Wireless Pers Commun (2017) 93:1027–1041
DOI 10.1007/s11277-017-3964-2

An Efficient MOP Decision Method Using Hop Interval for RPL-Based Underwater Sensor Networks

Sungwon Lee[1] · Yonghwan Jeong[1] · Eunbae Moon[1] · Dongkyun Kim[1]

Published online: 24 January 2017

Abstract In underwater wireless sensor networks (UWSNs), not only data packets but also sensor nodes can be lost due to ocean current, running out of battery and etc. Among various routing protocols, Routing Protocol for Low Power and Lossy Networks (RPL) is a promising routing protocol to overcome this problem since it provides multiple upward path establishments and path repair methods. However, unlike upward paths, downward routing in RPL have low reliability since it source routing technique for reducing routing table size. In order to increase the reliability of downward paths, RPL standard defines two operational modes (Mode Of oPeration, MOP) for intermediate nodes, i.e., storing mode and non-storing mode. A node which operates in storing mode can increase the reliability, however it cannot be operated in storing mode for the all instances due to the limited memory. In this paper, we therefore propose a hop interval based MOP decision method for intermediate nodes. In our proposed method, the sink calculates the hop interval between storing nodes based on the number of application and each intermediate nodes decide their MOP according to the calculated hop interval. In addition, we propose a QoS-based initial storing node decision scheme to reflect differences between swallow and deep water.

Keywords UWSNs · RPL · Downward path · Mode of operation

✉ Dongkyun Kim
dongkyun@knu.ac.kr

Sungwon Lee
swlee@monet.knu.ac.kr

Yonghwan Jeong
yhjeong@monet.knu.ac.kr

Eunbae Moon
ebmoon@monet.knu.ac.kr

[1] School of Computer Science and Engineering, Kyungpook National University, Daegu, Korea

1 Introduction

Recently, researchers have paid much attention to underwater wireless sensor networks (UWSNs) to support a variety of applications such as pollution detection, tactical surveillance and so on. Different from terrestrial sensor networks, UWSNs have distinctive characteristics such as high propagation delay, limited bandwidth and high error rate since acoustic signals are used for underwater communications, rather than radio signals. Hence, new communication protocols should be developed to take into account these characteristics [1, 2].

Particularly, in network layer, many routing protocols have been proposed to enable a packet from sensor nodes to reach the sink. Among them, the flooding-based routing protocols which can transmit data packets from sensor node to the sink along multiple paths are known as appropriate routing protocols for UWSNs [3, 4]. In most of flooding-based routing protocols, a sensor node which has a data packet sets a flooding area based on geographical information and broadcasts the data packet. Neighbor nodes located in the flooding area will participate in next forwarding to deliver the data packet towards the sink. These routing protocols also increase the packet delivery ratio by allowing multiple copies of a packet to reach the sink along different paths. However, these routing protocols still suffer from various problems such as void problem. In addition, the flooding based routing protocols require GPS location information of the destination node. Different from the sink, underwater sensor nodes have mobility due to ocean current and the sink cannot know the current location of the destination sensor node. Therefore, flooding-based routing protocols cannot support downward packet transmission efficiently in UWSNs.

Nowadays, various underwater applications which generate downward traffics for supporting software update, actuator control and etc are newly developed [5]. Moreover, multiple underwater applications such as pollution detection application and co-operative pollution elimination application have operated in same node. The co-operation of multiple applications is a current trend in sensor networks because maintenance of the sensor nodes is costly [6]. In this case, routing protocols should reflect the different requirements of the applications and provide downward path [7]. Among various routing protocols, RPL (Routing Protocol for Low Power and Lossy Network) which is standardized from Internet Engineering Task Force (IETF) can be a promising routing protocol for current UWSNs since it provides not only downward paths but also multiple instance management methods [8]. However, different from upward routing which establishes multiple upward paths, downward routing has low reliability since the downward routing uses source routing technique for reducing routing table size.

To increase reliability of the downward paths, RPL standard defines Mode Of oPeration (MOP) which consist of non-storing mode and storing mode for intermediate nodes (the detailed operations of RPL will be described in the next section) [9]. A node which works in storing mode provide partial source routing and local repair procedures. Since these procedures can be operated based on downward routing table, a storing node spend its memory space. In contrast to the storing mode, non-storing mode only support packet relay operation without any memory consumption.

Therefore, as the number of storing node increases, throughput and reliability of downward paths are also increased. However, when multiple applications are operated, each sensor nodes cannot be operated as storing node in the all instances due to limited memory capacities. In other words, if an intermediate node belongs to N instances and its limited memory spaces are enough to be storing node only in K instances, the node might

be operated as storing node and non-storing node in K instances and (N–K) instances, respectively. However, current RPL standard does not define any method to decide MOP of the intermediate node for each instances. Hence, we proposed a hop interval based MOP decision method for terrestrial IoT networks in our previous work [10]. To the best of our knowledge, our MOP decision method is the first MOP decision method for RPL.

In this paper, we extend our previous work to reflect the characteristics of underwater environment. In our proposed method, the sink calculates a hop interval value between storing nodes based on the number of application in UWSNs. When an intermediate node receives routing message, it checks it's hop distance to find the closest storing node among its ancestor nodes. If the closest storing node is located closer than the calculated hop interval, the intermediate node operates as non-storing node to save its memory space. Otherwise, if the closest storing node is located further than the calculated hop interval, the intermediate node becomes a storing node in this instance. Moreover, our propose scheme selects a location of the first storing node over a downward path to reflect differences between swallow and deep water.

The rest of the paper is organized as follows. In Sect. 2, routing procedure of RPL standard is described. In Sect. 3, we explain our proposed MOP decision method. In Sect. 4, we evaluate the performance of the proposed method. Section 5 concludes the paper.

2 Related Work

2.1 Upward Routing in RPL Standard

In RPL standard, the sink establishes a DODAG (Destination Oriented Directed Acyclic Graph) for not only upward paths but also downward paths. To establish the DODAG, the sink floods DODAG Information Object (DIO) message in the entire network periodically. When an intermediate node receives multiple DIO messages which are transmitted along different paths, it stores all DIO transmitter in its upward routing table as feasible successors. After then, if the node has data packets towards the sink, it selects a preferred parent node which includes the lowest routing metric among the feasible successors and transmit the packets to selected preferred node.

During this packet transmission phase, if the preferred node is lost, the intermediate node selects a new preferred node according to the routing metric and transmits the packets to the new preferred node, immediately.

2.2 Downward Routing in RPL Standard

Different from the upward routing described in Sect. 2.1, downward routing is defined as an optional operation in RPL standard. In order to provide downward routing, RPL relies on source routing mechanism for low power devices. An intermediate node which receives DIO message transmits DAO (Destination Advertisement Object) towards the sink along its upward paths. A preferred parent which receives the DAO message records its address in the received DAO message and relays the DAO message. Based on addresses which are recorded in the DAO message, the sink maintains downward routing table.

After then, if the sink generates a downward data packet, it finds a downward path which has to deliver the data packet to the destination. The sink records all addresses of

intermediate nodes from the sink to the destination in header of the packet. Based on the recorded address information, intermediate nodes which receive the data packet determine their next hop without downward routing table likely Fig. 1.

However, if the determined next hop node is lost, the intermediate node cannot determine the new next hop by itself since the intermediate node does not have any routing table. Hence, RPL standard defines a path repair method. In this path repair method, the intermediate node which detects path loss sends a DIS (DODAG Information Solicitation) message to the sink or its neighbors. When a sink receives the DIS message, the sink is able to flood the new DIO message to whole network to renew both upward and downward paths.

2.3 Mode of Operation

Even though the downward routing is able to support downward traffic without additional routing table, it suffers from various problem due to characteristics of source routing mechanism. In particular, increased header size and high overhead for path repair are known as the most important problem of source routing technique.

In order to overcome these problems, RPL standard defines MOP (Mode Of oPeration) which consist of storing mode and non-storing mode for intermediate node. If an

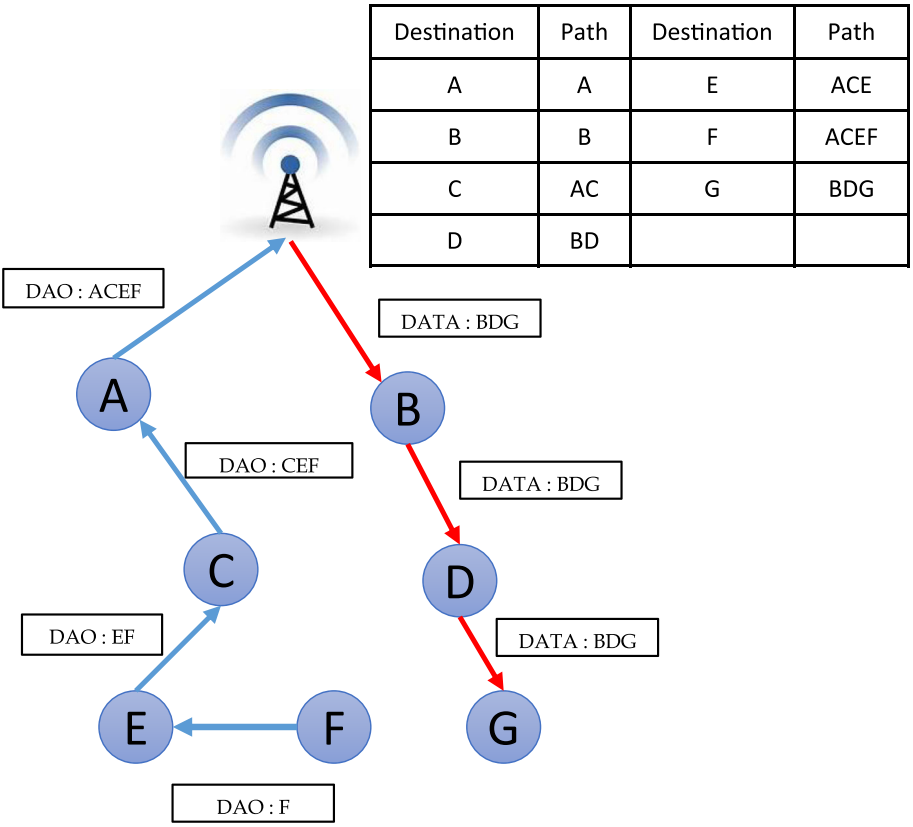


Fig. 1 Downward routing in RPL standard

intermediate node determines its MOP as non-storing mode, the operations of intermediate node which were described in Sect. 2.2 is performed. Otherwise, if an intermediate node determines its MOP as storing mode, it maintains downward routing table. Based on the routing table, the storing mode supports partial source routing. In addition, if a storing node receives DIS message from its children, it is able to flood DIO message likely to the sink. The path repair method is denoted by local repair.

Figure 2 shows an example of the storing mode. In this example, node B and C operate as storing mode. When node C receives DAO message from node E, it stores the downward paths towards both of node E and F. Similarly, node B maintains downward paths for node D and G. When the sink transmits a data packet to node G, the data packet includes only addresses of node B and G. If the storing node B receives the packet, it updates the packet header (from BG to DG) based its routing table and transmits to node D. Moreover, when a non-storing node D detects path loss and sends DIS message, the storing node B floods DIO message to update downward path between the storing node B and the destination.

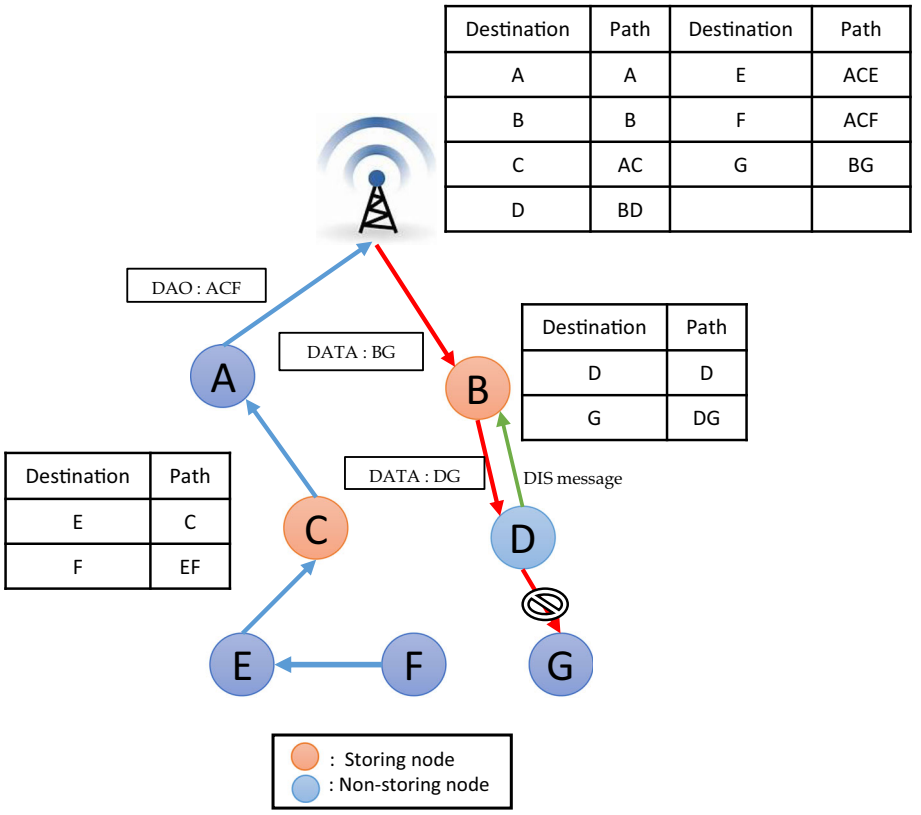


Fig. 2 Operation of storing mode

3 Proposed Scheme

3.1 Overview

As mentioned above, we propose a Hop-Interval based Mode Of oPeration Decision (HI-MOPD) method which assigns mode of operation for intermediate nodes. When a route establishment phase is started, the sink calculates an Optimal Hop Interval between Storing Nodes (OHIS) value which means "how many non-storing nodes should operate between storing nodes for each instances" based on number of instances.

After calculating the OHIS value, the sink floods a DIO message which includes the calculated OHIS value to establish upward and downward paths. When an intermediate node receives the DIO message, it measures the hop interval from itself to the closest storing node among its ancestor node.

If the measured hop interval is lower than OHIS value which is included in the received DIO message, the node decides its MOP as non-storing mode. Otherwise, if the measured hop interval is bigger than the OHIS value, the node operates as storing mode to provide partial source routing and local repair between itself and next storing node.

3.2 Assumptions

Our HI-MOPD is designed on following assumptions.

1. Every underwater sensor nodes have same specification such as memory spaces.
2. The sink knows sensor node's memory specification information. Assumptions 1 and 2 are reasonable because the sink and all sensor nodes might be made at the same company for supporting co-operative applications.
3. The sink knows how many instances will be operated. Because the number of instances and the number of applications which are installed at the sink are same, therefore it is easy for sink to find out the number of instances.
4. The QoS information of each applications are given when the applications are installed.
5. Since All underwater sensor nodes are deployed randomly. If the sensor nodes are deployed according to node deployment method, operation mode could be determined based on the deployment method.

3.3 OHIS Calculation

In order to calculate OHIS value which represent hop interval between storing nodes, the sink checks the number of instances in the UWSNs (denoted by NoI) and the number of instances which sensor node can operate as storing mode (denoted by NoIS). Because all UWSNs applications should be installed in the sink, the sink can check the number of instances in the UWSNs without additional overhead. When the sink floods DIO message to establish DODAG, the sink calculates the OHIS value according to following equation.

$$OHIS = \lfloor NoIS/NoI \rfloor \tag{1}$$

Figure 3 introduces an example to verify the Eq. 1. In this example, NoI and NoIS are set to 3 and 1, respectively.

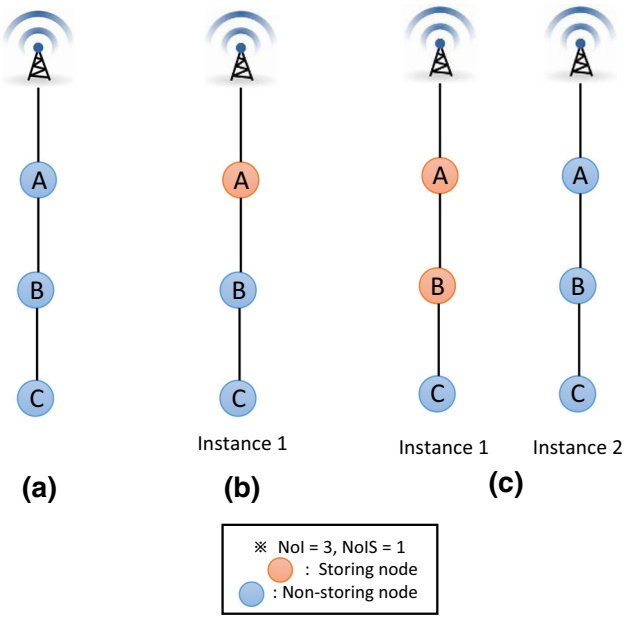


Fig. 3 Example of MOP decision

In this example (Fig. 3a), a node A receives DIO messages of instance 1 from the sink. Since the node A has enough memory spaces to work in storing mode, it becomes storing node likely Fig. 3b for instance 1. After this procedure, the node A receives new DIO messages of instance 2 and 3. Because the node A already spent its memory spaces for instance 1, it should operate as non-storing mode for instance 2 and 3.

Next, the node A broadcasts the DIO messages (for instances 1, 2 and 3) and a node B receives these 3 DIO messages. If the node B operates as storing node for instance 1, partial source and local repair will be provided in instance 1 only when instance 2 and 3 do not include any storing nodes. Otherwise, if the node B operates in non-storing mode for instance 1, it can operate as storing mode for instance 2 (likely Fig. 3c). In this situation, partial source and local repair can be performed in not only instance 1 but also instance 2. Due to the same reason, the node C should operate in storing mode for instance 3 rather than instance 1 or 2 as shown in Fig. 4. Therefore, the optimal hop interval between storing nodes is calculated as 3 in this example.

3.4 Operation for MOP Decision

After calculating OHIS value, the sink broadcasts DIO messages to establish upward paths for each instances. In these DIO messages, two additional fields namely OHIS field and Hop Count after Storing node (HCS) field are included. The sink records the calculated OHIS value and OHIS + 1 into the OHIS field and HCS field, respectively. The HCS field records the number of non-storing nodes which are operated after the last storing node.

When an intermediate node receives a DIO message from its parent node, it checks whether its available memory space is enough to operate as storing node or not. If the intermediate node does not have enough memory capacity to work as storing node, it becomes non-storing node for this instance. The intermediate node increases HCS value by

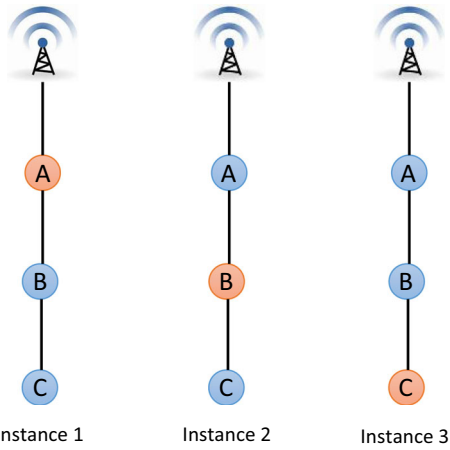


Fig. 4 Optimal MOP decision

1 and broadcasts the updated DIO message. Otherwise, if the intermediate node has enough memory capacity to operate in storing mode, it compares two values which are recorded HCS and OHIS field, respectively.

If the OHIS value is lower than HCS, at least one ancestor node works as storing mode within OHIS hops. Hence, the intermediate node works as non-storing mode to save its memory space. After this procedure, the intermediate node increases HCS field by 1 and broadcasts the DIO message to its children. Otherwise, if the OHIS value is bigger or equal to HCS, it means that the all ancestor nodes operate as non-storing mode within the counted number of hops for HCS. Hence, the intermediate node operates in storing mode to reduce the header size of downward packet and provide local repair for this instance. After then, the intermediate node resets HCS field to 1 and broadcasts the DIO message. Following algorithm represents the proposed operation for MOP decision.

Algorithm 1 Hop Interval based MOP Decision Method

(1) A new DIO message is received;
(2) OHIS and HCS value is a recorded in the DIO message;
(3) Node can check its memory capacity(MEM_CAP);
if ($MEM\_CAP < Required\_Memory\_For\_StoringMode$) **then**
  Operates as Non-storing Mode;
else
  if ($HCS < OHIS$) **then**
    Operates as Non-storing Mode;
    $HCS \leftarrow HCS + 1$;
    Broadcast DIO Message;
  end if
  if ($HCS >= OHIS$) **then**
    Operates as Storing Mode;
    $HCS \leftarrow 1$
    Broadcast DIO Message;
  end if
end if

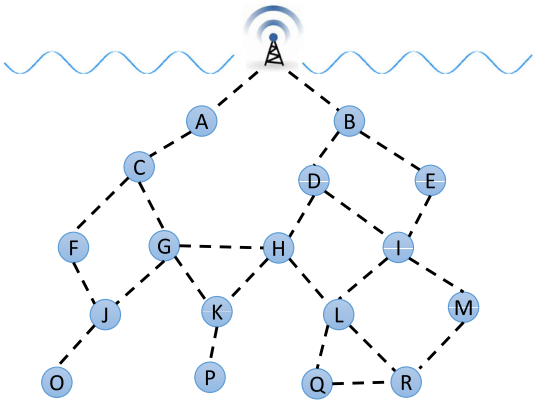


Fig. 5 Example of UWSNs topology

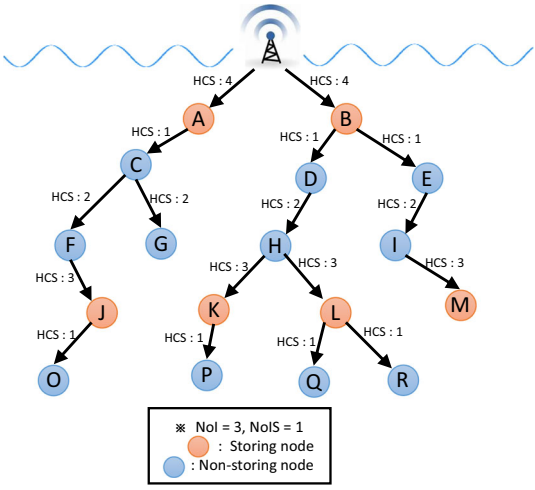


Fig. 6 MOP decision in instance 1

Figures 5, 6 and 7 introduce examples of proposed MOP decision scheme. In this example, the sink calculates OHIS value as 3 because NoI and NoIS are 3 and 1, respectively.

When underwater sensor nodes are deployed likely Fig. 5, node A and B receive DIO messages for instance 1 from the sink. Since these nodes have memory space and HCS value in the received DIO messages are bigger than OHIS. Hence, node A and B operate in storing mode for this instance and update DIO message. When the node C, D and F receive the DIO message from node A and B, they operate as non-storing mode since the HCS value in the updated DIO messages is 1. For the same reason, Node H, I, J and K also operate as non-storing mode for the instance 1 because they receive DIO message which include 2 HCS value. Different from these nodes, node L, M, N and O receive DIO messages which include 3 HCS value from their parent nodes. Since the HCS value is equal to OHIS, node L, M, N and O operate in storing mode for the instance 1 likely Fig. 6.

Figure 7 shows the result of proposed MOP decision scheme in instance 2. Since each instances use different routing metric, the downward paths of instance 2 can be different from the downward paths of instance 1. In the instance 2, node A and B operate as non-

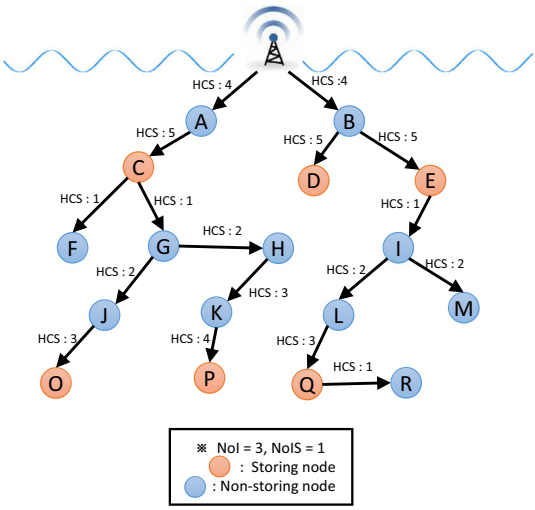


Fig. 7 MOP decision in instance 2

storing node because they already spent their memory spaces for instance 1. Node C, D and F become storing node because they receive DIO message which include 5 HCS value. After then, each intermediate nodes decide their MOP as mentioned in proposed scheme. During this procedure, node K receives a DIO message from node H which include 3 HCS value. In this situation, node K operates as non-storing mode for instance 2 since it spent its memory space for instance 1. On behalf of node K, node P which is a child node of node K could be operated as storing mode to provide partial source routing and local repair for instance 2.

3.5 Operation for Initial Storing Node Allocation

As mentioned above, our proposed scheme allows each intermediate nodes to decide their MOP for providing partial source routing and local repair. If our proposed scheme operates correctly, each instances will have same number of storing nodes and same hop interval between the storing nodes. However, the location of first storing node would be different among these instances. As the first storing node is located closer to the sink, the throughput of downward path could be increased since intermediate nodes from the sink to the first storing node operate as non-storing node.

Of course, the increment of throughput might be infinitesimal in the terrestrial sensor networks. However, in the underwater environment, the throughput increase due to the characteristics of swallow water. In swallow water, lots of local repair operations are performed because winds, fishes, ships and ocean current disturb more acoustic signal propagation and navigation system of nodes compared to deep water [11]. If an instance does not have any storing node in swallow water, the throughput of the instance can be decreased. Therefore, if an instance has high QoS requirement for downward traffics, it has to include at least one of the storing node in swallow water.

Therefore, our proposed MOP decision method includes one optional operation to determine the location of the first storing node in each instance based on QoS requirement of application. In this operation, the sink classifies the applications into QoS level from 0 to OHIS. As an application generates downward traffic frequently and the QoS requirement

of the downward traffic is high, increased QoS level (*QLevel_Inst*) is assigned to this instance. Based on the QoS level, the sink modifies initial HCS value in DIO message according to following equation.

$$HCS\_Init = QLevel\_Inst + 1 \tag{2}$$

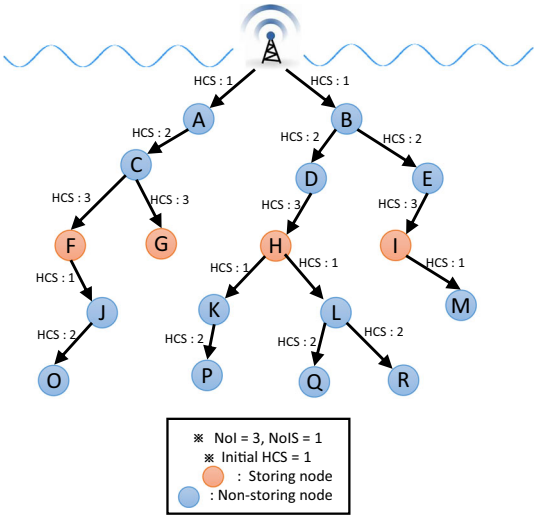
For example, if an instance is activated and the sink sets QoS level of this instance to level 0 (minimum QoS requirement), the initial HCS value is set to 1 as shown in Fig. 8. Node A and B which are located in swallow water operate as non-storing mode since the recorded HCS value is less than OHIS value. After then, if a new instance which has high QoS requirement for downward traffics is activated, the sink sets the QoS level to 3 for the instance. In this instance, node A and B would be operated as storing mode because the initial HCS value is bigger than OHIS value and they saved their memory capacities for the previous instance (instance 1).

4 Performance Evaluation

4.1 Simulation Environments

We evaluated the performance of our proposed scheme (HI-MOPD) against the random-based MOP decision method in underwater environment. For this evaluation, we used Network Simulator 2 (NS-2) since the simulater has underwater channel and PHY models which are designed according to real underwater environments [12]. We positioned 300 sensor nodes at random locations in the square of 2000 m $\times$ 2000 m. One sink was located at the top center of the square. The maximum transmission distance of each node is set to 250 m and each the payload size was set to 512 bytes. Among the sensor nodes which were located more than 5 hops away from the sink, source nodes were chosen randomly. We also set NoIS and NoI to 2 and N, respectively. The QoS level of each instances were chosen randomly.

Fig. 8 Example of initial HCS



To emulate high packet loss rate in UWSNs, link error rate were set to 12 and 3% for swallow water (400 m depth from sea surface) and deep water, respectively. Moreover, if hop-by-hop packet retransmissions was performed 3 times, the packet forwarder node tried to repair the downward path according to its MOP.

4.2 Simulation Results

First, we investigated the average packet length of downward packets which were transmitted in all instances (refer Fig. 9). Because the constant payload size was set, the average packet length depends on how many addresses are stored in packet header for source routing technique.

If the number of instance(NoI) equals to 2, the average packet size is same in both of the proposed scheme and random-based MOP decision scheme since the all intermediate nodes have enough memory capacity to operate as storing mode for the all instances. However, when NoI is bigger than 2, our proposed method has lower average packet size than random-based MOP decision scheme regardless of the NoI. In our proposed method, intermediate nodes which operate as storing modes are deployed in every OHIS hops and they perform partial source routing to reduce header length. On the other hand, random-based MOP decision method has longer header length since the hop interval between storing nodes can be longer than OHIS hops. Due to the limited bandwidth of UWSNs, the long header length decreases the throughput of downward traffics.

We also evaluated the number of DIS message transmission in the entire network for both of our proposed MOP decision method and random-based MOP decision method as shown in Fig. 10. If NoI equals to 2, DIS message is not transmitted since the all intermediate node perform local repair when downward path is broken. Otherwise, if NoI is bigger than 2, proposed MOP decision method has lower number of DIS message transmission since at least one storing node operates within OHIS hops.

Finally, we investigated the throughput of each instances which have different QoS level in our proposed scheme. Because link error rate in swallow water is higher than deep water, more local repair are performed in swallow water rather than deep water. Since the local repair overhead spend high bandwidth, throughput is increased as many storing node are operated in swallow water. In our proposed scheme, the sink sets increased initial HCS value to DIO message according to QoS level of each instances. Therefore, throughput of

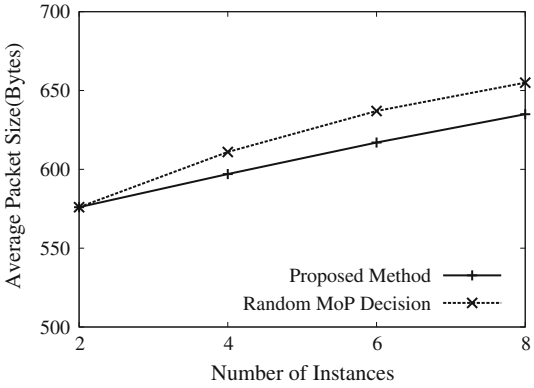


Fig. 9 Average packet size

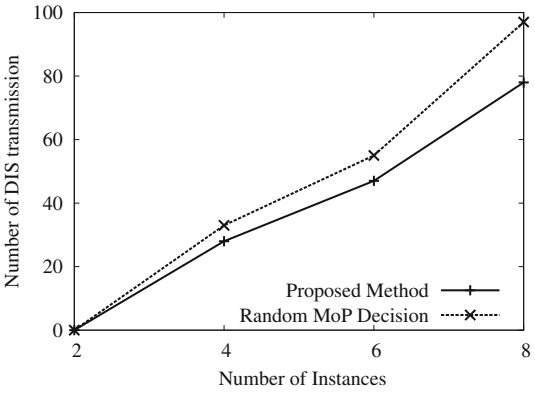


Fig. 10 Path repair overhead

instances which have high QoS requirement increases since hop distance from the the sink to the first storing node is decreased (Fig. 11).

5 Conclusions

In this paper, we proposed a MOP decision method for memory-limited RPL nodes to increase throughput and reduce route repair overhead in underwater wireless sensor networks. In our proposed method, the sink calculates an Optimal Hop Interval between Storing Nodes (OHIS) based on number of instances and memory capacity of each nodes. When an intermediate node receives DIO message, it checks its memory capacity and hop distance to find the closest storing node among its ancestor nodes. If memory space of the node is enough to operate as storing mode and any ancestor nodes are not operated as storing node within OHIS hop, the node operates as storing node to provide partial source routing and local repair.

We observed that our proposed method shows performance improvements in the throughput and path repair recovery overhead decrements as compared to the random based MOP decision method, respectively. Moreover, we observed that our first storing node allocation operation can provide improved throughput to instances which have high QoS requirement for downward traffic. Of course, if an instance has high level of QoS

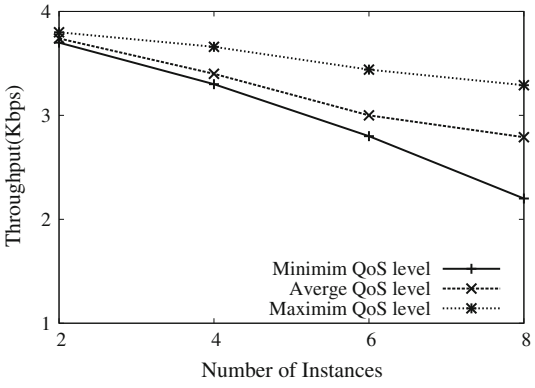


Fig. 11 Throughput

requirement for downward traffics, a low OHIS value should be allocated even though the first storing node allocation operation is implemented. It is our future work.

Acknowledgements This research was supported by the Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education, Science and Technology (NRF-2016R1D1A3B01015510).

References

1. Han, G., Jiang, J., Bao, N., Wan, L., & Guizani, M. Routing Protocols for Underwater Wireless Sensor Networks. *IEEE Communications Magazine*, *53*.
2. Pompili, D., & Akyildiz, I. F. (2009). Overview of networking protocols for underwater wireless communication. *IEEE Communication Magazine 2007*, vol. 47, pp. 97–102, February 2009.
3. Shin, D., Hwang, D., & Kim, D. (2012). DFR: An efficient directional flooding-based routing protocol in underwater wireless sensor networks. *Wireless Communication and Mobile Computing*, *12*, 1517–1527.
4. Rahman, R. H., Benson, C., & Frater, M. (2012). Routing protocol for underwater ad hoc networks. *OCEANS 2012*, May 2012.
5. Heidemann, J., Stojanovic, M., & Zorzi, M. (2011). Underwater sensor networks: Applications, advances and challenges. *Philosophical Transactions of the Royal Society A Mathematical Physical and Engineering Sciences*, *370*, 158–175.
6. Kappor, N. K., Nandy, B., & Majumdar, S. (2015). Dynamic allocation of sensor nodes in wireless sensor networks hosting multiple applications. *WiMOB2015*, December 2015.
7. Farooq, M. O., Sreenan, C. J., Brown, K. N., & Kunz, T. (2015). RPL-based routing protocol for multi-sink wireless sensor networks. *WiMOB2015*, December 2015.
8. Winter, T. (2011). RPL: IPv6 routing protocol for low-power lossy networks. *draft-ietf-roll-rpl-19*, September 2011.
9. Ko, J., Jeong, J., Park, J., Jun, J., Gnawali, O., & Paek, J. (2015). DualMOP-RPL: Supporting multiple of mode downward routing in a single RPL network. *ACM Transaction on Sensor Network (TOSN), 11*(2), may 2015.
10. Jeong, Y., Lee, S., Park, H., Yoo, H., & Kim, D. (2016). Hop-interval based decision of operational mode In RPL with multi-instance. *International Conference on Ubiquitous and Future Networks 2016(ICUFN)*, July 2016.
11. Ahmed, A., & Younis, M. (2015). Accurate shallow and deep water range estimation for underwater networks. *Global Communications Conference (GLOBECOM) 2015*, December 2015.
12. III, A. F. H., & Zorzi, M. (2007). Modeling the underwater acoustic channel in ns2. *Proc. NSTools '07*, October 2007.

Sungwon Lee received M.S. degree from Graduate School of Electrical Engineering and Computer Science, Kyungpook National University, Daegu, Korea in 2011. He is currently Ph.D. course student in School of Computer Science and Engineering, Kyungpook National University in Korea. His research interests are wireless mesh network, wireless sensor network, vehicular ad hoc network and underwater wireless sensor network.

Yonghwan Jeong received bachelor degree from Department of Computer Engineering, Yeungnam University, Daegu, Korea in 2014. He is currently master course student in School of Computer Science and Engineering, Kyungpook National University in Korea. His research interests are Internet of Things and wireless sensor network.

Eunbae Moon received bachelor degree from School of Computer Science and Engineer, Kyungpook National University, Daegu, Korea in 2015. He is currently master course student in School of Computer Science and Engineering, Kyungpook National University in Korea. His research interests are wireless sensor network, vehicular ad hoc network and underwater wireless sensor network.

Dongkyun Kim is a professor with the Department of Computer Engineering, Kyungpook National University, Daegu, Korea. He received the B.S. degree at Kyungpook National University. He pursued his M.S. and Ph.D. degrees at Seoul National University, Korea. He was a visiting researcher at Georgia Institute of Technology, Atlanta, GA, USA. He also performed a post-doctorate program at University of California, Santa Cruz. He has been a TPC member of several IEEE conferences. He received the best paper award from the Korean Federation of Science and Technology Societies, 2002. His research interests are Ad-Hoc network, sensor network, and wireless LAN, etc.