservation packets sent by a source node to its neighbor nodes and a destination node, respectively. *ACK-REV* is an acknowledgment packet sent by the destination node to its neighbor nodes and the source node. *ACK-DATA* is an acknowledgment packet for *DATA* packet.

The NR-MAC protocol sends one control packet to each neighbor node to make a channel reservation. Therefore, it consumes too much time for the exchange of control packets.

In order to reduce the time taken for control packet exchange, the proposed CPOR protocol sends the reduced number of control packets to all neighbor nodes. In another words, several neighbor nodes share one control packet.

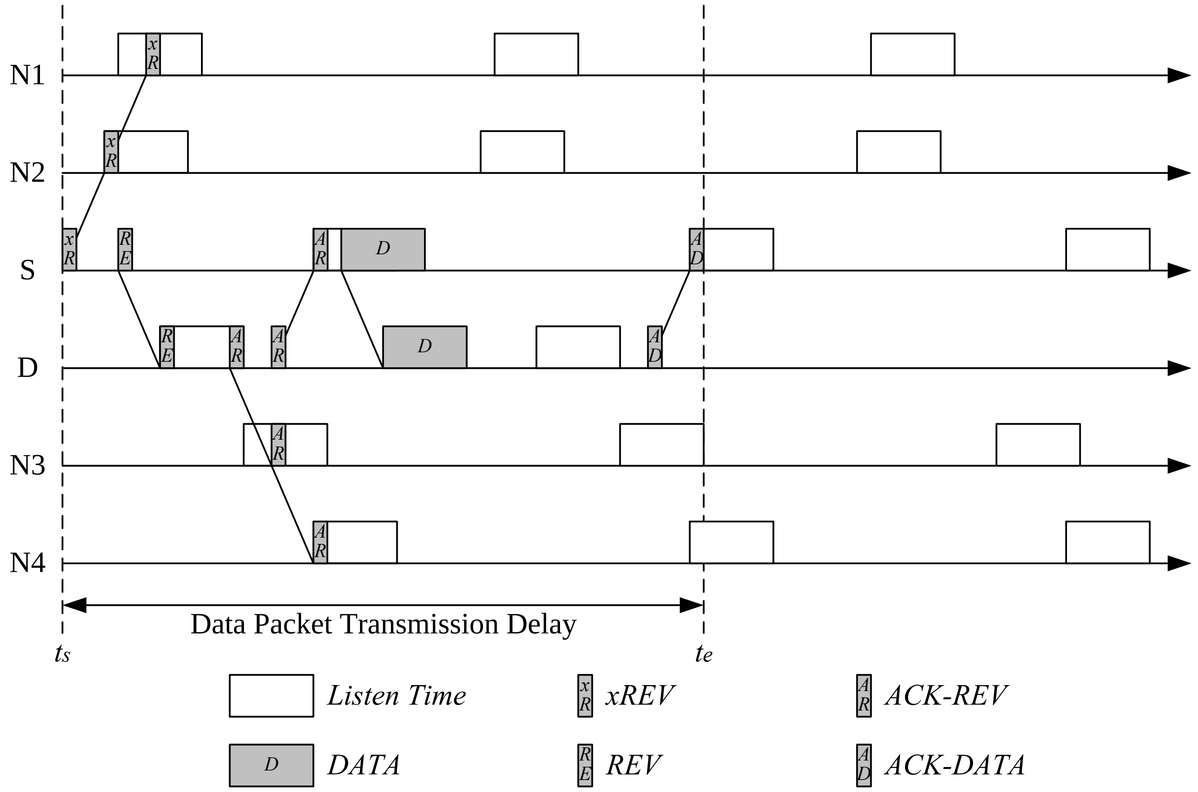

Figure 5: Example of data transmission in the CPOR protocol

Fig. 5 shows an example of a data transmission in the CPOR protocol. The source node (S) has a data packet to send at time *ts*. It sends one *xREV* packet to its neighbor nodes (N1 and N2) and a *REV* packet to the destination node (D) to make a channel reservation. If node D is able to arrange the reservation, it first reserves a time slot for this data transmission. Then it sends one *ACK-REV* packet to its neighbor nodes (N3 and N4). Upon receiving *ACK-REV*s, nodes N3 and N4 keep silent. It also sends an *ACK-REV* to node S. After receiving the *ACK-REV* packet, node S transmits a data packet. Finally, node D sends an *ACK-DATA* packet. The data transmission of node S ends at time *te*. As shown in Fig. 5, the CPOR protocol sends less control packets than the NR-MAC protocol. Therefore, the data packet transmission delay (*te* - *ts*) of the CPOR protocol is shorter than that of the NR-MAC protocol in Fig. 1.

In the CPOR protocol, we design scheduling algorithms for *xREV* and *ACK-REV* packets to reduce the control packet overhead for better use of channel resources.

Here, we describe the scheduling algorithm for *xREV* packets. When a source node has a data packet to send, it should make sure that *xREV* packets arrive in the listen times of all its neighbor nodes. To do this, the source node schedules the transmissions of *xREV* packets to its neighbor nodes. We consider Fig. 6 to explain the algorithm. There are one source node (S) and its neighbor nodes (N1 ~ N6). In the figure, a rectangle means the listen time of node *i*. The node S has a data packet to send at time t_o.

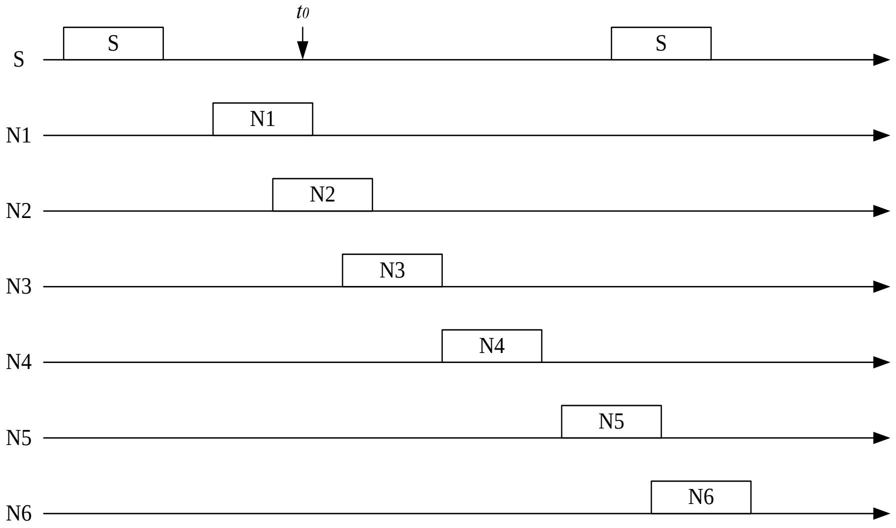

Figure 6: Topology for explaining the scheduling algorithm

The scheduling algorithm consists of four steps (see Fig. 7). In the step 1, node S maps the listen times of its neighbor nodes onto its own time line by using *LTD*s obtained in (1). In the step 2, if the listen time of node *i* meets the following criterion, node S remaps the listen time based on its next listen time.

$$L_{end,i} - D_{xREV} \le t_0 \qquad (2)$$

where, $L_{end,i}$ is the end time of listen mode for node *i*, D_{xREV} is the duration of an *xREV* packet. For example, the end time of listen mode for N1 is smaller than t_o, node S moves the listen time for N1 after its next listen mode. Even though the end time of listen time for N2 is larger than t_o, available listen time is not enough to receive an *xREV* packet. Therefore, node S also moves it.

In the step 3, node S obtains intersections of the listen times. Diagonal stripes represent indicate intersections. If a node has two intersections, node S selects the first one (e.g., N4). If the time duration of an intersection is smaller than D_{xREV} (e.g., an intersection of N4 and N5), node S ignores it. In the step 4, node S decides when to send *xREV* packets. And then, node S sends three *xREV* packets at time t_1 for N3 and N4, at time t_2 for N5 and N6, and at time t_3 for N1 and N2, respectively. The NR-MAC protocol sends six *xREV* packets for this example.

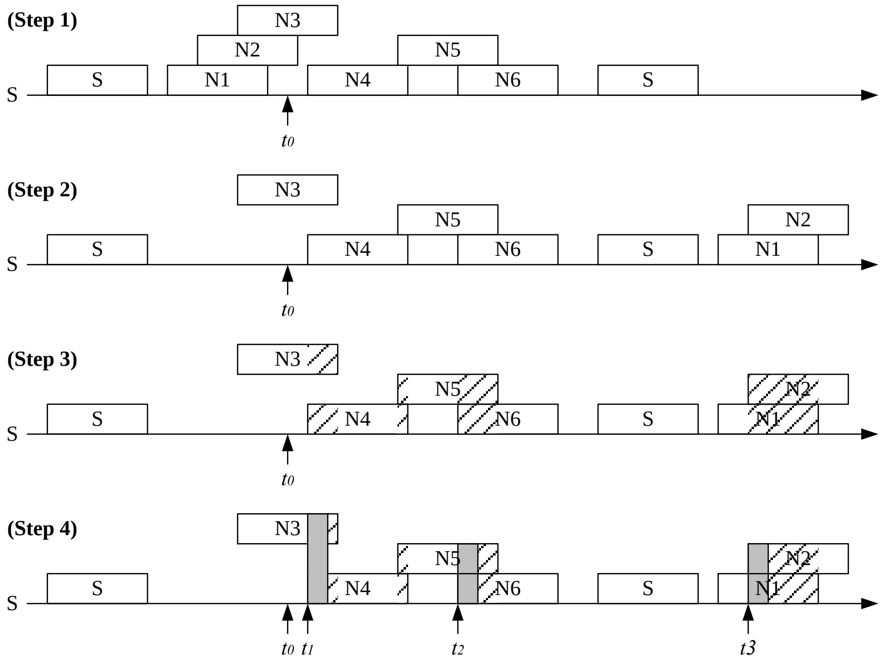

Figure 7: Example of scheduling *xREV* packets

The destination node starts its scheduling algorithm of *ACK-REV* packets after receiving a *REV* packet from the source node. This algorithm is the same as that of *xREV* packets.

SIMULATION RESULTS

In this section, we discuss the simulation results of the proposed CPOR protocol. To study the performance of the CPOR protocol, we implemented the protocol in C++. We compared the results to the results of the NR-MAC protocol. The parameters used in the simulation are listed in Table I. In the simulation, data rate is 10kbps and sound speed is 1500m/s. Data packet size is 60 Bytes and control packet size such as *xREV*, *REV*, *ACK-REV*, and *ACK-DATA* is 5 Bytes.

Table I. Simulation parameters

Parameter	Value
Data Packet Size	60 Bytes
Control Packet Size	5 Bytes
Period Length	1000 ms
Listen Length	100 ms
Data Rate	10 Kbps
Transmission Range	500 m
Sound Speed	1,500 m/s

We consider the topology in Fig. 8. In the simulation topology, there are one source node (S) and one destination node (D). They have the same number of neighbor nodes. If we say there *n* neighbor nodes, then it means that the source and destination nodes have *n* neighbor nodes, respectively. Neighbor nodes are randomly distributed over the network. The distance between S and D is 200m.

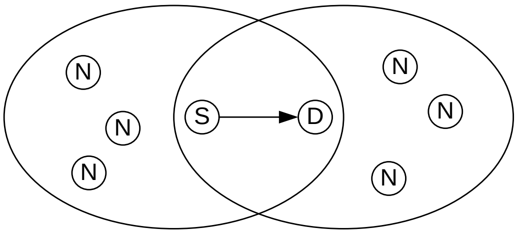

Figure 8: Simulation topology

Main performance metrics of interest are throughput, average delay, and number of control packets. Delay is the time elapsed from the moment a packet arrives at the queue header of the source node until the packet is successfully transmitted to the destination node, including the receipt of acknowledgement. The number of control packets is the sum of the control packets such as *xREV*, *REV*, *ACK-REV*, and *ACK-DATA* sent by the source and destination nodes to transmit one data packet.

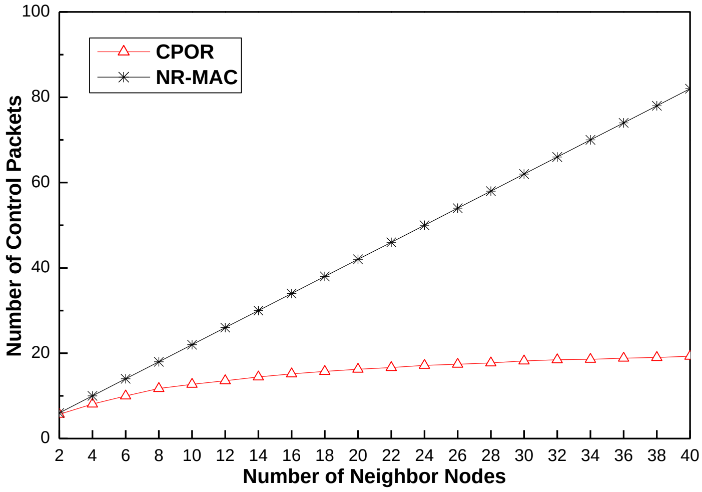

Figure 9: Number of control packets

Fig. 9 shows the number of control packets according to the number of neighbor nodes. From the figure, we can see that the proposed CPOR protocol always sends less number of control packets than the NR-MAC protocol. As the number of neighbor nodes increases, the difference becomes noticeable. In the NR-MAC protocol, the number of control packets linearly increases. This is because source and destination nodes sends one control packet to each neighbor node, respectively. Whereas, in the CPOR protocol, the number of control packets slowly increases. The CPOR protocol sends one control packet for several neighbor nodes.

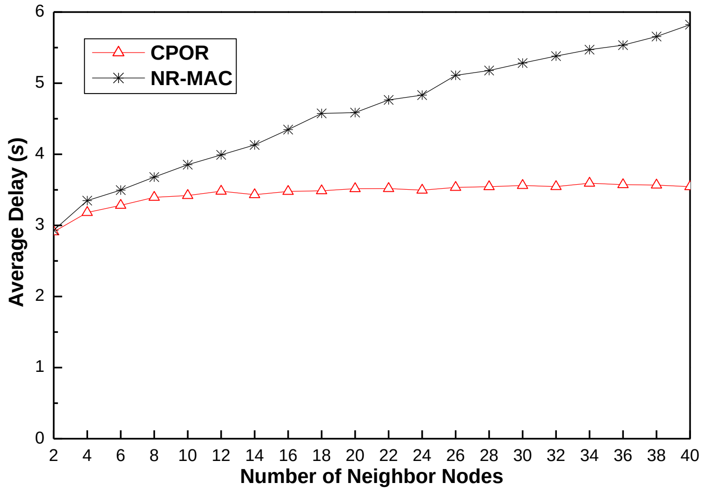

Figure 10: Average delay

Fig. 10 shows the average delay according to the number of neighbor nodes. The average delay is increasing due to control packet overhead. As shown in Fig. 9, as the number of neighbor nodes is increasing, the number of control packets is raised. The average delay is proportional to the number of neighbor nodes. For the proposed CPOR protocol, it increases very slowly. The proposed CPOR protocol is always superior to the NR-MAC protocol, regardless of the number of neighbor nodes.

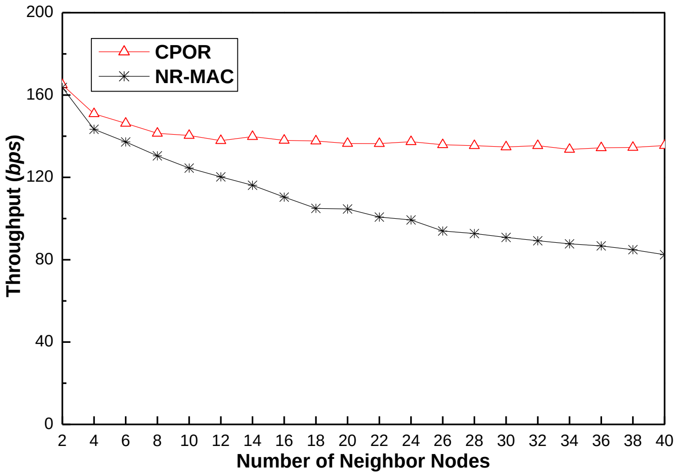

Figure 11: Throughput

Fig. 11 shows the throughput result according to the number of neighbor nodes. From the figure, we can see that the performance of the proposed CPOR protocol is always better than that of the NR-MAC protocol. Especially, the performance difference between them is getting higher as the number of neighbor nodes increases.

CONCLUSION

The handshaking protocols cause the low channel utilization due to the presence of long propagation delays. To overcome the problem, reservation MAC protocols were proposed. The NR-MAC protocol schedules the transmission of control packets and data packets at both the source and the destination nodes. This protocol sends control packets excessively, resulting in large overhead and low network performance. In this paper, we proposed the CPOR protocol to reduce the control packet overhead and improve network performance. The CPOR protocol reduces the time taken for control packet transmission by sending a reservation packet for several neighbor nodes instead of each neighbor node. Simulation result shows that the proposed protocol significantly outperforms the previous protocol.

ACKNOWLEDGEMENT

This research was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (2015R1D1A1A01059644).

REFERENCES

[1] P. Casari and M. Zorzi, “Protocol Design Issues in Underwater Acoustic Networks,” *Computer Communications*, vol. 34, no. 17, pp. 2013-2025, Nov. 2011.

[2] Y. Zhu, Z. Peng, J.-H. Cui, and H. Chen, “Toward Practical MAC Design for Underwater Acoustic Networks,” *IEEE Transactions on Mobile Computing*, vol. 14, no. 4, pp. 872-886, April 2015.

[3] 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*, vol. 56, pp. 47-59, Feb. 2015.

[4] 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*, to be published.

[5] 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*, vol. 38, no. 3, pp. 547-565, July 2013.

[6] 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*, vol. 13, no. 4, pp. 766-782, April 2014.

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

[8] P. Xie, and J. H. Cui, “R-MAC: An Energy Efficient MAC Protocol for Underwater Sensor Networks,” *International Conference on Wireless Algorithms Systems and Applications*, pp. 187-198, 2007.

[9] W. Zhai, J. Zhang, and S. H. Gupta, “Asynchronous Reservation MAC Protocol for Underwater Acoustic Sensor Network,” *International Conference on Signal Processing and Integrated Networks* (SPIN), pp. 136-145, 2014.

[10] M. Liu, W. Huang, L. Qian, and S. Zhang, “An Improved R-MAC based MAC Protocol for Underwater Acoustic Networks,” *IEEE International Conference on Signal Processing, Communications and Computing* (ICSPCC), 2016.