

Research Article

Avoiding Spurious Retransmission over Flooding-Based Routing Protocol for Underwater Sensor Networks

Sungwon Lee , Yeongjoon Bae , Muhammad Toaha Raza Khan , Junho Seo ,
and Dongkyun Kim 

School of Computer Science & Engineering, Kyungpook National University, Daegu, Republic of Korea

Correspondence should be addressed to Dongkyun Kim; dongkyun@knu.ac.kr

Received 2 April 2020; Revised 30 June 2020; Accepted 1 August 2020; Published 25 August 2020

Academic Editor: Hideyuki Takahashi

Copyright © 2020 Sungwon Lee et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

In underwater wireless sensor networks (UWSN), acoustic communication naturally introduces challenges such as long propagation delay and high packet loss. The flooding-based routing protocol can address these challenges with its multipath characteristics. As in flooding-based routing, due to multipath propagation mechanism, not only DATA but also ACK messages are transmitted through multiple routes however still some packet loss will degrade the performance. So, to provide high reliability of message delivery, an efficient retransmission mechanism is inevitable. Though, if the network uses conventional transport layer protocol such as TCP, it will suffer a spurious retransmission problem as TCP was originally not designed for the multipath environment. In this paper, we propose route discrimination for flooding-based routing to reduce spurious retransmission in UWSN to solve the limitation. The notion of ACK copies waiting time (ACWT) is utilized which is selectively updated based on the similarity of paths of transmission of ACK message copies. We also improved our previous solution that lacks flexibility to cope with dynamic link error characteristics. Through evaluation, we verified that our new scheme achieves the performance improvements of 14%~84% in terms of retransmission ratio compared to the previous research.

1. Introduction

Nowadays, research interest for IoT (Internet for Things) technology has been increased to provide Internet technology to various “Things.” With the support of the IoT techniques, new IoT applications are developed that can make human lives more comfortable. The range of the “Things” includes vehicular, mobile and sensor devices. Underwater sensor nodes are one of the promising “Things” because of the various underwater applications. Those applications include seawater temperature monitoring, underwater pollution monitoring, and coastal threat monitoring, developed to serve the ecosystem and human lives [1–6].

UWSNs have different characteristics from the terrestrial wireless sensor networks as they commonly use acoustic waves instead of radio waves. Due to the characteristics of the acoustic waves, underwater communication suffers from various limitations including slow propagation, high error rate [7–11]. To overcome these limitations and to achieve

high throughput, various protocols are newly proposed for UWSNs. Among them, flooding-based routing protocols are proposed to transmit data packets from source to the destination reliably. Unlike conventional routing protocols designed for the terrestrial networks such as ad hoc on-demand distance vector (AODV) [12], the flooding-based routing protocols have two advantages for error-prone and slow UWSNs [13]. Since the flooding-based routing protocols send multiple copies of a data packet along different paths, the original packet could be received at the sink even if few copies of the packets are lost. Moreover, the source node could send a data packet without the path establishment in advance. Detailed descriptions for the flooding-based routing protocols are introduced in Section 2.1.

Although, these routing protocols can improve the reliability but still lack far behind from meeting the IoUT (Internet of Underwater Things) application requirements. For example, in some IoUT applications, large and sensitive data (oil pipeline leakage image) is to be transmitted between the

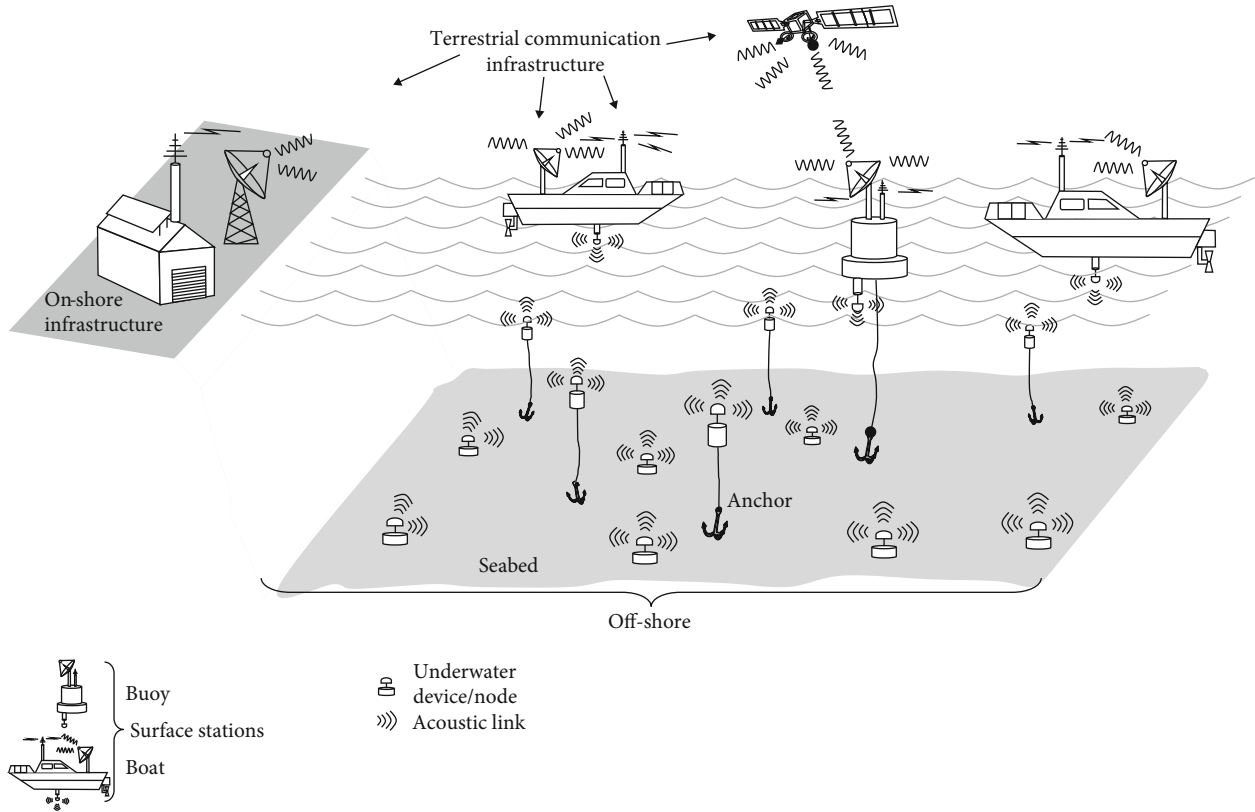


FIGURE 1: UWSN architecture.

sensor node and the sink node (Figure 1). Moreover, similar to software-defined networking architectures, if files which are used to upgrade the sensor node firmware should be transmitted, all sensor nodes have to receive these files and as even one bit of error is not allowed [14]. To support this packet transmission, reliable transport protocols which can detect the packet loss and support packet retransmission operation are required over the flooding-based routing protocols.

One of the most famous transport protocols which provides packet loss detection and retransmission operation is the transmission control protocol (TCP) [15]. In the TCP, the source set retransmission timeout (RTO) based on round trip time (RTT). If the source node cannot receive the acknowledgment (ACK) message from the destination node, it retransmits the packet [16]. Hence, the RTO calculation is one of the most important operations in TCP. If the calculated RTO is too long, throughput could be decreased because long time is required for retransmissions. On the other hand, if the RTO expires before ACK is received, a spurious retransmission problem could occur. The spurious retransmission wastes the bandwidth since the source node retransmits data packet which is successfully transmitted at the destination node. In the terrestrial networks, the source uses various RTO calculation algorithms which are well-operated over single-path routing protocols.

However, as mentioned above, IoUT devices use the flooding-based routing protocols which transmit packets (both of data and ACK) along multipaths. Over these routing

protocols, even though an ACK message is transmitted at the destination, the source node will receive multiple ACK messages which are flooded along different paths. If the traditional RTO calculation algorithms such as Khan's algorithm are implemented, the source updates RTT when the first ACK is received. Other ACKs will be ignored because the source node already knows that the destination node received the corresponding data packet. Therefore, the source node will set RTO to the time required for the first received ACK to arrive. When the source node uses this RTO for retransmission and the first ACK is lost during transmission, the source will perform packet retransmission (although another ACK messages which are transmitted along different paths) as described in Figure 2. Therefore, in order to avoid the spurious retransmission problem over the flooding-based routing protocol, the new RTO calculation which reflects another ACK messages which are received after the first ACK reception method is required.

In other words, RTO should be calculated in a way such that, for each data transmission, at least one of the many ACK packets transmitted over multiple paths can be received successfully. In our previous work [17], we proposed a technique that uses the similarity of paths from which packets have been sent to identify ACK packets that have a low probability of loss. The scheme also updates the RTO selectively where path similarity is defined in relationship with the same nodes that both paths share. However, in our previous work, a number of shared ancestor nodes among multiple paths are used only to determine whether to reflect the RTT of the

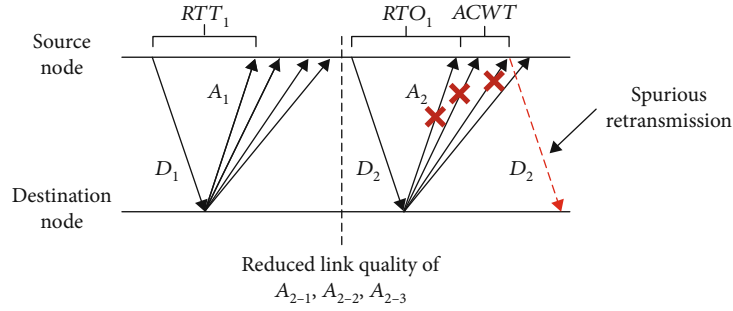


FIGURE 2: Spurious retransmission.

newly received ACK message or not. In other words, if the number of nodes shared by both paths is less than the threshold, the RTTs of all ACK messages are equally reflected. Due to this feature, a path which has poor link quality, or a large number of nodes could change the RTO significantly, thereby reducing throughput.

Hence, in this paper, we propose a technique to calculate the RTO that adapts to the current network situation by considering the information of each path and by applying path similarity analysis. We simulated and compare the proposed scheme using a C program with conventional RTO calculation mechanism [18] and previous work [17].

The rest of this paper is organized as follows. In Section 2, related works are discussed. Section 3 contains the detail description of the proposed techniques to solve the problems mentioned in this paper, while Section 4 analyses the performance of the proposed scheme through simulation comparison. Finally, Section 5 concludes our work.

2. Related Work

2.1. Flooding-Based Routing Protocols. As mentioned above, in flooding-based routing protocols, nodes forward every received packet without defining a separate path [19–21]. In these routing protocols, each node establishes a flooding area that directs its location information to the sink node. Then, whenever each node transmits a data packet, information related to the flooding area is recorded in the header of the data packet and broadcasted. If the neighbouring nodes of the transmitting nodes receive this packet, it confirms that they are included in the flooding area. If it is included in the flooding area, it updates the header information in the same way and broadcasts the packet. If it is not included in the flooding area, discard the packet.

Among various flooding-based routing protocols, the depth-based routing (DBR) protocol [20] is a representative flooding-based routing protocol. In the DBR, when the sender node transmits a packet, its depth information is included in the routing header. On arrival of a packet, nodes located shallower than the sender node operated as relays and retransmit it towards the surface. Hence, the DBR not only allows packets to be delivered away from the sink but also has a problem that the size of the flooding area is always fixed regardless of link quality.

To overcome this problem, in the Directional Flooding-based Routing (DFR) [21] protocol, the size of the flooding area is adjusted by the average link quality. The transmitting node measures the average ETX (Expected Transmission Count) with neighbouring nodes. Each time a packet is transmitted, the sender node records its location information and the base angle, which increases as the average ETX value becomes lower, in a data packet and broadcasts it. The receiving node computes the angle between the vector connecting itself and the sender node and the vector connecting itself and the sink node. If the calculated angle is less than the base angle included in the packet, the node transmits the packet. Therefore, the worse the link quality, the more nodes participate in the packet transmission and improve the reliability. Conversely, the better the link quality, the fewer nodes can participate in packet transmission and save network resources [22].

2.2. Retransmission Technique. The transport layer protocols provide an end-to-end connection between a sender and a receiver. These protocols are classified into two types: user datagram protocol (UDP) [23] and transmission control protocol (TCP) [15]. UDP has a very simple transmission method without any guarantee of successful delivery. In TCP, it confirms sender nodes regarding successful packet delivery. Since in UDP, the transmission overhead is small as compared to TCP, it is preferable in real-time applications like voice over internet protocol (VoIP) and real-time streaming. TCP, on the other hand, establishes a connection between a sender and a receiver before data transmission and transmits data reliably by providing functions such as acknowledgment, retransmission, and message reordering in data transmission. Therefore, it is mainly used in the web, e-mail transmission, or file transfer application where data must be normally transmitted. The TCP has multiple versions such as TCP-Tahoe, TCP-Reno, TCP-SACK [24], TCP-New Reno [25], TCP-Vegas [26], and TCP-Westwood [27] are available. Each version targets different performance matrix for operation such as congestion control, packet transmission rate control, packet loss, and duplication problems while operating following the core TCP.

CoAP (Constrained Application Protocol) is a lightweight transport protocol for constrained nodes (CN). Taking into account the limited computing power of CNs, CoAP is designed to work over UDP instead of TCP.

However, since UDP is a protocol that cannot provide end-to-end reliability, CoAP can use the ACK message in the form of a UDP packet. A packet requesting transmission of an ACK packet is defined as a Confirmable message, and a packet not requesting transmission of an ACK packet is defined as a nonconfirmable message (the type of this message is determined by the upper layer such as application). If the source node sends a confirmable message, it sets a retransmission timer (RTO) likely TCP. If the corresponding ACK message is not received before the expiration of the retransmission timer (due to delay or loss), the source node retransmits the confirmable message. Hence, appropriate RTO calculation is needed for not only TCP but also CoAP.

3. Proposed Scheme

In this section, after defining the problem, the proposed scheme is introduced. In the proposed scheme, we assume that DFR and Khan's algorithm in TCP-Reno are used for flooding-based routing protocol and retransmission algorithm, respectively. This is because DFR is a routing protocol that not only improves reliability but also saves network resources, and Khan's algorithm is a proven retransmission algorithm in various networks such as the Internet.

3.1. Problem Definition. In our previous work [17], ACWT is selectively updated considering the path similarity between received ACK copies. ACWT is calculated systematically to incorporate randomly arranged network topology while reducing unnecessary retransmission.

However, in our previous work, the maximum number of ACK copies (ACK_{max}) is set as a fixed value. We assumed that the quality of packet transmission is constant for all links, and only the difference in arrival time of ACK copies is considered. However, in real underwater networks, link qualities of each path changes dynamically. The number of ACK messages received for each segment of one flow changes. ACWT is calculated based on the received amount of the previous ACK message makes it impossible to accurately set the retransmission timer of this ACK message. So, we utilize the ratio of the successfully delivered packet over the total number of packets transmitted between two nodes as a parameter to depict the link quality between the two nodes.

If the ACWT is calculated using the fixed ACK_{max} without considering the transmission success rate of the ACK copies transmitted to the source node, it cannot cope with the dynamic changes in the underwater wireless communication channel. One of the scenarios is depicted in Figure 2 in which a sudden decrease in the link quality of the path leads to spurious retransmission. For ACWT calculation, ACK_{max} is continuously updated in the new RTO calculation technique based on the instantaneous network conditions.

In this paper, we propose a method in which the source nodes manage the information of a distinguished path by using bitmap operation while keeping a separate table. In this way, ACK copies transmitted to the source node form different paths. For every individual path based on the average RTT calculation, the change in the packet transmission suc-

cess rate is monitored. As a result, the source node continuously manages the average RTT and the average packet transmission success rate of different paths formed between itself and the destination node. The source node adjusts the ACK_{max} flexibly based on the average RTT and the average packet transmission success rate.

3.2. Proposed Solution. In the common IoUT, if nodes have low remaining energy or have a high computing demand sleep to save energy. Otherwise, if nodes want to communicate with each other, they are awake and start idle listening to receive to transmit or relay data packets. The sender node that intends to send some collected data is termed as the source node. Every sensor node creates and stores a separate path information table for the path information distinguished through the bit map operation and continuously manages this information for the packet transmission. Every node manages information regarding travelled section (TS) field representing the corresponding path followed by the packet, an average RTT of ACK copies received through the corresponding path and an average transmission success rate of ACK copies. By managing this information, the source node can know how many routes are formed between the node itself and the destination node, and how much the expected success rate of ACK copy is obtained when the value of ACK_{max} is changed. Algorithm 1 illustrates the pseudocode of the path distribution-based information update along with ACWT calculation.

First, when a new ACK copy (ACK_{recv}) is received, the similarity between the TS fields stored in the current path table and the TS field (TS_{recv}) of the corresponding ACK copy is calculated to determine whether a similar path is already managed. If it is determined that the TS_{recv} is similar to the TS field stored in the route table, it is determined that the ACK copy is transmitted through the same route, and the average RTT of the route is updated by the RTT obtained through ACK_{recv} . On the other hand, if TS_{recv} is not similar to any TS field stored in the current path table, it is determined that the ACK copy is transmitted through the new path. The Num_{path} is incremented by 1, and TS_{recv} is stored as the TS field of the new path in the path table. Then, the source node calculates the expected packet transmission success rate of each path based on the number of times that the source node has transmitted the data packet and the number of ACK copies received through each path. When the calculation of the average RTT and the average expected packet transmission success rate of all paths is completed through this operation, the ACK_{max} is calculated.

3.3. Calculation of the ACK_{max} . In algorithm 1, the utility function $U(i)$ is one of the most important factors to determine a new round trip time measured by the i th ACK (namely $RTT[i]$) is an appropriate value to update of ACWT. To calculate the $U(i)$, the equations (1) and (2) are used as follows.

$$V(i) = \omega time \times \left(2 - \frac{RTT[i] - RTT[i-1]}{RTT[0]} \right). \quad (1)$$

```

1:receive new ACK(ACKrecv)
2:decide ← true
3:for i=0; i < Numpath; i++ do
4:  if PS(TSpath[i], TSrecv) > PSthresh then
5:    decide ← false, update RTTpath[i]
6:  end if
7:end for
8:if (decide) then
9:  Numpath ← Numpath + 1, TSpath [Numpath] ← TSrecv
10:end if
11:for i=0; i < Numpath; i++ do
12:  update RTTpath[i]
13:end for
14:for i=0; i < Numpath; i++ do
15:  if U(i) > Uthresh then
16:    ACK ← i + 1
17:  end if
18:for i=0; i < ACKmax; i++ do
19:  calculate Tack and update ACWT with RTTpath[i]
20:end for

```

ALGORITHM 1. Path distinction-based information update and ACWT calculation.

In this equation (1), $V(i)$ and ω time are a vector value of $RTT[i]$ and a weight factor for $V(i)$, respectively. After calculating the $V(i)$, the $U(i)$ is calculated as follows.

$$U(i) = V(i) + \omega \text{prob} \times (1 - (1 - PDR[i]) \cdots (1 - PDR[0])) - PDR[0]. \quad (2)$$

In this equation (2), $PDR[i]$ and ω prob are an expected packet delivery ratio of i th path and a weight factor for $U(i)$, respectively, when considering the i th path, compared with the case considered only up to the $(i-1)$ th path. The larger the difference between the i th ACK arrival path and the $(i-1)$ th arriving RTT path, the smaller the value of the term itself. Instead of using the absolute difference between these RTTs, that difference is normalized by the $RTT_{path}[0]$ to be proportional to the RTT of the first arriving ACK. Therefore, the first term has a larger value as the difference between the i th ACK arrival path and the $(i-1)$ th arrival path RTT becomes smaller. The second term is used for calculating the difference in the expected PDR when considering the i th path, compared with the case considered only up to the $(i-1)$ th path. The larger the difference between the expected packet transmission success rate when considering the i th path and the expected packet transmission success rate when considering up to the $(i-1)$ path, the greater the value of the term itself.

Therefore, the equation (2) calculates the appropriate value of $U(i)$ how the i th path to be reflected in the ACWT update through the time difference that occurs when considering the i th path and the increase of the packet transmission success rate. In addition, the equation compares the value

with a threshold value (U_{thresh}) and finally obtains the optimal performance in the current path situation.

In this paper, we set these 3 network parameters (ω time, ω prob, and U_{thresh}) to 0.25, 0.5, and 0.45, respectively. These values are measured by repeated simulations.

3.4. Demonstration. The functioning of the proposed technique will be exemplified by a simple case depicted in Figure 3. As shown in the figure, four paths exist between the source node and the sink node.

In this case, the average RTT and link quality for each path in (a) and (b) are changed as shown in Table 1.

First, RTT increases gradually in the order of Path0, Path1, Path2, and Path3 in (a) and (b). In (a) situation, the link quality of Path0, Path1, and Path2 was 85%, and Path3 alone had a relatively low link quality of 40%. In each situation, $U(i)$ is calculated as shown in Table 2.

Therefore, in (a), $U(1)$ is calculated to be higher than U_{thresh} and the optimal ACK_{max} is calculated as 2. However, in (b), when the RTT of Path0 increases by 0.1 seconds and the transmission quality drops to 40%, the optimal ACK_{max} is calculated to be 3 because $U(2)$ is larger than U_{thresh} . Therefore, if the transmission quality of Path0 drops, it can be confirmed that in addition to Path1, ACK can be further awaited by reflecting the addition of Path2, which was not previously reflected, to the calculation of ACWT.

4. Performance Evaluation

4.1. Simulation Environments. To evaluate the performance of our scheme, we used NS-2.34 simulator and underwater acoustic signals propagation model. Other simulation parameters were set as shown in Table 3. Moreover, to evaluate the contribution of the proposed scheme only, we

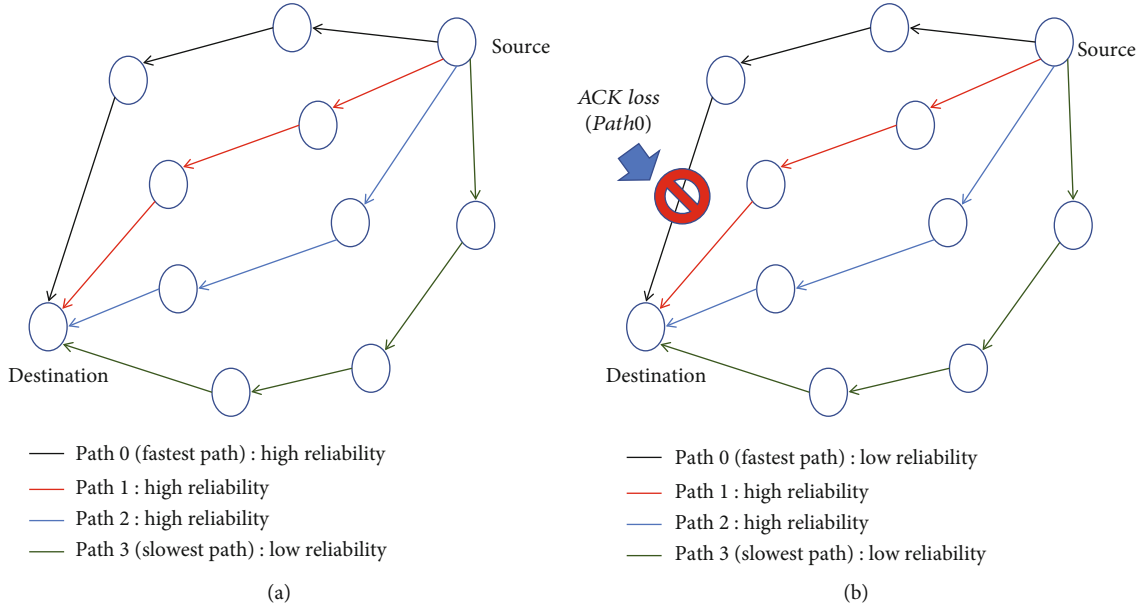


FIGURE 3: Example Scenario.

TABLE 1: RTT and link quality.

Path	RTT	(a)	RTT	(b)
		Link quality		Link quality
Path0	5.0 sec	85%	5.1 sec	40%
Path1	5.3 sec	85%	5.3 sec	85%
Path2	6.5 sec	85%	6.5 sec	85%
Path3	7.6 sec	40%	7.6 sec	40%

TABLE 2: $U(i)$ values.

$U(i)$	(a)	(b)
$U(1)$	0.548	0.745
$U(2)$	0.449	0.479
$U(3)$	0.445	0.448

TABLE 3: Simulation parameters.

Parameter	Value
Transmission range	250 meters
Routing protocol	DBR [14]
ACK_{max}	2
PS_{thresh}	0.25
ω_{time}	0.25
ω_{prob}	0.5
U_{thresh}	0.45

selected the original TCP instead of other TCP versions. If the proposed technique is applied to other TCPs, it is not possible to verify whether the improved performance is due to the proposed technique or another technique.

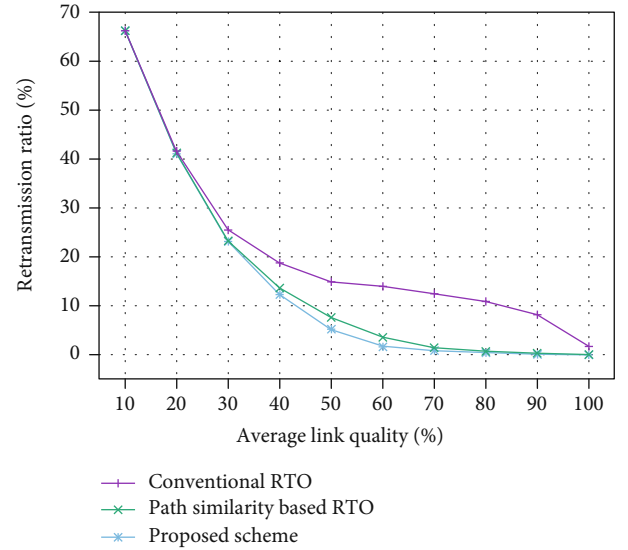


FIGURE 4: Retransmission ratio in uniform LQ scenario.

The size of the network is set to 1600×800 m. The whole network is divided into 8 rows and 4 columns resulting in a total of 32 sections. The node nearest to the point $(0, 800)$ is designated as the source node, and the node closest to the point $(1600, 0)$ is designated as the sink node. The nodes were relocated randomly in each experiment run. In addition, DBR is used as the routing protocol.

4.2. Performance Factor. We assume two different situations for performance evaluation. First, the link quality (LQ) of the entire network is uniform. This is to evaluate the performance of the proposed technique for the comparison with the existing method (path similarity-based RTO). In second, the link quality of the entire network

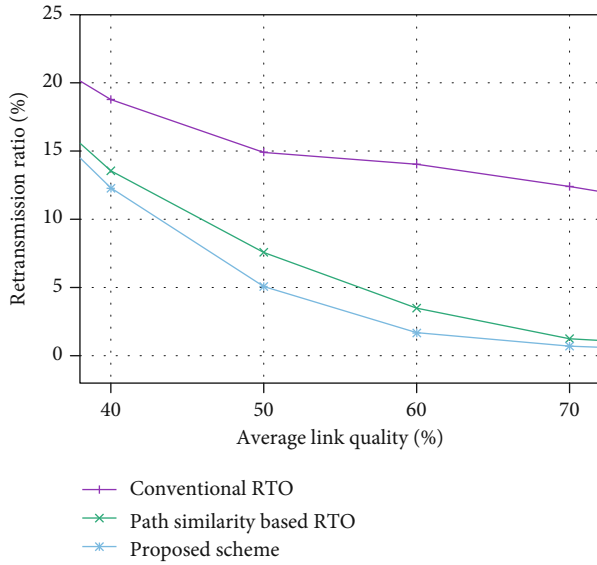


FIGURE 5: Retransmission ratio in uniform LQ scenario (40~70 nodes).

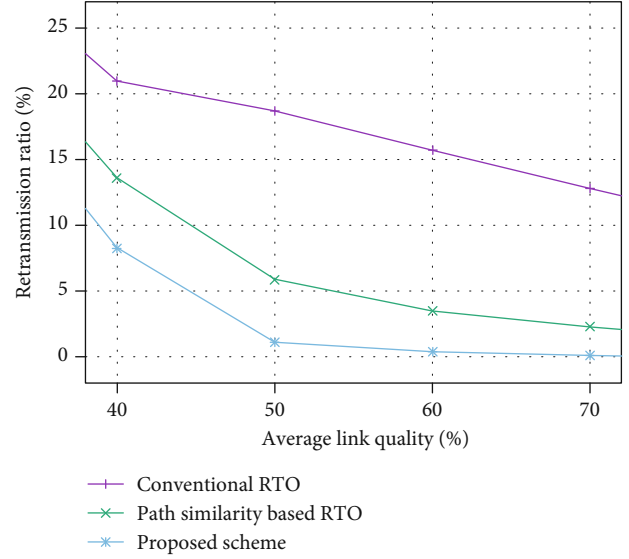


FIGURE 7: Retransmission ratio in different LQ scenario (40~70 nodes).

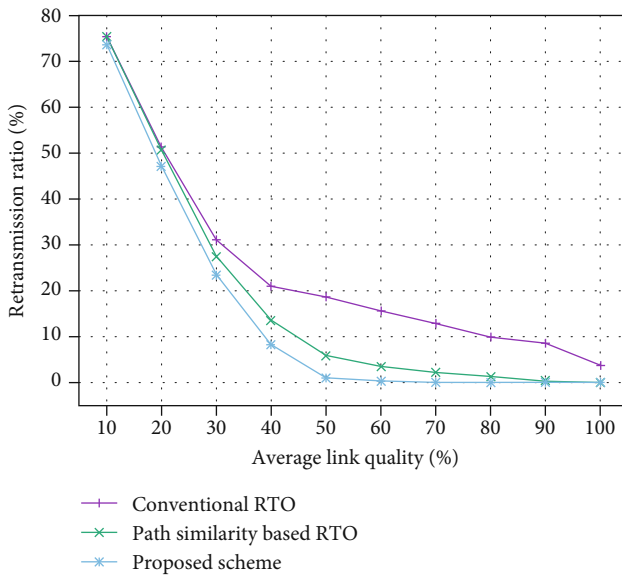


FIGURE 6: Retransmission ratio in different LQ scenario.

is nonuniform that varies from time to time. We can evaluate the performance of the proposed method by comparing it with that of the existing method. The evaluation parameters include “retransmission ratio” that is the number of packets retransmitted by the source node divided by the total number of packets transmitted by the source node, where “average link quality” indicates the average transmission success rate of the entire network.

4.3. Performance Comparison in Uniform Link Quality of Network. In this experiment, we first set the average link quality of all the links uniformly, and experimentally examined how the retransmission ratio changes according to the change of the average link quality. As shown in Figure 4,

the retransmission ratio increases as the average link quality decreases regardless of the RTO calculation technique used. Especially, it can be confirmed that the proposed algorithm based on the path similarity-based RTO that can additionally wait for the ACK copy and the proposed scheme achieves a lower retransmission ratio than the conventional RTO calculation method regardless of the change in link quality.

The performance difference between the two schemes with ACWT and the existing RTO calculation technique starts from when the average link quality is 100% and gradually increases from 70% until the average link quality becomes lower than 60%. All converge to one point. This is because, when the average link quality is high, it is possible to reduce the retransmission by using additional received ACK copies. As the link quality decreases, ACK copies are also lost as a result successful ACK copy cannot arrive.

However, as shown in Figure 5, the difference in performance between the path similarity-based RTO calculation technique and the proposed technique is conspicuous in the interval of 70%~40% of the average link quality. This is because the fixed ACK_{max} , the proposed method in this paper seems to be due to the fact that the expected rate of packet transmission is low for all the paths at present, so that the ACK_{max} is made large and waiting for the maximum number of ACK copies.

4.4. Performance Comparison with Different Link Qualities. In this experiment, the link quality of the entire network is not uniform, and the link quality of each link is constantly changed. As can be seen in Figures 6 and 7, the retransmission ratios are increased as the average link quality decreases regardless of the RTO calculation technique used in this experiment.

However, unlike the experimental results in Figure 6, it can be seen that the performance difference between the proposed method and the path similarity-based RTO calculation

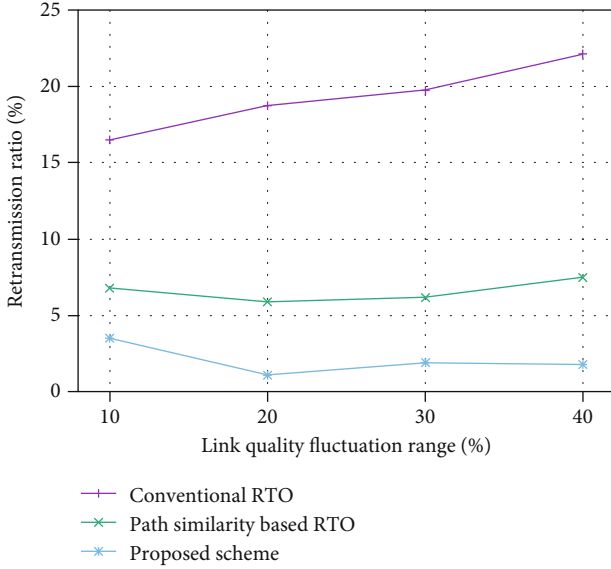


FIGURE 8: Retransmission ratio in LQ fluctuation scenario.

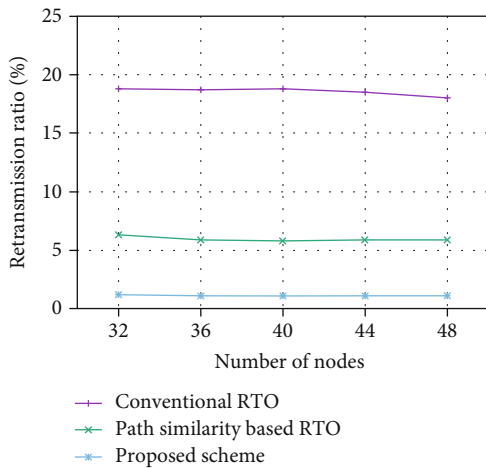


FIGURE 9: Retransmission ratio in LQ fluctuation scenario (32~48 nodes).

technique is noticeable regardless of the average link quality. It monitors the state of multipath formed between the source node and the destination node, measures the RTT and the expected transmission success rate for each path, and calculates the ACK_{max} value that can achieve the optimum performance based on this. This is a result of proving that the operation is effective in the same environment as the experiment in which the link quality fluctuates. It is shown that the proposed method improves performance by at least 14% and up to 94% compared with the existing similarity-based RTO calculation method.

Also, in this experiment, the performance difference is conspicuous in the case where the average link quality is higher than a certain level, but the performance difference is decreased as the average link quality is lowered. This is because, as mentioned in the previous experimental results,

when the average link quality is high, it is possible to reduce the retransmission by using additional received ACK copies. However, as the link quality is lowered, ACK copies are also lost and waiting for ACK copies. This is because it is not possible to successfully receive an ACK copy.

After fixing the average link quality at 50%, we experimented the change of retransmission ratio according to the link quality fluctuation width by changing the fluctuation range of the link quality of the whole network. As shown in Figure 8, the fluctuation width is changed from 10% to 40%, and the retransmission ratio is generally increased as the variation of the link quality is larger regardless of the RTO calculation method used. Regardless of the fluctuation of the link quality, the proposed scheme achieves a lower retransmission ratio than the other two schemes.

Also, the retransmission ratio increases linearly with the increase of the link quality fluctuation in the case of the conventional RTO calculation technique. However, the path similarity-based RTO technique and the proposed technique do not. This seems to be caused by the ACWT, which is commonly used by both schemes. It is because the link quality of the multipath is changed differently, and k and TACK are not calculated stably.

Finally, after changing the average link quality to 50% and the link quality fluctuation to 20%, we experimented with changing the number of nodes in the network and changing the retransmission ratio. As can be seen in Figure 9, the number of nodes varied from 32 to 48. The results show that the proposed method shows the best performance irrespective of the number of nodes. In particular, in the case of the conventional RTO calculation technique, a slight improvement in performance was confirmed as the number of nodes increased. However, in the case of the other two techniques, no noticeable performance change was observed even when the number of nodes increased. This is because the ACWT is calculated through the routes with a similar degree of similarity even if more multipath are formed as the number of nodes increases.

5. Conclusion

In this paper, the information of the multipaths established between the source and the destination node is utilized over traditional TCP to avoid spurious retransmission in UWSN while using flooding-based routing. In the proposed scheme, the source node retransmission timeout (RTO) calculation measures an expected number of ACK messages (ACK_{max}) based on the reliabilities of paths. Using ACK_{max} , the proposed scheme reduces the unnecessary retransmission. While waiting for the ACK of the maximum number of paths, ACK_{max} is increased, as the success rate of ACK transmission of each path is lower. If ACK-transmission rate is high, ACK_{max} is decreased enabling faster retransmission. Through performance evaluation via multiple simulations, the proposed scheme achieves 14%~84% improved performance in the form of retransmission ratio.

In future work, we will implement our proposed scheme to real underwater communication devices such as Benthos modem. Based on the real test result, we will

improve our proposed scheme. Moreover, we will work on an underwater congestion control method utilizing the concept of consistency.

Data Availability

All data used to support the findings of the study is included within the article.

Conflicts of Interest

None of the authors have any conflicts of interest.

Acknowledgments

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] S. Mustafa, M. A. Khan, and S. A. Malik, "Design of underwater sensor networks for water quality monitoring," *World Applied Sciences Journal*, vol. 17, no. 11, 2012.
- [2] F. Raimondi, M. Trapanese, P. Martorana, V. Franzitta, and V. A. Di Dio, "Security and surveillance application of the innovative monitoring underwater buoy systems (MUnBuS) on the protected marine area (AMP) of capo gallo (PA-IT)," in *OCEANS'15 MTS/IEEE*, Washington, 2015.
- [3] K. Lam, R. S. Bradbeer, P. K. Shin, K. K. Ku, and P. Hodgson, "Application of a real-time underwater surveillance camera in monitoring of fish assemblages on a shallow coral communities in a marine park," in *IEEE OCEANS*, Vancouver, Canada, 2007.
- [4] C. C. Kao, Y. S. Lin, G. D. Wu, and C. J. Huang, "A comprehensive study on the internet of underwater things: applications, challenges, and channel models," *Sensors*, vol. 17, no. 7, article 1477, 2017.
- [5] G. Xing, Y. Chen, L. He et al., "Energy consumption in relay underwater acoustic sensor networks for NDN," *IEEE Access*, vol. 7, pp. 42694–42702, 2019.
- [6] E. Liou, C. Kao, C. Chang, Y. Lin, and C. Huang, "Internet of underwater things: challenges and routing protocols," in *Proc. of 2018 IEEE International Conference on Applied System Invention (ICASI)*, pp. 1171–1174, Chiba, April 2018.
- [7] M.-S. Kim, S.-W. Lee, M. T. R. Khan et al., "A new CoAP congestion control scheme using message loss feedback for IoUT," in *Proc. of the 34th ACM/SIGAPP Symposium on Applied Computing (SAC '19)*, pp. 2385–2390, Limassol, Cyprus, April 2019.
- [8] J. Seo, S. Lee, M. T. R. Khan, and D. Kim, "A new CoAP congestion control scheme considering strong and weak RTT for IoUT," in *Proceedings of the 35th Annual ACM Symposium on Applied Computing (SAC '20)*. Association for Computing Machinery, pp. 2158–2162, New York, NY, USA, 2020.
- [9] G. Han, X. Long, C. Zhu, M. Guizani, and W. Zhang, "A high-availability data collection scheme based on multi-AUVs for underwater sensor networks," *IEEE Transactions on Mobile Computing*, vol. 19, no. 5, pp. 1010–1022, 2020.
- [10] H. U. Yildiz, "Maximization of underwater sensor networks lifetime via fountain codes," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 8, pp. 4602–4613, 2019.
- [11] I. F. Akyildiz, P. Wang, and Z. Sun, "Realizing underwater communication through magnetic induction," *IEEE Communications Magazine*, vol. 53, no. 11, pp. 42–48, 2015.
- [12] C. Perkins, E. Belding-Royer, and S. Das, "Ad hoc on-demand distance vector (AODV) routing," Technical report, 2003.
- [13] S. LEE and D. KIM, "aCK loss-aware RTO calculation algorithm over flooding-based routing protocols for UWSNs," *IEICE Transactions on Information and Systems*, vol. E97.d, no. 11, pp. 2967–2970, 2014.
- [14] J. Postel and J. Reynolds, "File transfer protocol," Technical report, 1985.
- [15] J. Postel, "Transmission control protocol," Technical report, 1981.
- [16] M. A. J. C. Paxson Vern and M. Sargent, "Computing TCP's retransmission timer," Technical report, 2011.
- [17] Y. Bae, S. Lee, Y. Jeong, and D. Kim, "Path similarity based spurious retransmission minimization over flooding based routing in UWSN," in *2018 Tenth International Conference on Ubiquitous and Future Networks (ICUFN)*, Prague, Czech Republic, July 2018.
- [18] V. Jacobson, "Congestion avoidance and control," *ACM SIGCOMM Computer Communication Review*, vol. 18, no. 4, pp. 314–329, 1988.
- [19] J. H. C. Xie Peng and L. Lao, *VBF: Vector-Based Forwarding Protocol for Underwater Sensor Networks*, International conference on research in networking, 2006.
- [20] Z. J. S. Yan Hai and J. H. Cui, *DBR: Depth-Based Routing for Underwater Sensor Networks*, International conference on research in networking, 2008.
- [21] D. Shin, D. Hwang, and D. Kim, "DFR: an efficient directional flooding-based routing protocol in underwater sensor networks," *Wireless Communications and Mobile Computing*, vol. 12, no. 17, 1527 pages, 2012.
- [22] D. S. J. De Couto, "High-throughput routing for multi-hop wireless networks," PhD thesis at Massachusetts Institute of Technology, 2004.
- [23] J. Postel, "User datagram protocol," Technical report, 1980.
- [24] K. Fall and S. Floyd, "Simulation-based comparisons of Tahoe, Reno and SACK TCP," *ACM SIGCOMM Computer Communication Review*, vol. 26, no. 3, pp. 5–21, 1996.
- [25] T. H. Floyd Sally and A. Gurtov, *The NewReno Modification to TCP's Fast Recovery Algorithm*, Technical report, 2004.
- [26] L. S. Brakmo and L. L. Peterson, "TCP Vegas: end to end congestion avoidance on a global internet," *IEEE Journal on Selected Areas in Communications*, vol. 13, no. 8, pp. 1465–1480, 1995.
- [27] S. Mascolo and R. Wang, "TCP westwood: bandwidth estimation for enhanced transport over wireless links," in *Proceedings of the 7th annual international conference on Mobile computing and networking*, New York, USA, 2001.