Available *online* at : http://jitce.fti.unand.ac.id/
JITCE (Journal of Information Technology and Computer Engineering)
| ISSN (Online) 2599-1663 |

Research article

ADBR: Accelerated Depth-Based Routing for Underwater Sensor Networks

Abdolreza Andalib[1], Fariba Abbasi Nia[2], Mojtaba Jamshidi[3], Abdusalam Abdulla Shaltooki [4], Mehdi Esnaashari[5]

[1] *Department of Computer Engineering, Borujerd Branch, Islamic Azad University, Borujerd, Iran*
[2] *Department of Computer Engineering, Kermanshah Branch, Islamic Azad University, Kermanshah, Iran*
[3,4] *Department of Information Technology, University of Human Development, Sulaymaniyah, Iraq*
[5] *Faculty of Electrical and Computer Engineering, K. N. Toosi University of Technology, Tehran, Iran*

ARTICLE INFORMATION

Received: March 29th, 2021
Revised: March 3rd, 2022
Available online: March 31st,2022

KEYWORDS

underwater wireless sensor networks, routing, DBR algorithm, accelerated routine

CORRESPONDENCE

Phone: +989186864571

E-mail: jamshidi.mojtaba@gmail.com

A B S T R A C T

It is challenging to propose an efficient routing algorithm for Underwater Wireless Sensor Networks (UWSNs) in terms of packet delivery ratio, end-to-end delay of packet delivery from the source to the destination, and energy consumption. The reasons of that are UWSNs have unique characteristics (e.g. using acoustic channels instead of radio channels for communications), and they have dynamic topology due to the movement of the sensor by the water flow. Depth-Based Routing (DBR) considers one of the well-known algorithms in this context. DBR is a very simple algorithm; however, it is inefficient in terms of packet delivery rate, end-to-end delay, and energy consumption. This study we developed DBR by adding an accelerated routine to it to improve its efficiency, the proposed algorithm; called Accelerated Depth-Based Routing (ADBR). In ADBR, a simple probabilistic mechanism is used to accelerate packet forwarding and provide more multi-path to the destination. In ADBR, each node immediately delivers received packet to the destination with a probability of $0< P_{f}. \leq 1$ and follows the DBR routine with a probability of $1 - P_{f}$. The performance of ADBR is evaluated via a set of experiments by using J-SIM simulator. Experimental results indicate the superiority of the ADBR over the DBR algorithm.

INTRODUCTION

Wireless sensor networks (WSNs) have extensive applications in many terrestrial domains, including urban services, smart city, Internet of Things, environment, industry, historical buildings, border surveillance, etc. Recently, WSNs have had an increasing application in underwater domains, such sensor networks, particularly built for underwater environments, are called underwater wireless sensor networks (UWSN). Routing and packet delivery from the source to the destination is one of the most important research issues in UWSNs, which has been the focus of many researchers [1-3].

On the other hand, according to their special characteristics, it is challenging to design a scalable and efficient routing algorithm for UWSNs. Some of the most important challenges include [4-7]:

I. Radio communications in underwater environments does not have a good performance due to its rapid attenuation. Therefore, acoustic communications are usually used underwater. However, acoustic channels mostly have low bandwidth and long propagation delay. Therefore, routing algorithms, which require high bandwidth or have high end-to-end delay, are not a good option.

II. These networks have very dynamic topologies, since sensor nodes are continuously moving by the water flow (except some sensors at the surface or depth of the water, which are stationary). Therefore, routing algorithms for terrestrial sensor networks (mostly with stationary topologies or low mobility) cannot be used for UWSNs.

III. The final reason, which is related to all sensor networks, including terrestrial and underwater networks, is the limited energy of sensor nodes, which makes it inappropriate to employ routing algorithms with high communication overhead.

So far, many algorithms [8-22] have been proposed for routing in UWSNs, DBR is one of the most well-known routing algorithms of them [8]. DBR is a very simple depth-based algorithm, which does not require node positioning. This algorithm only utilizes the nodes' depth in the water for packet routing. In the next section, we will discuss more details about this algorithm.

https://doi.org/10.25077/jitce.6.01.19-28.2022

Generally, DBR is not efficient in terms of packet delivery rate, end-to-end delay, and energy consumption. Therefore, in this study, we add an accelerated routine to the basic DBR algorithm to improve its efficiency. We used a simple probabilistic mechanism to accelerate packet transmission and increase the number of paths to the destination. Thus, the performance of DBR is enhanced to a great extent.

The rest of this paper is organized as follows. Section 2 describes pervious works. Section 3 explains the proposed algorithm. Section 4 evaluates the performance of the proposed algorithm and finally, Section 5 presents conclusions.

RELATED WORK

Vector-Based Forwarding (VBF) [9] is the first routing algorithm proposed for UWSNs. VBF assumes that each node is aware of its position. Moreover, the position of source, sink, and sender nodes are embedded in the packet. The main notion of this algorithm is using a virtual routing pipe in which the source-to-sink vector is its pivot (pipe) and W is its radius as shown in Figure 1. Parameter W is a predefined threshold. If a node is placed in this pipe, it forwards the packet from the source to the destination.

More specifically, each intermediate node u, which receives the data from the source, first calculates its distance from the source-to-sink vector and forwards the packet if this distance is smaller or equal to the pipe's radius. Otherwise, node u drops the packet. This algorithm defines a virtual pipe for each source node. Generally, the drawbacks of this algorithm include: 1- each node requires determining its location at each given moment of network lifetime, which (positioning) is a challenging and costly. 2- packet delivery rate of this algorithm is very low for sparse networks, since it is possible that the pipe does not have the necessary nodes to forward the packets form the source to the sink; whereas, there may be a path to forward it outside the pipe. 3- due to the single-vector source-to-sink design, the radius threshold, that is W, greatly affects the performance of the algorithm (as mentioned in [9]).

Hop-by-Hop Vector-Based Forwarding (HH-VBF) [10] is another algorithm, which was proposed to eliminate the weaknesses of VBF. This algorithm also uses the same concept of routing vector proposed in VBF; however, instead of a single virtual pipe form the source to the sink, HH-VBF employs different virtual pipes at each point of packet transmission (intermediate nodes). Figure 2 presents how this algorithm creates paths. Similar to VBF, one of the important drawbacks of this algorithm is that it requires periodically determine the location of all nodes during network lifetime.

Reliable and Energy Balanced Algorithm Routing (REBAR) [11] is a position-based routing algorithm, which focuses on the three important issues in UWSNs, that is, energy consumption, packet delivery rate, and the void problem. This algorithm first uses a sphere energy depletion model to analyze nodes' energy consumption in UWSNs. It is then extended to consider node mobility in UWSNs and assumes that node mobility is a positive factor, which can be effective in balanced energy depletion in the network and increasing network lifetime. In REBAR, using geographical information, nodes only broadcast packets in a certain range between the source and sink nodes. It means that it does not employ network level broadcast, which consumes a large amount of energy.

Focused Beam Routing (FBR) [12] is a location and effective energy based routing algorithm, which assumes that each node is aware of its location and the location of the destination node [13]. Therefore, this algorithm does not require accurate information of the destination position. Moreover, variable transmission levels are used to send data packets. High node mobility and acoustic environment conditions are two important factors of packet loss in the routing process and thus, reducing network reliability. Directional Flooding-based Routing (DFR) [14] focused on this issue and considered link quality in its packet forwarding strategy. This algorithm also assumes to have access to geographical information. That is, all nodes are aware of the location of themselves, single-hop neighbors, and the sink. Moreover, each node can assess its link quality with its neighbors.

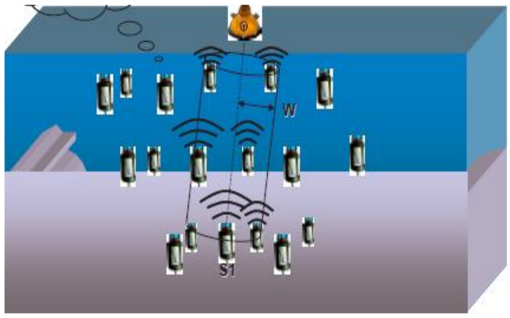

Figure 1. A high-level view of VBF algorithm [9].

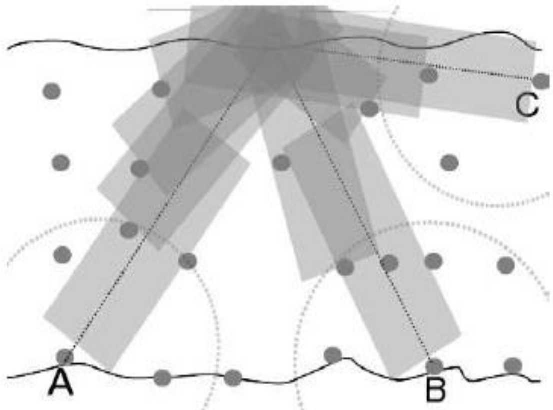

Figure 2. HH-VBF with per-hop vector computing [10].

A clustering algorithm based on the geographical location of sensor nodes in the 3D hierarchical architecture is proposed for UWSNs [15]. This algorithm divides the entire network into 3D grids. Data communications consists of three phases: 1-initialization selects the cluster-head, 2- data collection in which the data is sent from nodes in the cluster to the cluster-head. 3-transmission phase, collected data by each cluster-head send to the base station.

An Autonomous Underwater Vehicle (AUV) aided Efficient Data-Gathering (AEDG) [16] algorithm employed an AUV for data collection and used a shortest path tree algorithm. This

 https://doi.org/10.25077/jitce.6.01.19-28.2022

algorithm used four types of nodes; member nodes, gateway nodes, AUV, and sink. Two multi-path protocols called Greedy Geographic Forwarding based on Geospatial Division (GGFGD) and Geographic Forwarding based on Geospatial Division (GFGD) [17] were proposed which they consist of two phases, choosing the next target small cube, and choosing the next hop node in the target small cube.

Delay-Sensitive Depth-Based Routing (DSDBR) [18] is a depth-based routing algorithm, which formulate delay-efficient priority factors and delay-sensitive holding time to decreases end-to-end delay with a small decrease in network throughput. This algorithm also uses an optimal weight function for the computation of transmission loss and speed of received signal. Javaid et al., [19] proposed three chain-based routing algorithms for application-oriented cylindrical networks. In these algorithms, after finding local optimum paths in separate chains, global optimum paths through their interconnection will be founded. An Ultrasonic Frog Calling Algorithm (UFCA) [20] was proposed, in which the process of selecting relay nodes to forward the data packet in this algorithm is similar to that of calling behavior of ultrasonic frog for mating.

Multi-population Firefly Algorithm (MFA) [21] is an optimization based approach that employ three types of fireflies and their coordination rules in order to choose an adaptable routing path considering the data correlation and their sampling rate in various sensor nodes. A three-level propagation mechanism [22] has proposed to deliver packets from the source nodes to the sink nodes in UWSNs. Reliable and Energy Efficient Routing Algorithm (REERA) [23] is proposed to eliminate the overhead of flooding-based routing algorithm by reducing the rebroadcasts. REERA uses learning automaton to choose a Connected Dominating Set (CDS) of nodes in the network.

Localization-Free Interference and Energy Holes Minimization (LF-IEHM) protocol [24] is proposed for UWSNs. In this protocol, forwarder nodes are selected on the basis of the level of the water pressure. Nodes close to the water surface have low water pressure and are good candidates for forwarding the packets. If two or more nodes have the same level of the water pressure, the response time is taken into account to choose the best forwarder. The response time is a measure of the distance of a forwarder node from the source node.

Region Based Cooperative Routing Protocol (RBCRP) [25] is proposed for amplify and forward over Rayleigh faded channels in UWSNs. In this protocol, source nodes send the sensed data to the sink and available forwarder nodes. At the sink node, bit error rate is checked on the basis of which, either ACK or NACK is sent to the sources and forwarder nodes. This protocol also uses mobile sinks and energy harvesting techniques to further prolong the network lifetime and maximize the throughput. Energy-Efficient and Reliable Depth-Based Routing (ER-DBR) [26] protocol is proposed for UWSNs. This protocol uses packet reception probability (PRP), signal-to-noise ratio (SNR) and link quality indication (LQI) to select the most reliable forwarding nodes.

DBR [8] is a greedy algorithm, which attempts to deliver a packet from the source to the sink. This algorithm does not require geographical locations of nodes and only exploits the level of depth to route the packets. Moreover, this algorithm is very effective in UWSNs in a multiple-sink architecture [27]. Figure 3 presents an example of such architectures. In this architecture, several sinks equipped with acoustic and radio modems (capable of communicating through acoustic and radio signals) are deployed at the water surface.

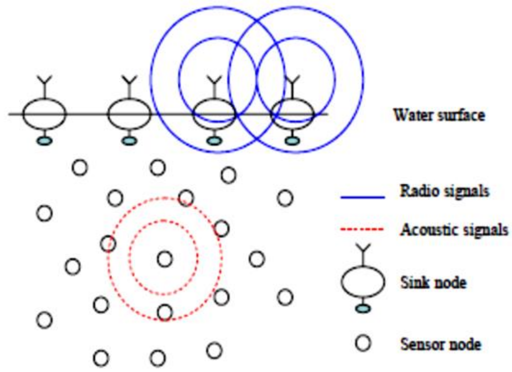

Figure 3. The multiple-sink UWSNs architecture [8].

Underwater sensors, which are only equipped with acoustic modems are scattered in the desirable 3D region (underwater). These underwater sensors can collect the data and help them to reach the sinks. Since all sinks have radio modems, they can effectively communicate information with one another through radio channels. Therefore, if a packet is delivered to a sink (S_i), it is assumed that S_i can deliver the received packet to other sinks or remote data centers. Therefore, if a packet is delivered to at least one sink, it is presumed that the packet is delivered to the destination. In DBR, it is assuming that each underwater sensor is aware of its level of depth. In practice, bathometer sensors can easily obtain depth information.

Figure 4 presents the format of the packets in this algorithm. The *SenderID* field contains the ID of the source node, the *Packet Sequence Number* contains the packet number, the *Depth* contains the depth of the packet sender and the last field contains the *data*.

SenderID	Packet Sequence Number	Depth	Data

Figure 4. DBR packet format

The procedure of DBR is briefly as follows. As soon as node *u* receives a packet from node *v*, it first extracts the depth of the node in the previous hop (i.e., *v*) from the received packet. Node *u* compares its own depth, d_c, with that of the node in the previous hop (i.e., *v*), d_p. If node *u* see itself closer to the water surface (i.e., $d_c<d_p$), it considers itself a candidate to forward the packet. Otherwise, node *u* drops the packet. It is clear that several neighbors of the sender node may be candidates to forward the packet. If all neighbors attempt to broadcast it, collision is increased and thus, more energy is consumed.

Therefore, in order to mitigate collision and energy consumption, the number of sender nodes should be controlled. Moreover, according to the multi-path feature of DBR, due to the broadcasting fashion of packet forwarding through the omnidirectional acoustic channel, a node may receive a packet several times. Therefore, this node may forward that packet several times, this, also, increases energy consumption. In order to overcome these two issues, that is, high collision and multiple forwards the same packet by a node, which increases energy consumption and reduces data delivery rate, the following approach is proposed in DBR:

Each node has a priority queue Q1 and a buffer Q2 in its memory. Each item in buffer Q2 has a unique ID, which is formed by the combination of *SenderID* and *Packet Sequence Number*. When this node successfully forwards a packet, the unique ID of this packet is inserted in Q2, each item in Q1 has two components: a packet and the scheduled sending time of that packet. The priority of an item in Q1 is determined by the scheduled sending time. An item with a closer sending time has a higher priority. As soon as a node receives a packet, it does not immediately forward it; it first stores an interval as the holding time of packet and then attempts to send it if necessary. The scheduled sending time of a packet is calculated based on its reception time and holding time.

When node *u* receives a packet, if it has not already forwarded it (it is not in Q2) and the packet is sent from a node from a deep depth (i.e., $d_p>d_c$), it inserts it in Q1. If the packet is already in Q1, it is received again (during holding time), and the new copy of the packet is received from a node with a lower than or equal depth (i.e., $d_p \le d_c$), this packet is removed from Q1. However, if the packet is received from a node with a deeper depth than *u* (i.e., $d_p>d_c$), the sending time of the packet is updated in Q1. After forwarding the packet, each node clears their Q1 from it and adds it to their buffer Q2. Equation (1) presents the calculation of the holding time.

$$f(d) = \frac{2\tau}{\delta} \cdot (R-d), \ \delta \epsilon (0, R], \ \tau = R/v_0 \qquad (1)$$

Where, *R* is the maximum transmission range of a node, *d* is the difference between the depths of the current and previous node, v_0 is the water propagation speed, and δ is a constant value in the network. Moreover, DBR considers a depth threshold, d_{th}, to control the number of nodes participating in sending the packet. A node only forwards a packet, when the difference between its depth and that of the node in the previous hop is larger than threshold d_{th} (i.e., $(d_p - d_c) > d_{th}$).

One of the drawbacks of DBR is its high end-to-end delay. Since in DBR, after receiving a packet, each node does not immediately forward it, but wait for an interval (holding time) and then forwards it if necessary, it increases the end-to-end delay of the packets. Moreover, packet delivery rate is low due to the intense multi-path control.

METHOD

In the proposed algorithm, a simple probabilistic mechanism is used to accelerate packet sending and improve multi-path of DBR. In ADBR, similar to DBR, after receiving packet *p* from node *v*, node *u* simply drops the packet if *u* is in a deeper depth than *v* (i.e., $d_p>d_c$), or has *p* in its buffer Q2. But, here in opposite of holding time in DBR, ADBR calls the accelerated routine. What will happen in the accelerate routine could be describe briefly as, in the accelerate routine, node *u* immediately sends the received packet with probability P_f it means that the corresponding holding time becomes 0. It is clear that in addition *u*, there may be several other neighbors receiving packet *p* from node *v*. Therefore, these nodes may also call the accelerated routine. Consequently, it is better to select probability P_f for each node in proportion to its depth difference with the node in the previous hop. Probability P_f is obtained by equation (2):

$$P_f = \frac{\left\lceil \frac{d}{\lambda} \right\rceil}{\frac{R}{\lambda}}, \quad \lambda = [1, R] \quad , d = d_p - d_c \qquad (2)$$

Where, *d* is the depth difference, *R* is the maximum transmission range, and λ is a constant value in the networks. In fact, ADBR divides the area above (shallower area) a sender node to R/λ subareas; these subareas are equal and have height λ. Selection function $\lceil \ \rceil$ in this equation makes probability P_f equal for all nodes in a certain area. Therefore, all nodes in a specific area have the same opportunity to instantly forward the packet. Consequently, packets are transferred from different routes to the sinks, which increases packet delivery rate. Fig 5 presents the accelerated routine.

According to Figure 5, assume that node *s* propagates a packet. Nodes *u*, *v*, *a*, *x*, *z*, *w*, and all neighbor nodes receive the packet. Now, all neighbor nodes shallower than *s*, which cannot find this packet in their Q2 buffer (i.e., *u*, *v*, *a*, *x*, *z*, and *w*), run the accelerated routine. Assuming that $\lambda = R/4$, four subareas are formed above *s*, where nodes *z* and *w* are in the first area, *x* is in the second area, *a* and *v* are in the third area, and *u* is in the fourth area. Since node *u* is in the fourth area, it forwards the packet with probability $P_f = 1$. However, the packet is forwarded by nodes *a* and *v* with probability $P_f = 0.75$, node *x* with probability $P_f = 0.50$, and nodes *z* and *w* with probability $P_f = 0.25$.

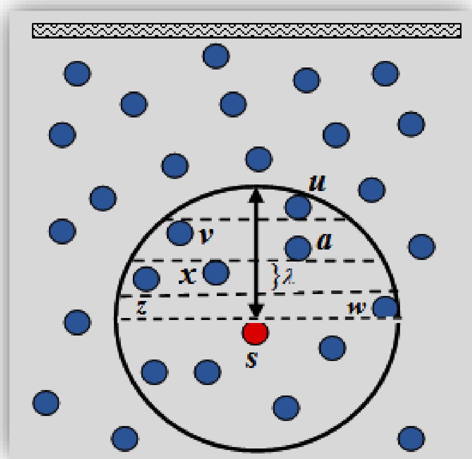

Figure 5. A high-level view of ADBR algorithm

It is clear that nodes in a shallower depth are preferred to forward the packet sooner to the sink. For instance, the node in the last subarea of the transmission range of the previous node (in Figure 5, *u* is in the last subarea of the transmission range of *s*) are preferred to instantly forward the packet to the sink and should

 https://doi.org/10.25077/jitce.6.01.19-28.2022

not delay their transmission (i.e., holding time of zero). Thus, accelerating technique reduces end-to-end delay. Moreover, it is likely that in the accelerated routine, some other nodes (in addition to *u*), for example, *z* instantly forward the packet. Therefore, the packet is forwarded to the sink through different paths, which increases packet delivery rate. The reason is that the packet forwarded by *u* may not reach the destination. In the accelerated routine, after forwarding the packet, each node removes it from its Q1 queue (if it exists) and adds it to its Q2 buffer.

However, if a node is not successful to forward the packet in the accelerated routine, that is, it does not get the chance to forward the packet, it proceeds with the DBR algorithms. It means that it inserts or updates the received packet into its Q1 queue to forward it after the holding time.

Now, in Figure 5, assume that nodes *u*, *a*, and *v* do not exist. In this case, if DBR is executed, nodes *x*, *z*, and *w* will wait for a long period (holding time) hoping that shallower nodes will forward the packet received from *s* to the water surface; whereas, there are no such nodes. This increases end-to-end delay. However, the scenario for the same case will be different with the proposed algorithm, the packet is instantly forwarded by node *x* with probability $P_f = 0.5$ and by nodes *z* and *w* with probability $P_f = 0.25$ (holding time 0). Thus, ADBR reduces end-to-end delay.

Accordingly, the proposed algorithm increases the multipath level and eliminates the holding time in some cases to increase the packet delivery rate and reduce the end-to-end delay. Moreover, by decreasing the number of collisions, energy consumption is also mitigated.

RESULTS AND DISCUSSION

This section evaluates the performance of ADBR algorithm using simulations. The results will be compared with those of DBR.

Simulation Model

JSIM simulator has used to perform all simulations [28, 29]. Simulation parameters are shown in Table 1. Sensor nodes are randomly deployed in a 500m x 500m x 500m 3D region. The network has N_{Sink} sinks, which are deployed at the water surface and it is assumed that these sinks are stationary at the water surface. Moreover, the network has N_{Source} source nodes. Although source nodes can be located anywhere, for the simplicity of simulations, they are randomly placed at the deepest layer of the environment (at the bottom of the water). Other sensor nodes are also deployed randomly in the environment. Here, the mobility model considered in [26] is used to model the mobility of sensors in the network environment.

It is assumed that source nodes generate a packet every 5 seconds (with a size of 50 bytes). The maximum transmission range of each node is 50 meters. The links' bandwidth is 1mb. The amounts of energy consumed for sending, receiving, and being idle are respectively 0.016J, 0.008J, and 0.0002J. The initial energy of sensor nodes is considered 5J; whereas it is assumed that sink nodes have unlimited energy. Moreover, the CSMA protocol is used in the MAC layer. The duration of each simulation is 1000 seconds and the final results are obtained by the mean of 20 different executions.

Table 1. Simulation parameters

Parameter	Values
Network size	500 ×500 ×500 m^2
Total nodes	N=200~500
Number of sinks	N_{Sink}=1, 5, 10
Number of source nodes	N_{Source}=1, 5, 10
Packet size	50 byte
Packet generation interval	5 seconds
Transmission range	50 m
links' bandwidth	1 mb
Initial energy for each node	5 J
Energy consumed for sending a packet	0.016 J
Energy consumed when receiving a packet	0.008 J
Energy consumed when being idle	0.0002 J
Simulation time	1000 seconds

Evaluation Metrics

Common measures to evaluate routing protocols include: packet delivery rate, mean end-to-end delay, and mean energy consumption. This study uses these measures to evaluate DBR and ADBR. In the following, they are explained in more details:

- **Packet delivery rate**: the ratio of the number of individual packets received by the sinks to the number of total generated packets by all source nodes.
- **Average end-to-end delay**: the average duration between sending instant packet by a source node and receiving it by a sink.
- **Average energy consumption**: the mean amount of energy consumed by each sensor node (except for sinks and sources) during the network lifetime.

Experimental Results

Experiment 1: this experiment aims to investigate the effect of parameter δ on the performance of the proposed algorithm and compare its results with the basic DBR algorithm. In this experiment, $\lambda=R/10$, $d_{th}=0$, and there are N_{Sink}=10 sinks and N_{Source}=1 sources in the network. Moreover, the number of nodes in the network are changed from *N*=200 to *N*=500. Figure 6 presents the results in terms of packet delivery rate, Figure 7 presents results of the average end-to-end delay, and Figure 8 shows the results in terms of average energy consumption.

As Figure 6, results of this experiment indicate that packet delivery rate of the proposed algorithm is better than that of DBR for *N*<500 and the packet deliver rate of both algorithms becomes about 100% for $N \geq 500$. However, for *N*=300, if $\delta = R$, packet delivery rates of the proposed algorithm and DBR are respectively 88% and 76%. This value reaches 88% and 86% for $\delta = R/2$. Moreover, for *N*=400 and $\delta = R, R/2$, this measure is 98% and 96% for the proposed algorithm and DBR, respectively.

The reason for these results is that the accelerated routine of the proposed algorithm increases the probability of forwarding packets through multiple separate routes to the sinks. Therefore, packet delivery rate of the proposed algorithm is more than DBR.

Furthermore, experimental results of Figure 7 indicate that for different values of N and δ , the average end-to-end delay of the proposed algorithm is less than that of DBR. The reason is that in the proposed algorithm, intermediate nodes instantly forwards received packets to the sink with the probability of P_f whereas, in DBR, after receiving a packet, intermediate nodes wait for a period (holding time) and then forward the packets to the sinks. Therefore, the average end-to-end delay of DBR is more than that of the proposed algorithm.

Finally, Figure 8 shows that the average energy consumption of the proposed algorithm is less than that of the DBR algorithm. The reason for these results is clear, since in the proposed algorithm, when node u attempts to forward its received packet in the accelerated routine, other neighbor nodes or nodes in subareas deeper than u, remove this packet from their Q1 queue and do not forward it anymore (of course, if they have not already forwarded it during the accelerated routine). However, in the basic DBR algorithm, more than one node may have the same holding time for the received packet and thus, after that period, they simultaneously forward the packet. This increases collisions and thus, energy consumption.

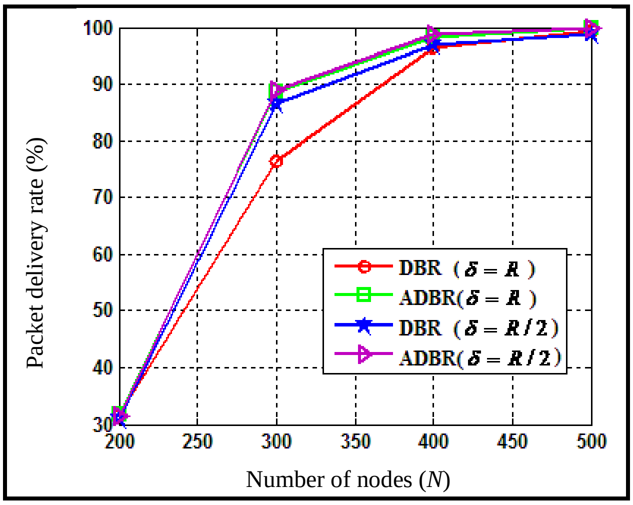

Figure 6. The effect of parameter δ on packet delivery rate in DBR and ADBR

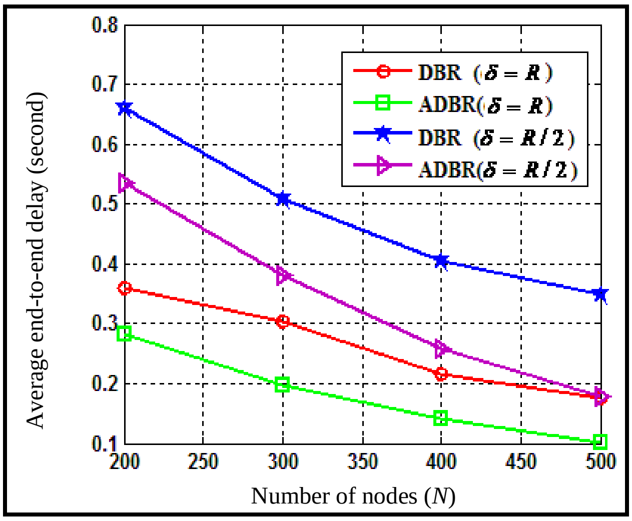

Figure 7. The effect of parameter δ on the average end-to-end delay in DBR and ADBR

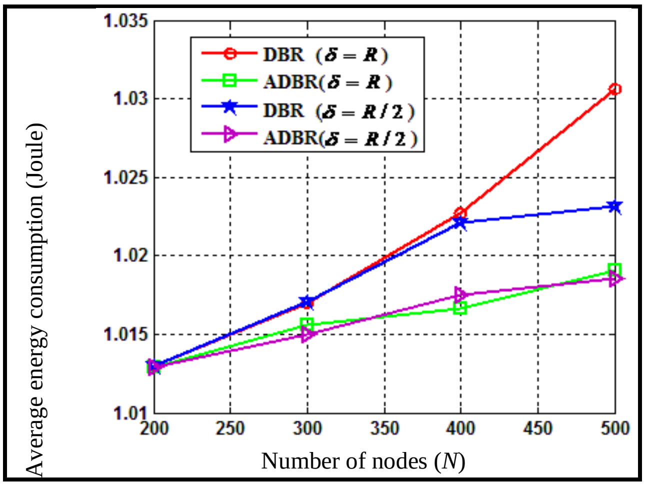

Figure 8. The effect of parameter δ on the average energy consumption in DBR and ADBR

Experiment 2: this experiment evaluates the performance of the proposed algorithm for different values of λ and compares its result with those of DBR. In this experiment, $\delta = R$, $d_{th} = 0$, N = 200 ~ 500 , and there are N_{Sink}=10 sinks and N_{Source}=1 sources. Figure 9 presents the results in terms of packet delivery rate, Figure 10 presents results of the average end-to-end delay, and Figure 11 shows the results in terms of the average energy consumption.

https://doi.org/10.25077/jitce.6.01.19-28.2022

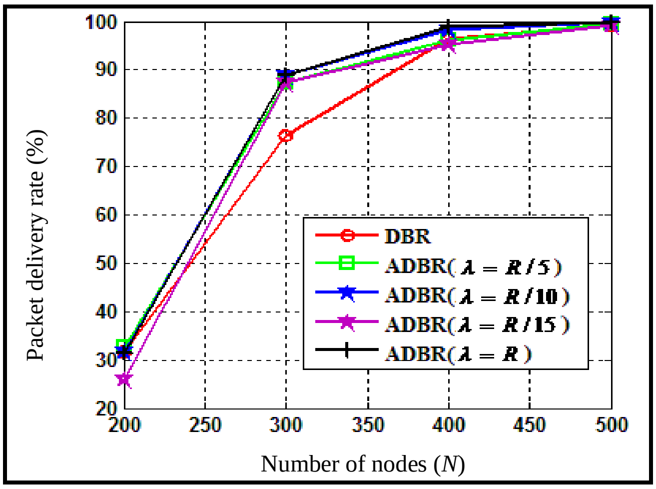

Figure 9. The effect of parameter λ on packet delivery rate in DBR and ADBR

As it is shown in Figure 9, packet delivery rate is about the same for different values of λ and accordingly, larger than that of the DBR algorithm. Moreover, experimental results of Figure 10 and Figure 11 indicate that increasing the value of λ reduces the average end-to-end delay (except for $\lambda = R$), and increases the energy consumption of the proposed algorithm. The reason is that increasing λ reduces the number of subareas above the packet sender node. Therefore, more nodes have the opportunity to forward the packet in the accelerated routine. Consequently, packet delivery delay is reduced and energy consumption is increased, however, energy consumption in the proposed algorithm is less than of the DBR. Whereas, if $\lambda = R$, since all neighbour nodes receiving the packet (of course in shallower depths than the sender node) are forwarding the packet in the accelerated routine, the probability is increased for collisions and the channel being busy, which increases the packet delivery delay.

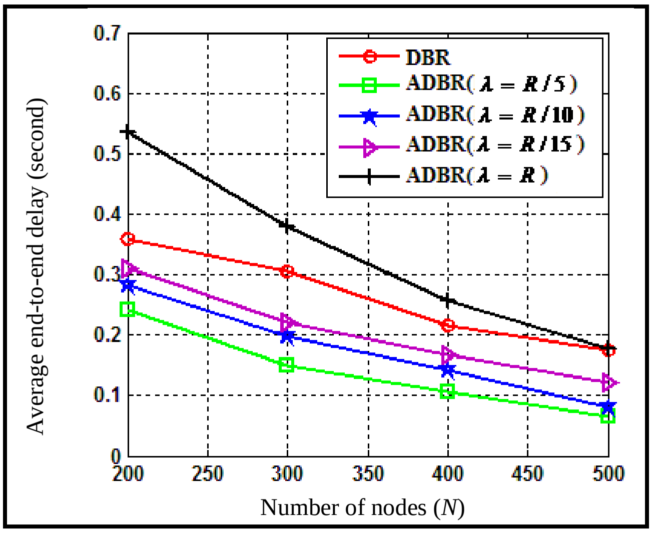

Figure 10. The effect of parameter λ on the average end-to-end delay in DBR and ADBR

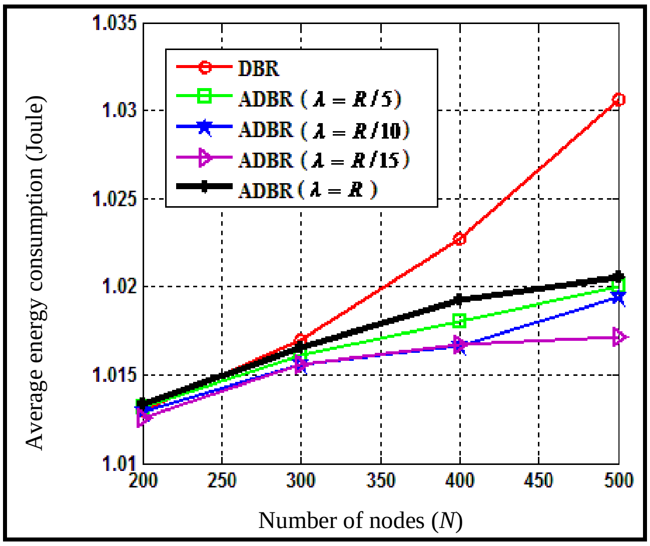

Figure 11. The effect of parameter λ on the average energy consumption in DBR and ADBR

Experiment 3: this experiment evaluates the number of source nodes, N_{Source}, on the performance of the proposed algorithms and DBR. In this experiment, $\delta = R$, $d_{th} = 0$, $\lambda = R/10$, $N = 200 \sim 500$, and N_{Sink}=10. Table 2, Figure 12, and Figure 13 present the results in terms of packet delivery rate, the average end-to-end delay, and the average energy consumption, respectively.

The results of this experiment show that the proposed algorithm outperforms DBR in terms of packet delivery rate, the average end-to-end delay, and the average energy consumption for different numbers of source nodes. It is clear that increasing the number of source nodes in the network increases the traffic, which increases collisions, makes transmission channels busy, and increases energy consumption. Therefore, increasing the number of source nodes in the network reduces packet delivery rate, while increasing the end-to-end delay and energy consumption. This is proved by the experimental results.

Table 2. The effect of the number of source nodes, N_{Source}, on the packet delivery rate of the proposed algorithm and DBR.

N_{Source}	Alg.	N=200	N=300	N=400	N=500
1	DBR	16.31	66.06	92.91	98.25
	ADBR	17.68	70.75	94.1	98.37
3	DBR	13	63.36	91.6	91.26
	ADBR	15.35	64.31	91.87	93.94
5	DBR	15.14	59.34	81.92	89.2
	ADBR	15.73	60.03	91.72	95.1

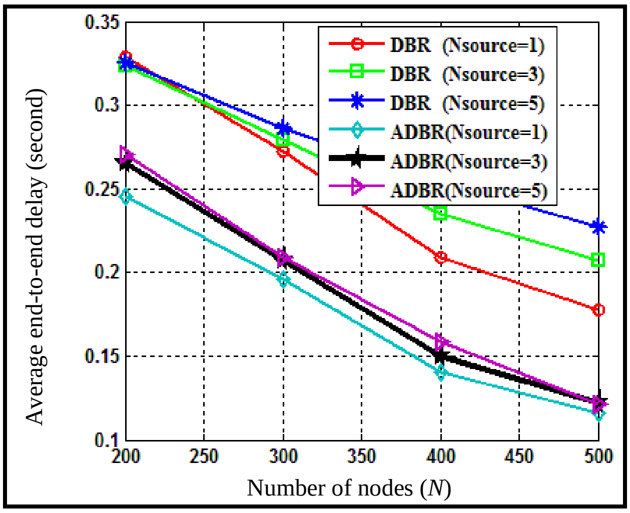

Figure 12. The effect of the number of source nodes, N_{Source}, on the average end-to-end delay of the proposed algorithm and DBR

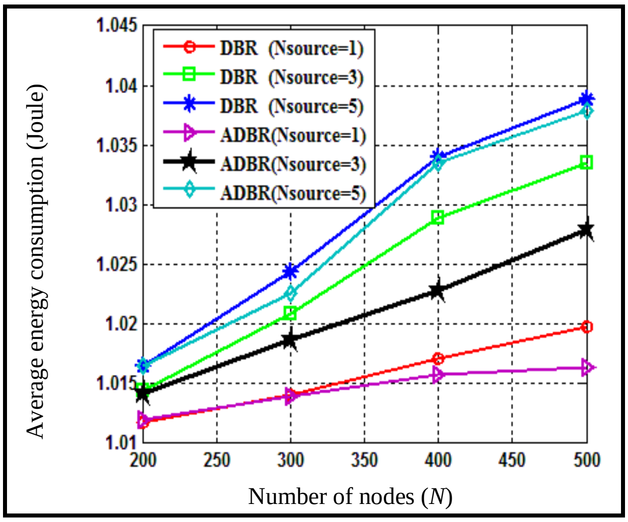

Figure 13. The effect of the number of source nodes, N_{Source}, on the average energy consumption of the proposed algorithm and DBR

Experiment 4: this experiment evaluates the number of sink nodes, N_{Sink}, on the performance of the proposed algorithms and DBR. In this experiment, $\delta = R$, $d_{th} = 0$, $\lambda = R/10$, $N = 200 \sim 500$, and N_{Source}=1. Table 3, Figure 14, and Figure 15 present the results in terms of packet delivery rate, the average end-to-end delay, and the average energy consumption, respectively.

The results of this experiment show that increasing the number of sink nodes also increases the packet delivery rate of the proposed algorithm and DBR. The reason is that both algorithms follow a greedy approach and try to forward the packets to the water surface. Therefore, a larger number of sink nodes at the water surface increases the chance of a packet to be received by a sink. Of course, the proposed algorithm outperforms DBR in terms of packet delivery rate for different values of N_{Sink}. The reason for this result is explained in the first experiment.

Furthermore, Figure 14 presents that increasing the number of sink nodes at the water surface slightly improves the end-to-end delay of both DBR and ADBR. The reason is that increasing the number of sinks increases the probability that a sink receives a packet earlier. The proposed algorithm outperforms DBR for this measure as well. Figure 15 also shows that changing the number of sinks has no considerable effect on the average energy consumption, since the number of sinks does not affect the routing and packet forwarding process.

Table 3. The effect of the number of sink nodes, N_{Sink}, on the packet delivery rate of the proposed algorithm and DBR

N_{Sink}	Alg.	N=200	N=300	N=400	N=500
10	DBR	31.62	76.31	96.5	99.33
	ADBR	31.62	88.75	98.37	99.83
5	DBR	28.87	75.31	94.8	98
	ADBR	30.14	80.75	94.7	98.06
1	DBR	23.43	75.56	93.3	97.13
	ADBR	23.5	77.43	94.16	97.33

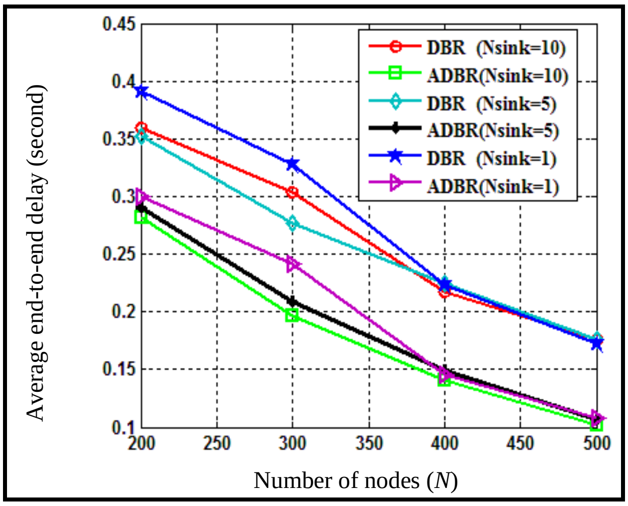

Figure 14. The effect of the number of sink nodes, N_{Sink}, on the average end-to-end delay of the proposed algorithm and DBR

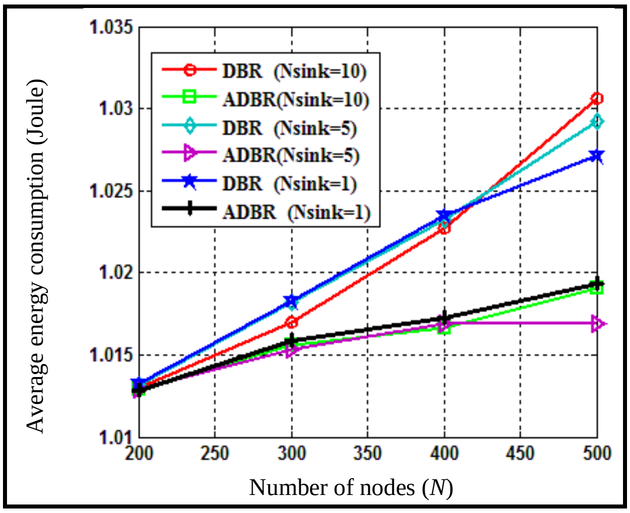

Figure 15. The effect of the number of sink nodes, N_{Sink}, on the average energy consumption of the proposed algorithm and DBR

https://doi.org/10.25077/jitce.6.01.19-28.2022

CONCLUSIONS

This study adds a probabilistic routine to the DBR algorithm to improve its performance. The improved algorithm, called ADBR, has an accelerated routine, which tries to send packets with no delay (temporary hold in the buffer) through different paths to the destination. The accelerated routine employs a simple probabilistic mechanism to make decisions about forwarding a packet after it is received from the node in the previous hop. Both the proposed algorithm and DBR were implemented by JSIM simulator and a set of experiments were conducted to evaluate and compare their performance in terms of packet delivery rate, average end-to-end delay, average energy consumption. Comparisons of results indicate that the proposed algorithm is more efficient than the basic DBR algorithm. For future works, the value of λ is dynamically selected according to the node density at each area of the network to increase the efficiency of the proposed algorithm.

REFERENCES

[1] J. Yick, B. Mukherjee and D. Ghosal. 2008. Wireless sensor network survey. Computer Networks. 52(12): 2292–2330.

[2] M. Jamshidi, M. Esnaashari, A. M. Darwesh and M. R. Meybodi, 2020. Using Time-Location Tags and Watchdog Nodes to Defend Against Node Replication Attack in Mobile Wireless Sensor Networks. International Journal of Wireless Information Networks, 27(1): 102-115.

[3] M. Jamshidi, M. Ranjbari, M. Esnaashari, N.N. Qader and M. R. Meybodi. 2018. Sybil Node Detection in Mobile Wireless Sensor Networks Using Observer Nodes. JOIV: International Journal on Informatics Visualization. 2(3): 159-165.

[4] A. Dalli and B. Seddik. 2016. ACQUISITION DEVICES IN INTERNET OF THINGS: RFID AND SENSORS. Journal of Theoretical and Applied Information Technology. 90(1): 194-200.

[5] J. H. Cui, J. Kong, M. Gerla and S. Zhou. 2006. Challenges: Building scalable mobile underwater wireless sensor networks for aquatic applications. IEEE Network. 20(3): 12-18.

[6] M. Jamshidi, E. Zangeneh, M. Esnaashari, A.M. Darwesh and M.R. Meybodi, 2018. A Novel Model of Sybil Attack in Cluster-Based Wireless Sensor Networks and Propose a Distributed Algorithm to Defend It. Wireless Personal Communications, pp.1-29 (in press).

[7] J. Partan, J. Kurose and B. N. Levine. 2007. A survey of practical issues in underwater networks. ACM SIGMOBILE Mobile Computing and Communications Review. 11(4): 23-33.

[8] H. Yan, Z. J. Shi and J. H. Cui. 2008. DBR: depth-based routing for underwater sensor networks. International conference on research in networking. Springer, Berlin, Heidelberg, 5 May: 72-86.

[9] Xie P., J.-H. Cui and L. Lao. 2006. VBF: Vector-Based Forwarding Protocol for Underwater Sensor Networks. Networking. 3976: 1216-1221.

[10] N. Nicolaou, A. See, P. Xie, J.-H. Cui and D. Maggiorini. 2007. Improving the robustness of location-based routing for underwater sensor networks. IEEE OCEANS Europe. Aberdeen, UK, 18-21 Jun: 1-6.

[11] C. Jinming, W. Xiaobing and C. Guihai. 2008. REBAR: a reliable and energy balanced routing algorithm for UWSNs. Seventh IEEE international conference on grid and cooperative computing. Shenzhen, China, 24-26 October: 349-355.

[12] J. M. Jornet, M. Stojanovic and M. Zorzi. 2008. Focused beam routing protocol for underwater acoustic networks. Third ACM International Workshop on UnderWater Networks WUWNet, San Francisco, California, USA, 15 September: 75-82.

[13] N. Chirdchoo, S. Wee-Seng and C. KeeChaing. 2009. Sector-based routing with destination location prediction for underwater mobile networks. International conference on advanced information networking and applications workshops. Bradford, United Kingdom, 26-29 May: 1148-1153.

[14] H. Daeyoup and K. Dongkyun. 2008. DFR: Directional flooding-based routing protocol for underwater sensor networks. IEEE OCEANS. 15 September: 1-7.

[15] K. R. Anupama, A. Sasidharan and S. Vadlamani. 2008. A location-based clustering algorithm for data gathering in 3D underwater wireless sensor networks. International Symposium on Telecommunications. Tehran, Iran, 27-28 August: 343-348.

[16] N. Javaid, N. Ilyas, A. Ahmad, N. Alrajeh, U. Qasim, Z. A. Khan, T. Liaqat and M. I. Khan. 2015. An Efficient Data-Gathering Routing Protocol for Underwater Wireless Sensor Networks. Sensors. 15(11): 29149-29181.

[17] J. Jiang, G. Han, H. Guo, L. Shu and J. J. Rodrigues. 2016. Geographic multipath routing based on geospatial division in duty-cycled underwater wireless sensor networks. Journal of Network and Computer Applications. 59: 4-13.

[18] M.R. Jafri, M. M. Sandhu, K. Latif, Z. A. Khan, A. U. H. Yasar and N. Javaid. 2014. Towards delay-sensitive routing in underwater wireless sensor networks. Procedia Computer Science. 37: 228-235.

[19] N. Javaid, M. R. Jafri, Z. A. Khan, N. Alrajeh, M. Imran and A. Vasilakos. 2015. Chain-based communication in cylindrical underwater wireless sensor networks. Sensors. 15(2): 3625-3649.

[20] M. Xu, G. Liu and H. Wu. 2014. An energy-efficient routing algorithm for underwater wireless sensor networks inspired by ultrasonic frogs. International Journal of Distributed Sensor Networks. 10(2): 351520.

[21] M. Xu and G. Liu. 2013. A multipopulation firefly algorithm for correlated data routing in underwater wireless sensor networks. International Journal of Distributed Sensor Networks. 9(3): 865154.

[22] M. Jamshidi, A. Andalib and L. Naseri. 2016. A Three-level Propagation Method of Routing Packets Specialized for Underwater Wireless Sensor Networks. International Journal of Computer Applications. 147(7): 29-33.

[23] R. Asgarnezhad and N. Nematbakhsh. 2015. A Reliable and Energy Efficient Routing Algorithm in Wsn Using Learning Automata. Journal of Theoretical and Applied Information Technology. 82(3): 401-409.

[24] A. Khan, I. Ahmedy, M. H. Anisi, N. Javaid, I. Ali, N. Khan, M. Alsaqer and H. Mahmood. 2018. A Localization-Free Interference and Energy Holes Minimization Routing for Underwater Wireless Sensor Networks. Sensors. 8(18): 165-183.

[25] N. Javaid, S. Hussain, A. Ahmad, M. Imran, A. Khan, and M. Guizani. 2017. Region based cooperative routing in underwater wireless sensor networks. Journal of Network and Computer Applications. 95: 31-41.

[26] S. N. Pari, M. Sathish and K. Arumugam. 2018. An Energy-Efficient and Reliable Depth-Based Routing Protocol for Underwater Wireless Sensor Network (ER-DBR). Advances in Power Systems and Energy Management. 436: 451-463.

[27] W.K. Seach, and H. X. Tan. 2010. Multipath virtual sink architecture for underwater sensor networks. In Underwater Acoustic Sensor Networks. 19 May: 78-113.

[28] J-SIM Simulator, https://sites.google.com/site/jsimofficial/, Accessed on August 25, 2018.

[29] M. Jamshidi, E. Zangeneh, M. Esnaashari and M. R. Meybodi. 2017. A lightweight algorithm for detecting mobile Sybil nodes in mobile wireless sensor networks. Computers & Electrical Engineering. 64: 220-232.

 https://doi.org/10.25077/jitce.6.01.19-28.2022