ISSN 2278-3091
Volume 8, No.4, July – August 2019
International Journal of Advanced Trends in Computer Science and Engineering
Available Online at http://www.warse.org/IJATCSE/static/pdf/file/ijatcse92842019.pdf
https://doi.org/10.30534/ijatcse/2019/92842019

MAC Protocol based on Double Cluster and Packet-Train for Underwater Acoustic Sensor Networks

Sunmyeng Kim
Department of Computer Software Engineering, Kumoh National Institute of Technology, South Korea
sunmyeng@kumoh.ac.kr

ABSTRACT

The underwater acoustic sensor networks (UASNs) are becoming increasingly important in ocean exploration. UASNs use acoustic waves since radio waves are extremely attenuated in underwater environments. UASNs are characterized by limited bandwidth and long propagation delays. These characteristics pose challenges to the design of medium access control (MAC) protocol. Especially, handshaking based MAC protocols have large delay in exchanging control and data packets. Also, they have higher probability of collision as the number of sensor nodes increases in networks. Therefore, they degrade network performance. In this paper, we propose a new MAC protocol based on double cluster and packet-train (DCPT) to improve the network performance. The double cluster reduces the channel contention among sensor nodes and lowers the collision probability. In the packet-train, multiple data packets are transmitted continuously in one handshaking. Thus, it reduces the time for control packet exchange. Performance evaluation is conducted using simulation, and confirms that the proposed protocol outperforms the previous protocol in terms of average delay.

Key words : Collision, Double Cluster, MAC, Packet-Train, UASN.

1. INTRODUCTION

Underwater acoustic sensor networks (UASNs) are a class of sensor networks deployed in underwater environments [1]. UASNs have attracted much attention in recent years due to their potential in various applications. UASNs consist of a surface buoy, on-shore data center and sensor nodes. Sensor nodes collect data and transmit them to the surface buoy. The surface buoy forwards them to the on-shore data center. The surface buoy includes an antenna for RF transmission.

Underwater communications are implemented using communication systems based on acoustic waves and electromagnetic (EM) waves [2]. EM waves are rapidly attenuated in seawater, seriously limiting the range of possible transmissions [3]. Given the difficulty of underwater communication via EM waves, acoustic waves have been widely adopted [4].

There are significant differences between UASNs and wireless networks because of the unique features such as low available bandwidth, long propagation delay, and dynamic channels in acoustic modems [5]. These features pose challenges to medium access control (MAC) protocol design [6-8]. 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) [9] cannot prevent packet collisions well among sensor nodes due to the long propagation delays in UASNs. 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 [7,8,10, 11]. Most of MAC protocols for USNs focus on the contention-based techniques since they facilitate an easy deployment on sensor nodes. Contention-based protocols are communication protocols that enable sensor nodes to use the same channel without pre-coordinating. Contention occurs when two or more sensor nodes attempt to access the channel at the same time. Contention causes packet collisions. Contention-based protocols 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 [12]. 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 [13]. In DOTS, each sensor node learns neighboring sensor 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

This research was supported by Kumoh National Institute of Technology (2019-104-046).

while reducing the collision probability. In Reference [14], 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 USNs. In the MACA-U protocol, a source sensor node transmits a RTS packet to a destination sensor node after channel contention. After receiving the RTS packet, the destination sensor node transmits a CTS packet. And then, the source sensor node transmits its own data packet to the destination sensor node. When other sensor nodes receive the RTS or CTS packets, they set their timer and do not participate in the data packet transmission process.

In traditional handshaking-based MAC protocols, the source node obtains channel access rights through a handshaking procedure and then sends its data packets to the destination node. Hence, the handshaking protocol causes low channel utilization due to the presence of long propagation delays in UASN. Also, it has higher probability of collision as the number of sensor nodes increases in networks. Therefore, they degrade network performance.

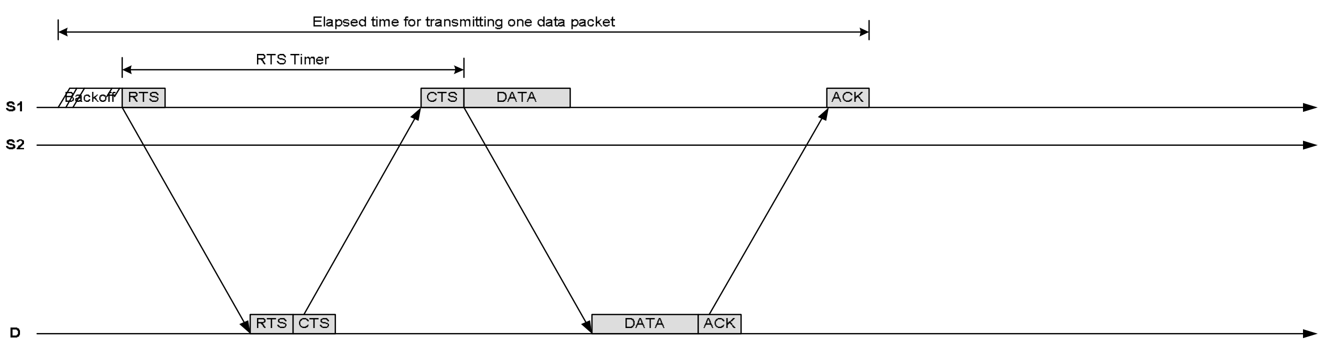
Cluster based MAC protocols have been proposed to reduce the energy consumption in order to prolong the network lifetime. The basic idea of clustering is to divide the sensing area into smaller segments that do not overlap one another. In a cluster, one of sensor nodes is selected as a cluster head (CH) and non-CH nodes become cluster members [15]. Cluster members transmit their data packets to the CH and the CH forwards them to the next CHs. Recently proposed clustering protocols include Optimal Clustering in Underwater [16], Distributed Underwater Clustering Scheme (DUCS) [17], and HydroCast [18]. When there are many cluster members in clusters, there are still high collision probability and low network performance.

In this paper, we propose a new MAC protocol based on double cluster and packet-train (DCPT) to improve the network performance. The double cluster reduces the channel contention among sensor nodes and lowers the collision probability. In the packet-train, multiple data packets are transmitted continuously in one handshaking. Thus, it reduces the time for control packet exchange.

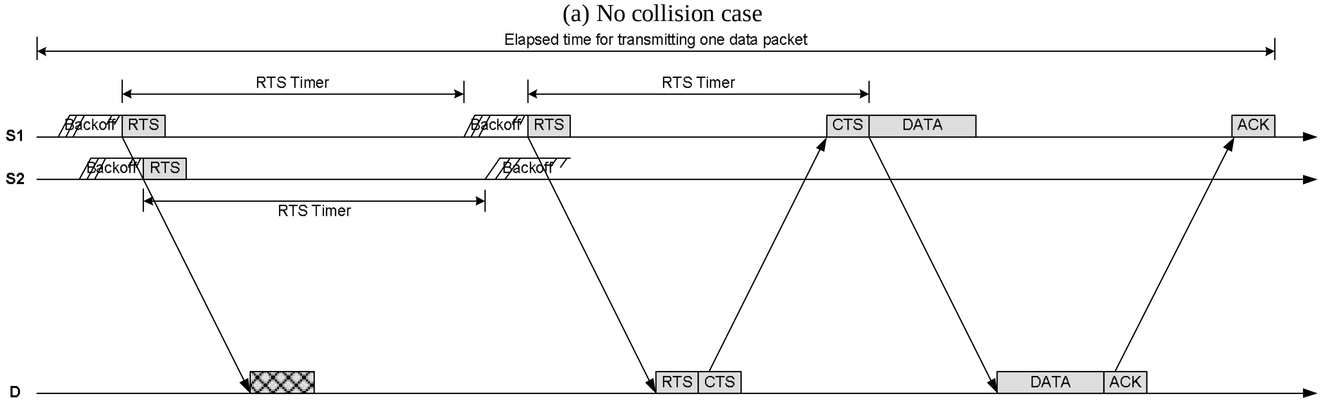
This paper is organized as follows. In section 2, we briefly describe the MACA-U with ACK protocol. In section 3, the proposed DCPT MAC protocol is presented in detail. In section 4, performance studies are carried out through simulation results. Finally, we draw conclusion in section 5.

2. MACA-U PROTOCOL

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



(a) No collision case



(b) Collision case

Figure 1: Example of MACA-U Protocol

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. Finally, the destination node sends an ACK packet to the source node.

Figure 1 shows an example of MACA-U protocol. There are three nodes: two source nodes (S1 and S2) and one destination node (D). In Figure 1(a), the node S1 has a data packet to send, it starts its backoff procedure. When its backoff counter reaches zero, it sets the RTS timer and sends a RTS packet to the destination node D. The node D sends a CTS packet after receiving the RTS packet successfully. The node S1 receives the CTS packet before its RTS timer expires. Then the node S1 turns off the RTS timer and transmits its data packet, and the node D sends an ACK packet. Even though, the node S1 transmits a data packet without collision, but channel waste occurs due to long propagation delay. In Figure 1(b), the source nodes S1 and S2 have data packets to send. After backoff procedure, the node S1 sends a RTS packet. The node S2 also sends a RTS packet because the RTS packet transmitted by the node S1 has not been reached until the backoff procedure of the node S2 is completed. They set their own RTS timer. The RTS packets from the nodes S1 and S2 are collided at the destination node D. The node S1 and S2 cannot receive the CTS packet from the destination node and their RTS timer expires. And then they perform the backoff process again. The backoff process of the node S1 is completed, and it transmits a RTS packet. The node S1 receives the RTS packet from the node S1 before its backoff process is completed, and the node S2 does not transmit the RTS packet. Therefore, no collision occurs at the destination node. Figure 1(b) wastes more channel time due to RTS packet collision than Figure 1(a).

3. PROPOSED DCPT MAC PROTOCOL

In the MACA-U protocol, all nodes participate in channel contention to send data packets to the destination node. Therefore, high collision probability and low network performance occur as the number of nodes increases.

In order to overcome the problem, sensor nodes form clusters and select cluster heads. The methods of cluster formation and header selection are beyond the scope of this paper. Please refer to the methods in previous clustering work.

Figure 2 shows the topology of the UASN after cluster formation. Each white circle represents a cluster member node and each black circle represents a cluster head. Surface buoy is placed on the water surface and communicate with the on-shore data center through radio frequency (RF) channel. It communicates with cluster heads through acoustic channels. Cluster communications are classified into inter-cluster and intra-cluster communications. In the intra-cluster communication, when a cluster member node has a data packet to send, it transmits the data packet to the cluster head within its cluster. In the inter-communication, each cluster head forwards data packets received from its member nodes to the next cluster head on the path to the surface buoy.

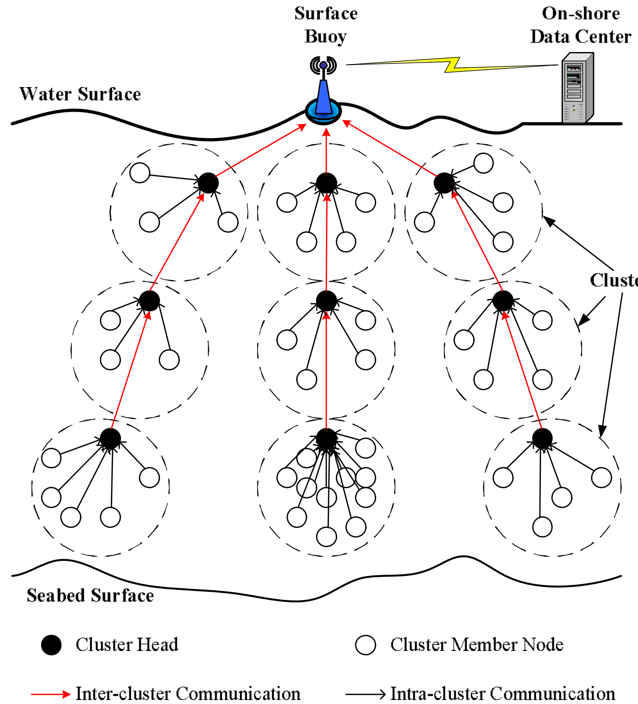


Figure 2: Topology after Cluster Formation

Some clusters may have a high density of member nodes. This means that some clusters include many cluster member nodes (see Figure 3). In such clusters, all the cluster member nodes participate in the channel contention to send data packets to their cluster head in the MACA-U protocol. This increases the collision probability and degrades network performance.

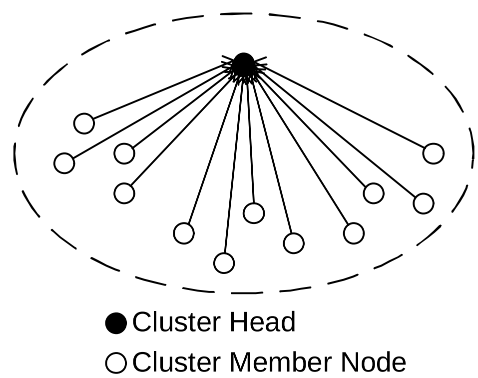


Figure 3: Example of High Density Cluster

In order to lower the density, the proposed protocol divides the high density cluster into smaller clusters. To distinguish between the two kinds of clusters, we use two cluster terms: network cluster and local cluster. Network clusters are clusters created for all nodes in a network (see Figure 2). Local clusters are small clusters created from dense network clusters (see Figure 4).

Local cluster member nodes select a local cluster head. The selected local cluster heads only participate in the channel contention in the proposed DCPT MAC protocol.

The proposed DCPT MAC protocol uses two kinds of clusters (double cluster) and has low collision probability.

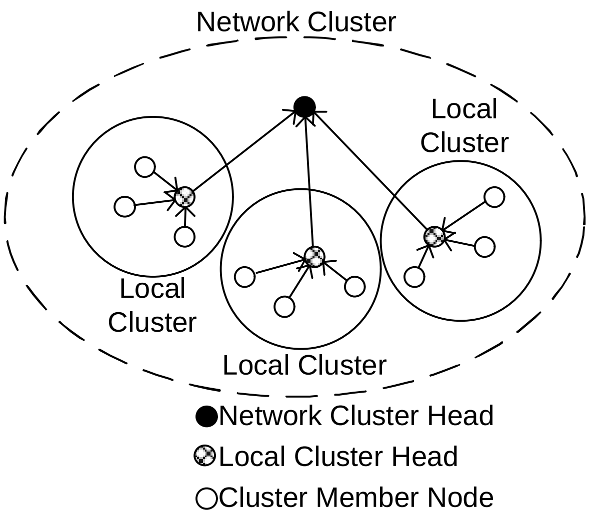


Figure 4: Example of Local Cluster

The DCPT MAC protocol consists of three phases: channel contention phase, data transmission phase, and data notification phase.

The local cluster head starts the channel contention phase when it or its member nodes have data packets to send. In the channel contention phase, the local cluster head performs backoff procedure. When the backoff procedure is completed, the local cluster head transmits an RTS packet to the network cluster head. The network cluster head responds with a CTS packet. Member nodes that have receive the RTS and/or CTS packets enter the data notification phase. In the data notification phase, member nodes with data packets to transmit send busy tone to their local cluster heads. The local cluster heads receiving the busy tones know that their member nodes have data packets to transmit, and start the data transmission phase.

When a local cluster member node receives an RTS and/or CTS, it starts the data notification phase. The behavior of the member node depends on which local cluster head sent the RTS packet. If the member node's local cluster head sent an RTS packet, the member node transmits a busy tone for its time slot (we will describe how to allocate time slots later). Otherwise, it sends a busy tone immediately after receiving the RTS packet. Also, when receiving a CTS packet, it transmits a busy tone immediately after receiving the CTS packet.

We use an adaptive transmission power control scheme. A member node sends a busy tone at low power to avoid inter-local cluster collisions.

Figure 5 shows the behavior of the local cluster member nodes in the data notification phase after receiving RTS and/or CTS packets. There is one network cluster including three local

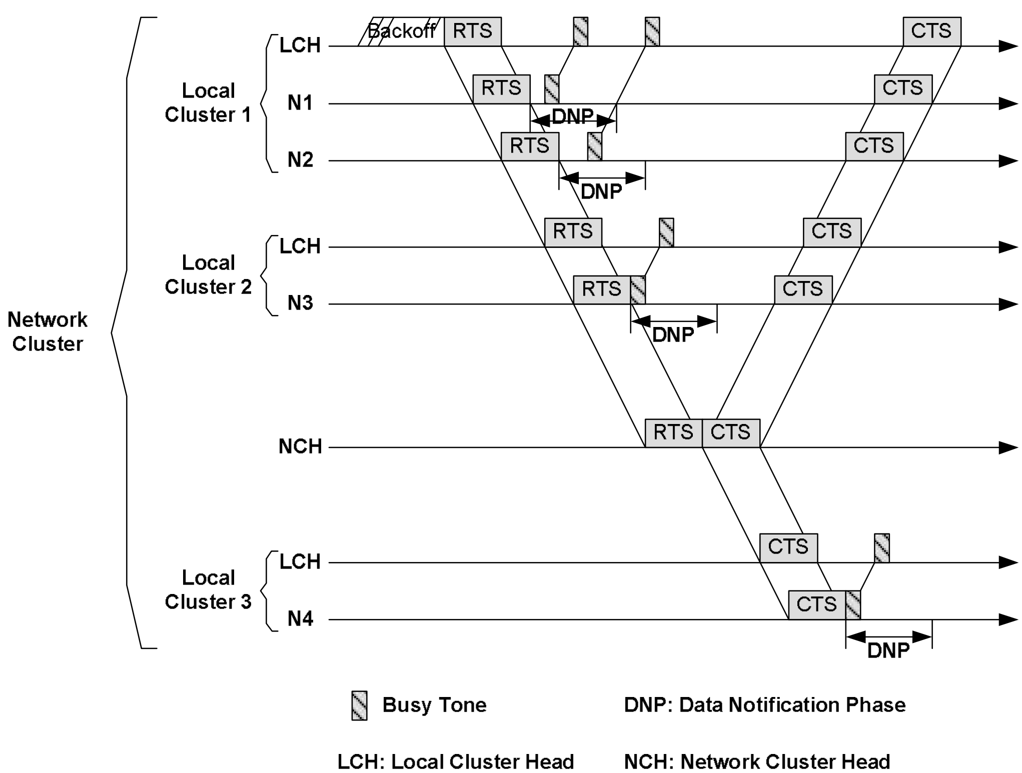


Figure 5: Member Node Behavior According to RTS/CTS Packet Reception

clusters. Nodes N1 ~ N4 are local cluster member nodes. We assume that the LCH in local cluster 1 knows that its member nodes have data packets to transmit. Therefore, it starts its backoff procedure. After completing the procedure, it transmits an RTS packet to the NCH. The nodes N1, N2 and N3 receiving the RTS packet enter the data notification phase. N1 and N2 send busy tones to their LCH during the time slot assigned to them. N3 sends a busy tone to its LCH immediately after receiving the RTS packet. After receiving the RTS packet, the NCH responds with a CTS packet. N4 sends a busy tone to its LCH immediately after receiving the CTS packet. Through the busy tones transmitted in the data notification phase, LCHs recognize that their member nodes have data packets to transmit. In particular, the LCH in local cluster 1 knows the IDs of the member nodes having data packets to transmit.

Here, we describe how to allocate time slots to member nodes. Time slot is different from that used in TDMA. In the proposed protocol, the time slot means a delay time indicating when a member node transmits a busy tone after receiving an RTS packet. The delay time is calculated based on propagation delay between the LCH and a member node. LCH knows propagation delay between itself and its member nodes. The LCH should make sure that busy tones from its member nodes arrive without any collisions. To do this, the LCH schedule the transmissions of busy tones of its member nodes. We consider Figure 6 to explain the algorithm. There are one LCH and five member nodes (N1 ~ N5). In the figure, a rectangle means a busy tone transmitted from member node *i*. First, the LCH calculates arrival times and end times of busy tones from its member nodes as following.

$$ArrTime(i) = PD(i) * 2 \quad (1)$$
$$EndTime(i) = ArrTime(i) + TxBusy \quad (2)$$

where, $ArrTime(i)$ and $EndTime(i)$ are the arrival time and end time of a member node *i*, respectively. $PD(i)$ and $TxBusy$ are propagation delay between LCH and member *i*, and the duration of a busy tone.

In the step 1, the LCH maps the arrival times of busy tones from its member nodes onto its own time line. If $EndTime(i\text{-}1)$ is smaller than or equal to $ArrTime(i)$, the busy tones collide. We can see that busy tones from member nodes N2 and N3, and N3 and N4 collide. In order to avoid collisions, the LCH calculates the defer time of the collided busy tones and moves them.

$$DefTime(i) = EndTime(i\text{-}1) - ArrTime(i) \quad (2)$$

where, $DefTime(i)$ is the defer time of member node *i*.

after calculating the defer time, the LCH updates the arrival time and end time of member node *i* as following.

$$ArrTime(i) = ArrTime(i) + DefTime(i) \quad (3)$$
$$EndTime(i) = EndTime(i) + DefTime(i) \quad (4)$$

In the step 2, the LCH calculates the defer time of member node N3 and then moves it. In the step 3, the LCH repeats this procedure for member node N4. Because of this, busy tones from N4 and N5 collide. Therefore, the LCH moves the busy tone for N5 in the step 4.

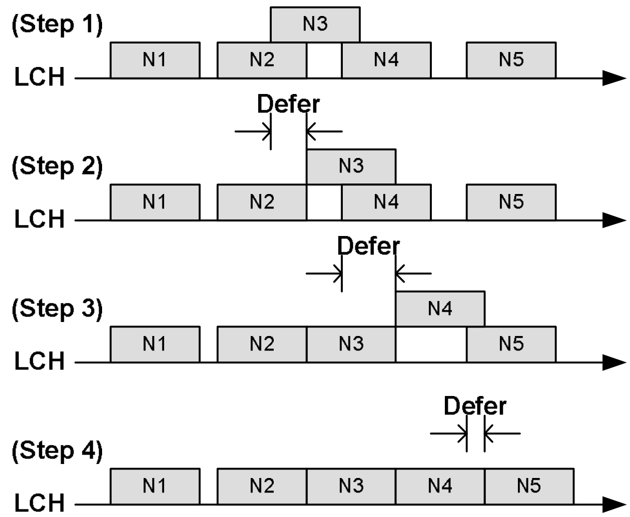


Figure 6: Example of Scheduling Busy Tones

After calculating the defer times, the LCH broadcasts them to its member nodes. The member nodes receive the RTS packet and transmit their busy tone after their defer time.

In an environment where data packets occur infrequently, member nodes hardly receive RTS and/or CTS packets. In this case, the member nodes do not enter the data notification phase. In this case, when the NO_COMM has elapsed after receiving the last communication packet, the member node with data packets enters the data notification phase and transmits the busy tone to its LCH. NO_COMM is calculated as follows;

$$NO\_COMM = PDmax * 2 + CWmax * SlotTime \quad (5)$$

where, $PDmax$ and $CWmax$ are the maximum propagation delay and contention window, respectively. $SlotTime$ is the duration of a slot time.

Figure 7 shows an example of busy tone transmission in an environment where there is no communication. There are three nodes. The NCH is the network cluster head and sends an ACK packet. The nodes N1 and N2 belong to different local clusters. They receive the ACK from the NCH. The node N1 generates a data packet before the NO_COMM elapses. It defers its busy tone transmission. When the time elapses, it transmits the busy tone to its LCH to start the channel contention phase. The N2 generates after the NO_COMM elapses. It transmits a busy tone immediately.

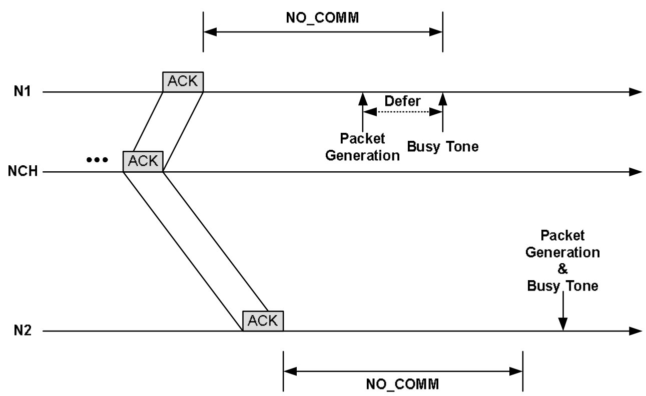


Figure 7: Example of Busy Tone Transmission

The data notification phase allows the LCH to know which member node has data packets. After receiving the CTS packet from the NCH, the LCH performs scheduling on the member nodes with data packets and informs the member nodes through the schedule (SCH) packet. The member nodes start the data transmission phase and transmit their data packets to the destination node at their scheduled times. Data packets from the member nodes are transmitted continuously in one handshaking. This is called packet-train.

Figure 8 shows the basic operation principle of the proposed DCPT MAC protocol. In the figure, there are two local clusters. Nodes N1 and N2 belong to the local cluster 1, node N3 belongs to the local cluster 2. The LCH in the local cluster 1 has a data packet and starts the channel contention phase. It transmits an RTS packet to the NCH after its backoff procedure is completed. Nodes N1, N2 and N3 enter the data notification phase and send busy tones to their LCHs. The NCH responds with a CTS packet after receiving the RTS packet. The LCH in the local cluster 1 enters the data transmission phase after receiving the CTS packet. It transmits a schedule (SCH) packet after performing scheduling of data packet transmission to itself and the node N1. And then, they transmit their data packets sequentially. Finally, the NCH sends an ACK packet.

4. SIMULATION RESULTS

In this section, we analyze simulation results of the proposed DCPT MAC protocol. To study the performance of the DCPT protocol, we actually implemented the protocol. Performance of the DCPT protocol is compared with that of the MACA-U protocol. We consider the topology in Figure 4 with one network cluster. Nodes are randomly distributed over the network cluster.

Main performance metric of interest is average delay. 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.

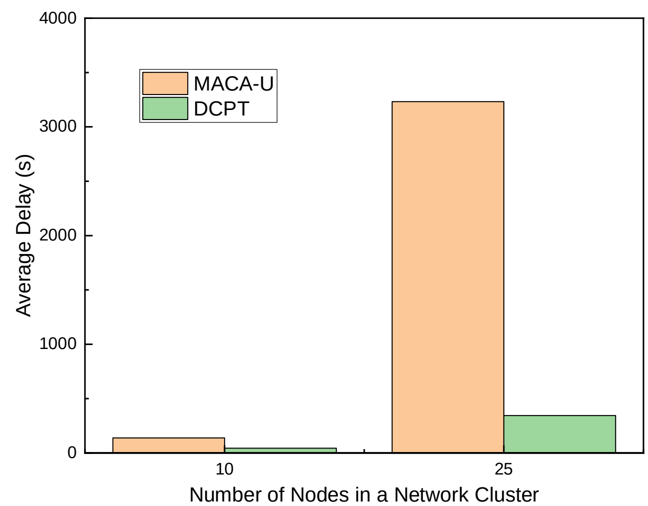


Figure 9: Average Delay according to the Number of Nodes in a Network Cluster

Figure 9 shows the results for the average delay according to the number of nodes in a network cluster. We assume that each local cluster consists of 5 member nodes. Therefore, 10

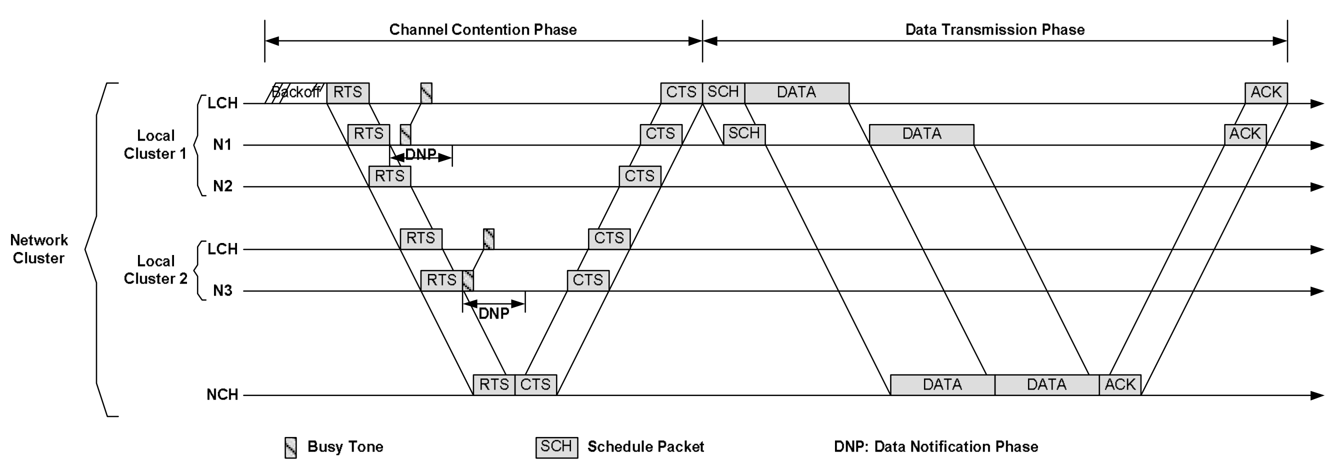


Figure 8: Operation of the Proposed DCPT MAC Protocol

nodes and 25 nodes are divided into 2 local clusters and 5 local clusters, respectively. From the figure, we can see that the proposed DCPT protocol always shows better performance than the MACA-U protocol. Especially, the delay for the DCPT protocol increases very slowly compared to that for the MACA-U protocol as the number of nodes is larger. In the DCPT protocol, the double cluster reduces the channel contention among nodes and lowers the collision probability. In the packet-train, multiple data packets are transmitted continuously in one handshaking. Thus, it reduces the delay.

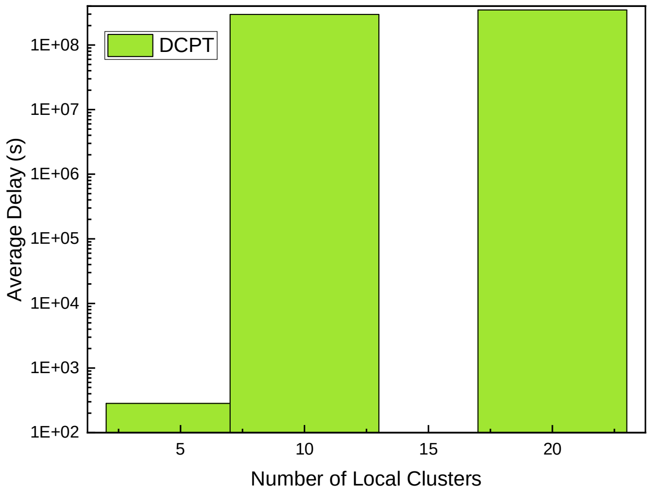


Figure 10: Average delay according to the Number of Local Clusters

Figure 10 shows the results for the average delay according to the number of local clusters in a network cluster. There are 100 member nodes in a network cluster. Therefore, 5, 10, 20 local clusters mean that each local cluster includes 20, 10, and 5 local cluster member nodes, respectively. As the number of local clusters increases, the delay is getting larger.

5. CONCLUSION

In the UASNs, handshaking based MAC protocols degrade the network performance due to the large delay in exchanging control and data packets. Also, as the number of nodes increases, the probability of collision increases and the performance decreases. In this paper, we proposed the DCPT MAC protocol to improve the network performance. The proposed DCPT MAC protocol uses double cluster and packet-train. The double cluster reduces the channel contention among nodes and lowers the collision probability. The packet-train reduces the delay. Therefore, the DCPT MAC protocol improves network performance.

REFERENCES

1. P. Casari and M. Zorzi. **Protocol design issues in underwater acoustic networks**, *Computer Communications*, vol. 34, no. 17, 2013-2025, 2011. https://doi.org/10.1016/j.comcom.2011.06.008
2. N. Saeed, A. Celik, and M.-S. Alouini. **Underwater optical wireless communications, networking, and localization: a survey**.
3. X. Che, I. Wells, G. Dickers, P. Kear, and X. Gong. **Re-evaluation of RF electromagnetic communication in underwater sensor networks**, *IEEE Communications Magazine*, vol. 48, no. 12, pp. 143-151, 2010. https://doi.org/10.1109/MCOM.2010.5673085
4. 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. https://doi.org/10.1109/MCOM.2009.4752682
5. K. M. Awan, P. A. Shah, K. Iqbal, S. Gillani, W. Ahmad, and Y. Nam. **Underwater wireless sensor networks: a review of recent issues and challenges**, *Wireless Communications and Mobile Computing*, vol. 2019, Jan. 2019. https://doi.org/10.1155/2019/6470359
6. N. Morozs, P. Mitchell, and Y. V. Zakharov. **TDA-MAC: TDMA without clock synchronization in underwater acoustic networks**, *IEEE Access*, vol. 6, pp. 1091-1108, 2017. https://doi.org/10.1109/ACCESS.2017.2777899
7. 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, 872-886, 2015. https://doi.org/10.1109/TMC.2014.2330299
8. 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, 2015. https://doi.org/10.1016/j.comcom.2014.09.006
9. K. Vallimadhavi, P. Sivaprasad, R. Baladinakar, Y.D.Siva Prasad and K.D.S. Naidu. **Automatic connection of various medical sensors by using WBAN adaptive routing protocol**, *International Journal of Advanced Trends in Computer Science and Engineering*, vol. 7, no. 6, pp. 136-139, 2018. https://doi.org/10.30534/ijatcse/2018/14762018
10. 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*, vol. 16, no. 11, pp. 4127-4136, 2015. https://doi.org/10.1109/JSEN.2015.2453552
11. S. Choi and S. Kim. **Multi-Channel MAC Protocol based on Dynamic Time Slot Allocation for Underwater Sensor Networks**, *International Journal of Advanced Trends in Computer Science and Engineering*, vol. 8, no. 2, pp. 187-193, 2019. https://doi.org/10.30534/ijatcse/2019/13822019
12. 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, 2013.

https://doi.org/10.1109/JOE.2012.2227553

13. 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, 2014. https://doi.org/10.1109/TMC.2013.2297703
14. 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, 2008.
15. K. Wang, H. Gao, X. Xu, J. Jiang, and D. Yue. **An energy-efficient reliable data transmission scheme for complex environmental monitoring in underwater acoustic sensor networks**, *IEEE Sensors*, vol. 16, no. 11, pp. 4051-4062, 2016. https://doi.org/10.1109/JSEN.2015.2428712
16. S. Yadav and V. Kumar. **Optimal clustering in underwater wireless sensor networks: acoustic, EM and FSO communication compliant technique**, *IEEE Access*, vol. 5, pp. 12761-12776, 2017. https://doi.org/10.1109/ACCESS.2017.2723506
17. M.C. Domingo and R. Prior. **A distributed clustering scheme for underwater wireless sensor networks**, *IEEE PIMRC 2007*, pp. 1-5, 2007. https://doi.org/10.1109/PIMRC.2007.4394038
18. U. Lee, P. Wang, Y. Noh, L.F.M. Vieira, M. Gerla, and J.-H. Cui. **Pressure routing for underwater sensor networks**, *IEEE INFOCOM*, pp. 1-9, 2010.