Applied Mathematics & Information Sciences
An International Journal

http://dx.doi.org/10.18576/amis/110223

MPT-MAC: Multi-Packet Transmission MAC Protocol in UWASNs

Dongwon Lee[1,*] *and Sunmyeng Kim*[2]

[1] Hanwha Coporation,264-36 Sanho-daero Gumi Kyungbuk, 730-904, Korea
[2] Department of Computer Software Engineering, Kumoh National Institute of Technology, Gumi, 730-701 Korea

Received: 22 June 2015, Revised: 21 Dec. 2016, Accepted: 23 Dec. 2016
Published online: 1 Mar. 2017

Abstract: Recently, there are lots of ongoing researches for underwater acoustic sensor networks (UWASNs). Unlike terrestrial wireless sensor networks which use radio waves, UWASNs communicate by using acoustic waves. The acoustic waves have long propagation delay. Therefore, media access control (MAC) for terrestrial wireless sensor networks does not operate appropriately. Most of recent researches for UWASNs are targeting simple network topology which consists of a single gateway and a number of nodes. However, there are mobile objects, such as autonomous underwater vehicles (AUVs) or multiple gateways, to guarantee reliability of the system in practical UWASNs. Therefore, data transmission with various routes from each node happens. Under this network environment, we propose a new MAC protocol, which can reduce frequent channel contentions among nodes arose by various transmission routes. A Sender transmits single RTS packet to multiple receivers, in order to reserve channel with them, and then it transmits data to them. Therefore, channel contention time for data transmission decreases and network performance improves. In this paper, we evaluate the proposed protocol, by comparing it to existing MAC for UWASNs through the simulation.

Keywords: UWASN, MAC, MPT-MAC, collision resolution, underwater

1 Introduction

Underwater acoustic sensor networks (UWASNs) can be used for research and development of ocean/underwater resource, so there are a lot of ongoing researches about UWASNs[1]. Researches about underwater communication, in the past, focused on long range communication between two nodes. In UWASNs, however, diverse information is gathered by using multiple nodes which are allocated nearby, and it is transmitted via various routes. Unlike a terrestrial wireless sensor network, UWASN communicates by using acoustic signals. The acoustic signal has about 1.5km/s speed, and this is much slower than terrestrial radio signals[2][3]. The protocols about wireless sensor network which were researched for terrestrial networks, are not available to use in UWASN, because acoustic signal has long latency and the available bandwidth is severely limited. To solve this problem, in recent years, a lot of new MAC protocols for UWASN have been proposed. In UWASN, a lot of sensor nodes and gateways are deployed in 3D space. The gateways transmit collected information from sensor nodes to the server on the ground. But it is impossible to collect sensing data from sensor nodes when there is destruction or loss of the gateway[3][4]. For this reason, network redundancy is needed in UWASN. Therefore, multiple gateways are commonly used in USASN topology to guarantee network robustness from data loss caused by absence of gateway. In multiple gateway system, various data transmission routes exist from each node. The situation cause bottleneck phenomenon of data flow. Hence, network performance of USASN is highly decreased. In addition, a similar situation occurs in Location-based Service and Tracking System for UWASN. The systems have very complex data flow. In this paper, we propose a new MAC protocol to improve network performance which consists of nodes having diverse direction for transmitting data. The proposed protocol is called MPT-MAC(Multi Packet Transmission MAC). In the MPT-MAC, each node measures propagation delay with neighbor nodes periodically and then saves the measured propagation delay value and neighbor address on its neighbor node list. In the proposed scheme, each node has multi queue for several destination nodes. When each

* Corresponding author e-mail: prolinuxer@gmail.com

node has data packets to transmit, it starts channel contention for transmission. After the channel contention, RTS packet is broadcasted to all destination nodes. RTS packet includes address of each destination node and CTS response time information which notices when each destination nodes should transmit its own CTS packet, in order to avoid collision of CTS packets transmitted from multiple destination nodes. After receiving the RTS packet from the sender, the destination nodes transmit CTS packets at their own CTS response time. The sender node transmits data packet after receiving CTS packet from each destination node without collisions. Similar to RTS packet, DATA packet includes information of ACK packet response time for each destination node. Each destination node transmits ACK packet at its own ACK packet response time after receiving data packet. The response time of CTS packet and ACK packet is calculated based on the measured propagation delay. With this technique, packet collision between nodes can be avoided, and data packet transmission to multiple nodes with only one contention is available. Therefore, the network performance can be improved by this technique. Organization of this paper is as following. Section 2 explains operation principle of proposed MPT-MAC. In Section 3, MPT-MAC protocol and existing protocol for underwater sensor network are compared and evaluated through simulation.

2 MPT-MAC

In the MPT-MAC protocol, each node measures propagation delay with its neighbor nodes, and then uses it to schedule CTS and ACK response times. Generally, the propagation delay is calculated by using RTT(Round Trip Time)[5]. Thus, measurement scheme of RTT is very important.

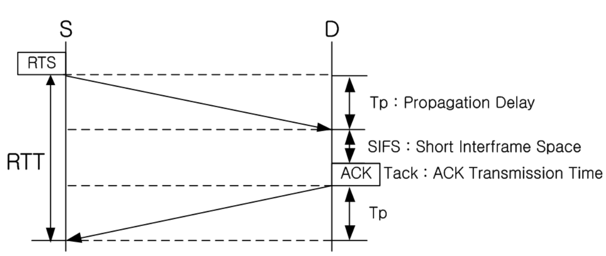


Fig. 1: Example of Round Trip Time Measurement

In Fig. 1, node S can get RTT value from node D. The RTT consists of 2 time of propagation, SIFS time and ACK packet transmission time. So, node S can get the propagation delay using this value easily. In this paper, we use similar scheme for RTT measurement (see Fig. 2). Each node maintains a neighbor node list, which contains the following information: neighbor node address and its propagation delay. If the neighbor node list is empty, a node broadcasts PDE (Propagation Delay Estimation) packet. The neighbor nodes which receive the PDE packet randomly select a time slot and transmit ACK packet, like Slotted Aloha[6]. After receiving ACK packets, the node stores the addresses and delay of neighbor nodes to its own neighbor node list and goes into IDLE state. In case of no receiving ACK packets from neighbor nodes, the node periodically transmits a PDE packet. The node overhears transmission packets of neighbor nodes in IDLE state. If the node receives a packet from a node not on its neighbor node list, it registers the address information of the node on the list. Each node periodically checks its neighbor node list. If there is no information of propagation delay to neighbor nodes, it transmits a PDE packet to measure propagation delay. At this moment, a PDE packet is not broadcasted, but transmitted using unicast to the nodes. A neighbor node which receives the PDE packet transmits ACK packet and becomes IDLE state. Once ACK packet is received, address and propagation delay are added on the neighbor node list. When a new node is found by packet overhearing, the process of transmitting PDE packet is repeated to measure the delay. In this way, each node in the network obtains propagation delay information between neighbor nodes.

In the proposed MAC protocol, each node has the same number of transmission queues as the number of neighbor nodes. When there are data packets to transmit in the queues, the node starts channel contention by using back-off algorithm. Back-off is a mechanism used to avoid collisions in mobile ad hoc networks.

Collision is avoided by requiring the node to wait for a time called back-off time before trying to access the channel after a transmission failure[7]. When the back-off counter becomes zero, the node broadcasts an RTS packet to destination nodes in order to reserve channel with them. The RTS packet contains addresses and CTS response times of the destination nodes. Fig. 3 shows an example of RTS-CTS transmission.

In Fig. 3(a), the distance between nodes B, C, and D is far enough, so no collision occurs even CTS packet is immediately transmitted after receiving RTS packet. Therefore, in this case, it is not necessary to schedule CTS packet response time. However, in case of short distance between nodes B, C, and D as shown in Fig. 3(b), if CTS packet is immediately transmitted after receiving RTS packet, collision may occur at node A. Therefore, it is necessary to schedule CTS packet response time to avoid CTS packet collision. The proposed MPT-MAC protocol schedules CTS response time of each destination node and then, transmits RTS packet which contains this information to avoid collision between CTS packets. For calculating CTS packet response time of each destination node, first of all, a node sorts destination nodes by using the propagation delay on the neighbor node list in ascending order. If the number of

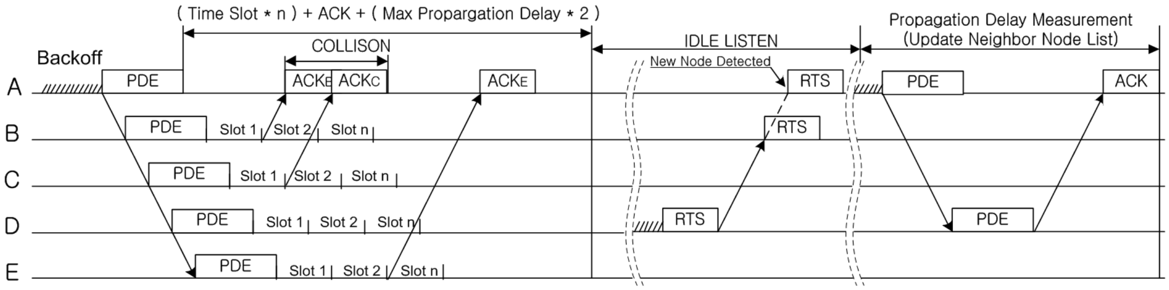
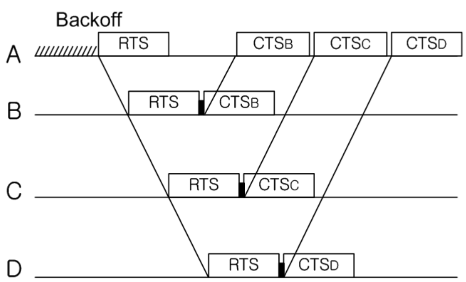
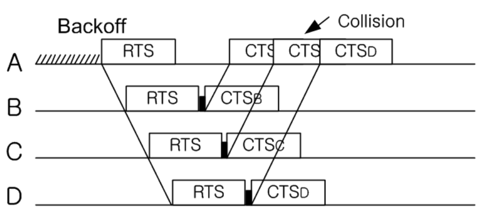


Fig. 2: Round Trip Time Measurement Modified for MPT-MAC



(a) collision-free case of CTS packets.



(b) collision case of CTS packets.

Fig. 3: Example of RTS-CTS Transmission

destination nodes is n, define the closest destination node ID as 1, and the farthest destination node ID as n. To calculate the CTS response time, we define two types of notation: CTS_START_i and CTS_END_i. CTS_START_i is starting time of CTS packet arrival, sent from destination node i, at the sender, and CTS_END_i is completion time of CTS packet arrival. They are calculated as follow. Based on starting point of RTS packet transmission, CTS_START_i is '$RTS_{time} + PD_i \times 2 + SIFS$', and CTS_END_i is '$CTS\_START_i + CTS_{time}$'. RTS_{time} and CTS_{time} are transmission time of RTS and CTS packet respectively, and PD_i is the propagation delay of destination node i. In order to avoid collision of CTS packets, CTS packet response time is scheduled by using CTS_START_i and CTS_END_i of each destination node. If CTS_START_i is bigger than '$CTS\_END_{i-1} + SIFS$', CTS packet response time is not necessary to be adjusted, because no collision happens in this case. However, if CTS_START_i is smaller than '$CTS\_END_{i-1} + SIFS$', it means the collision happens, so CTS response time should be adjusted. In this case, CTS_START_i of destination node i is set as '$CTS\_END_{i-1} + SIFS$' to avoid collision. Therefore, CTS_END_i becomes '$CTS\_START_{i-1} + CTS_{time}$'. Repeat the above process from the first to the last destination node sequentially. PD_i is time difference from the time that destination node i transmits CTS packet to the time that a sender receives it, so CTS response time of destination node i is '$CTS\_START_i - PD_i$'.

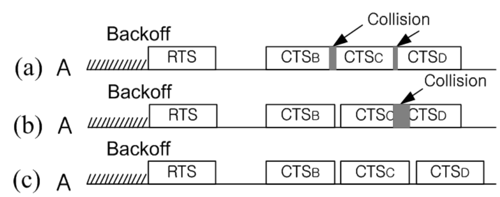


Fig. 4: Example of Collision Avoidance among CTS Packets

Fig. 4 shows an example of CTS response time scheduling for resolving CTS packet collision problem in Fig. 3(b). Fig. 4(a) is the node A in Fig. 3(b). Fig. 4(b) shows that CTS packet response time of node C is adjusted as '$CTS\_END_B + SIFS$' to avoid collision between node B and C, which was caused by the reason that start time of CTS packet at node C is less than END time of CTS packet at node B. Fig. 4(c) explains that CTS response time of node D is adjusted as '$CTS\_END_C + SIFS$' to avoid CTS packet collision between node C and D. Once CTS packet is received, node A transmits data in the related transmission queue to each destination node as shown in Fig. 5. DATA packets are transmitted sequentially from the node with the smallest propagation delay. The DATA packet transmitted at this time contains information of ACK packet response time as shown in Fig. 5.

To calculate the ACK response time, we define two types of notation: ACK_START_i and ACK_END_i. They are similar to CTS_START_i and CTS_END_i. ACK_START_i is starting time of ACK packet arrival, sent from

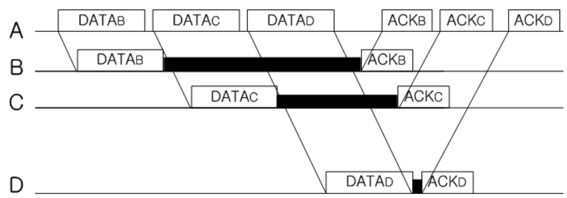


Fig. 5: Example of DATA-ACK Transmission

destination node i, at the sender, and ACK_END_i is completion time of CTS packet arrival. ACK_START_i is calculated as follows:

$$ACK\_START_i = \sum_{i=1}^{n} DATA_{time} + SIFS \times (n-1) + PD_i \times 2 + SIFS$$

where, $DATA_{time}$ and ACK_{time} are transmission time of DATA and ACK packet for destination node respectively. ACK_END_i is '$ACK\_START_i + ACK_{time}$'. ACK_START_i and ACK_END_i of each destination node are used to resolve ACK packet collision problem. If ACK_START_i is bigger than '$ACK\_END_{i-1} + SIFS$', ACK packet response time is not necessary to be adjusted, because no collision happens in this case. However, if ACK_START_i is smaller than '$ACK\_END_{i-1} + SIFS$', it means the collision happens, so ACK response time should be adjusted. In this case, ACK_START_i of destination node i is set as '$ACK\_END_{i-1} + SIFS$' to avoid collision. Therefore, ACK_END_i becomes '$ACK\_START_{i-1} + ACK_{time}$'. Repeat the above process from the first to the last destination node sequentially. PD_i is time difference from the time that destination node i transmits ACK packet to the time that sender receives it, so ACK transmission time of destination node i is '$ACK\_START_i - PD_i$'. In case of a sender does not receive ACK packet from neighbor nodes, channel contention is started again. All packets used for transmission have a duration field as specified at 802.11 standard[8], and channel occupancy time is transmitted by using this field. Each node which received packet sets NAV(Network Allocation Vector) timer as received the duration field value, and stays in IDLE state until the timer becomes '0'.

3 EXPERIMENT

In this section, we discuss the performance of the proposed scheme. We have implemented the proposed scheme with the NS-3 simulator. In the simulation, we consider the topology as shown in Fig. 6, where there are 25 nodes. In the simulation, each source and destination pair is fixed. There are 12 flows in both the horizontal and vertical directions, as shown in Fig. 6. We use a static

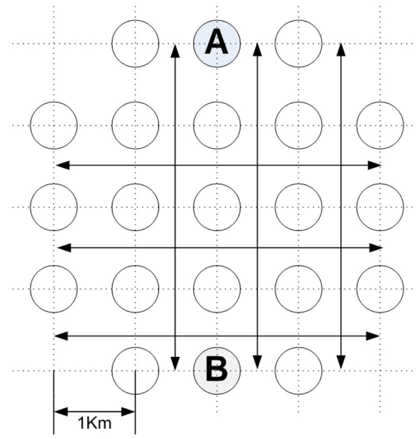


Fig. 6: Simulation Topology

routing protocol. Node transmission speed is set as 2,400bps, distance between nodes is set as 1km, and the transmission range of the nodes is limited as 1km which is the distance of one hop. A CBR (Constant Bit Rate) model is used. Each source node generates a packet per second, and packet size varies from 100 bytes to 1,000 bytes. Therefore, each source node sends data packets at a rate of from 800bps to 8,000bps.

Main performance metrics of interest are throughput and hop delay. The throughput is the amount of data, successfully transmitted from node B to node A in a given time period that it is measured in bits per second (bps). The hop delay is the average per hop time required to deliver the packet from node B to node A. it is as follows:

$$HopDelay = \frac{PacketArrivalTime - PacketGenerationTime}{NumberOfHops}$$

For MPT-MAC performance comparison, performance of MACA-U[9] was measured in the same condition. MPT-MAC and MACA-U include ACK packet in the simulation.

Fig. 7 is the results for the throughput according to the packet length. From the figure, we can see that the proposed scheme has better performance than the MACA-U scheme regardless of the variation of packet lengths. Compared to MACA-U, MPT-MAC has 40% higher throughput at small packet size, 81% at medium packet size, and 74% at large packet size. At medium packet size, performances of both schemes are dramatically dropped. As the packet size increases, transmission time increases. Therefore, collision probability becomes higher. Consequently, network throughput decreases hugely.

Fig. 8 shows the simulation results for the hop delay according to the packet length. The proposed MPT-MAC has about 33% lower hop delay than the MACA-U. Delay value is slowly growing at small packet size, but it is

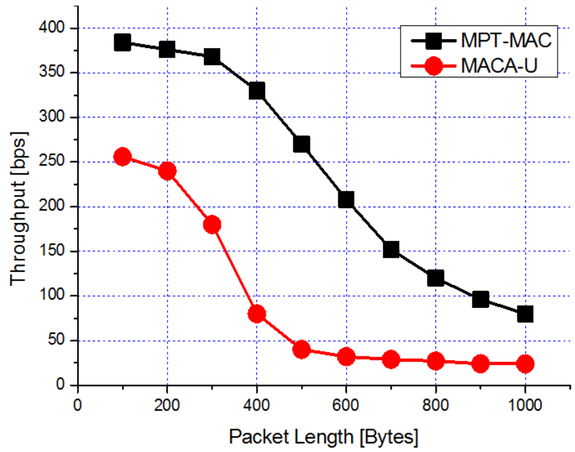


Fig. 7: Throughput VS. Packet Length

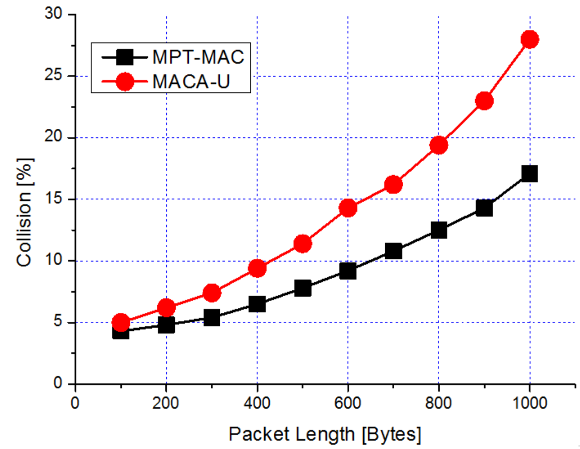


Fig. 9: Collision Rate VS. Packet Length

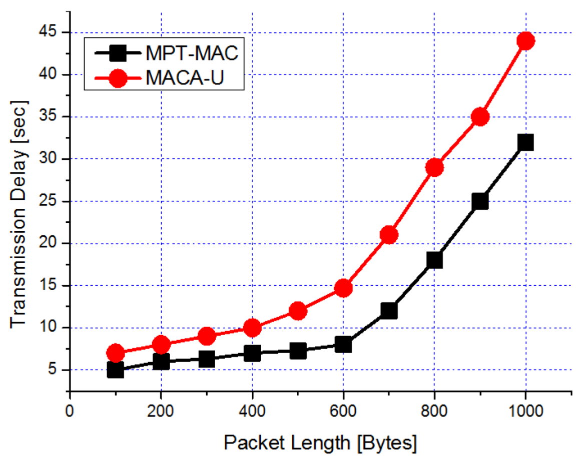


Fig. 8: Transmission Delay VS. Packet Length

growing rapidly from medium packet size. Collided packets are retransmitted until they are successfully delivered, or they reach their retry limit. This retransmission increases channel contention between nodes as time goes by, and also causes huge increase of transmission delay by increasing CW(Contention Window)[10] value dramatically. However, MPT-MAC reduces the number of contentions for data transmission between neighbor nodes. As a result, the increase of hop delay is relatively slower compared to MACA-U.

Fig. 9 shows the simulation results for the collision count according to the packet length. In this paper, collision is defined as when the source node does not receive the ACK. The Collision rate is fail rate of data packet transmission from node A to its neighbors. It is as follows:

Compared to MACA-U, MPT-MAC has 1.4% lower collision rate at small packet size, 3.1% at medium packet size, and 8% at large packet size. As the packet size increases, transmission time increases. Therefore, collision probability becomes higher. Consequently, collision rate increases. MPT-MAC has a longer transmission time than MACA-U. However, proposed MAC has a lower collision rate than MACA-U. This is due to the fact that MPT-MAC decreases channel contention and thus leading to lower number of collision. When there are many flows as shown in Fig. 6, queue size of each node increases due to collisions. Different from MACA-U, MPT-MAC transmits a number of packets to neighbor destination nodes with only one channel reservation. Therefore, it fast reduces the queue size and channel contention between nodes. It is noticed from the simulation results that MPT-MAC, compared to MACA-U, has outstanding performance due to lower channel contention in networks where complex traffic exists. This experiment result shows that MPT-MAC is appropriate to be used in complex traffic condition.

4 Conclusions

In this paper, we proposed MPT-MAC protocol which is for performance increase in underwater sensor network condition which has complex transmission routes. MPT-MAC protocol measures propagation delay time between neighbor nodes, and schedules the packets sent from neighbor nodes to avoid collision by using the measured propagation delay time. The proposed MAC implements channel reservation with various destination nodes by using single RTS packet. At this point, CTS packets, generated from various destination nodes, make channel reservation difficult due to collisions between packets. MPT-MAC protocol resolves this CTS packet collision problem by using CTStime. As soon as the channel reservation is completed with neighbor nodes, DATA is transmitted to each destination node. At this time, collision problem between ACK packets, transmitted by each destination node, is resolved by using ACK time. In this paper, MPT-MAC operation method is explained, and performance of the protocol is compared to MACA-U by using simulator. In UWASN environment, it is approved that the proposed MAC in this paper shows outstanding performance compared to existing MAC protocols.

References

[1] E. Sozer, M. Stojanovic, J. Proakis, Underwater Acoustic Networks, IEEE Journal of Oceanic Engineering VOL **25** NO **1**, 72-83 (2000).

[2] X. Guo, M. R. Frater, M. J. Ryan, A propagation-delay-tolerant collision avoidance protocol for underwater acoustic sensor networks, Proceedings of OCEANS 2006 - Asia Pacific, 1-6 (2007).

[3] I. F. Akyildiz, D. Pompili, T. Melodia, Underwater Acoustic Sensor Networks: Research Challenges, Elseviers Journal of Ad Hoc Networks VOL **3**, NO **3**, 257-279 (2005).

[4] D. Pompili, I. F. Akyildiz, Overview of networking protocols for underwater wireless communications, Proceedings of IEEE Communication Magazine, Feature Topic Underwater Wireless Communication, 97-102 (2009).

[5] Brakmo, L. S., L. L. Peterson., TCP Vegas:end-to-end congestion avoidance on a global Internet, IEEE Journal on Selected Areas in Communications VOL **13(8)**, 1465-1480.

[6] N.Abramson, The ALOHA System: Another Alternative for Computer Communications, Proceedings of 1970 Fall Joint Computer Conference. AFIPS Press, 17-19 (1970).

[7] Wenhai Liu, Jiarong Zhang, Xiangliang Yu , An improved BEB algorithm for MAC protocol of underwater sensor networks, Proceedings of Broadband Network & Multimedia Technology, 2009. IC-BNMT '09. 2nd IEEE International Conference, 800-804 (2009).

[8] IEEE. Part 11: wireless LAN medium access control (MAC) and physical layer (PHY) specifications. IEEE Standard 802.11. (1999).

[9] Hai-Heng Ng, Wee-Seng Soh, Mehul Motani, MACA-U: A Media Access Protocol for Underwater Acoustic Networks, Proceedings of Global Teleommuni-ations Conference, 2008. IEEE GLOBECOM 2008. IEEE, 1-5 (2008).

[10] A. Ksentini, A. Nafaa, A. Gueroui, M. Naimi, Deterinist contention window algorithm for IEEE 802.11, Proceedings of Personal, Indoor and Mobile Radio Communications, 2005. PIMRC 2005. IEEE 16th International Symposium VOL **4**, 2712-2716 (2005).

Dongwon Lee received the MS and Ph.D. degree in computer software engineering from Kumoh National Institute of Technology, Gumi, Korea, in 2008 and 2012 respectively. He is currently a senior research engineer in Hanwha Coporation, Gumi, Korea. His research interests are in the areas of Underwater Acoustic Sensor Networks, Wireless Sensor Network and Modem design for wireless communication.

Sunmyeng Kim received the B.S., M.S., and Ph.D. degrees in information and communication from Ajou University, Suwon, Korea, in 2000, 2002, and 2006, respectively. From May 2006 to February 2008, he was a Postdoctoral Researcher in electrical and computer engineering with the University of Florida, Gainesville. In March 2008, he then joined the School of Computer and Software Engineering, Kumoh National Institute of Technology, Gumi, Korea, as an assistant professor. His research interests include resource management, wireless LANs and PANs, wireless mesh networks, UWSNs and quality of service enhancement.