An opportunistic void avoidance routing protocol for underwater sensor networks

Ghoreyshi, Seyed Mohammad; Shahrabi, Ali; Boutaleb, Tuleen

Published in:
IEEE 30th International Conference on Advanced Information Networking and Applications (AINA)

DOI:
10.1109/AINA.2016.96

Publication date:
2016

Document Version
Author accepted manuscript

An Opportunistic Void Avoidance Routing Protocol for Underwater Sensor Networks

Seyed Mohammad Ghoreyshi, Alireza Shahrabi, Tuleen Boutaleb
School of Engineering and Built Environment
Glasgow Caledonian University
Glasgow, UK
{Seyed.MohammadGhoreyshi, A.Shahrabi, T.Boutaleb}@gcu.ac.uk

***Abstract*—An Underwater Sensor Network (UWSN) has many unique features that makes it different from terrestrial network. This includes lower bandwidth, longer propagation delay, dynamic topology, high error rate, and energy constraint. To overcome the limitations of such an environment, opportunistic routing has recently attracted much attention due to its ability to improve the performance of UWSNs in terms of packet delivery ratio and energy saving. With the aid of opportunistic data routing, underwater sensors can collaboratively route a packet towards the destination which is a more adequate approach for sparse and lossy channels. In this paper, we propose a new routing protocol, called Opportunistic Void Avoidance Routing (OVAR), to address the void problem and high bit error rate without relying on any positioning system. OVAR is able to efficiently bypass all kinds of void areas with the lowest possible cost (including energy and delay) while prioritising the group of candidate nodes with the highest packet advancement. Given the density of neighbours (sparse or dense), each forwarding node is able to hold a trade-off between packet advancement and energy consumption by adjusting the number of nodes in its forwarding set. OVAR is also able to select the forwarding set in any direction from the sender without including any hidden node. The results of our extensive simulation study show that OVAR outperforms other protocols in terms of packet delivery ratio, energy consumption, and average end-to-end delay.**

I. Introduction

Underwater acoustic sensor network has recently attracted much attention due to its significant ability in ocean monitoring and resource discovery. Due to restrictions on the use of the radio waves, acoustic transmission is most commonly used in the underwater environment. Required data is collected by the underwater sensors and directed towards the sink on the surface. Afterwards, sink can transmit collected information to the monitoring centre via satellite for further analysis [1], [2]. From the perspective of routing protocols, the presence of void areas, high bit error rate, and energy-conservation are perhaps the most challenging issues.

There are various reasons for the presence of void areas in UWSNs. In most cases, lack of employing enough number of sensor nodes, due to their high cost, while covering a large monitoring area might lead to sparsely deployment of the sensors and consequently creation of some void area. Moreover, relocation of underwater sensor nodes by water current can potentially create a void area [3]–[5]. On the other hand, adverse characteristics of underwater channel can cause the high bit error resulting in high attenuation, channel fading, noise, and Doppler spread, etc. Finally, the limited bandwidth of acoustic transmission reduces communication efficiency between underwater nodes [6], [7]. In terms of energy consumption, there are also some restrictions due to difficulties of replacing or recharging of batteries, which are the main energy supply for the nodes, in the adverse and often deep underwater environment. In addition, underwater sensors consume more energy than terrestrial sensors because of using acoustic communication [8]. Thus, employing an efficient routing protocol is quite essential to prolong the whole network lifetime.

Opportunistic routing is a promising scheme in sensor networks because of its remarkable ability to increase transmission reliability and network throughput. In this way, packet forwarding is enhanced by taking advantage of simultaneous packet reception among one node's neighbours and their collaboration to forward the packet [9]. However, applying a terrestrial opportunistic routing protocol in UWSNs without considering its specific features is not possible in most cases. In underwater environment, forwarding set selection without hidden terminal, and prioritizing them are affected by features like high error bit rate, energy consumption, node movement and slow propagation speed. Furthermore, some terrestrial opportunistic protocols are GPS-based which make them inappropriate for GPS-denied underwater environment.

In this paper, we propose a new Opportunistic Void Avoidance Routing (OVAR) protocol in order to increase the throughput and reliability in the sparse and lossy underwater environment. The way which OVAR deals with sparse and lossy environment imposes less overhead in comparison to those protocols using high cost localisation to obtain their geographic coordinates in underwater environment. Furthermore, unlike the stateful protocols which require global topology information, OVAR only depends on the information provided by one-hop neighbouring nodes. Each forwarding node selects its forwarding set with the aid of information obtained from the distributed beaconing mechanism initiated from the sink node. OVAR is able to bypass void areas before being stuck in a void node, and simultaneously selects group of candidate nodes with the highest advancement towards the sink. Forwarding set is selected in such a way that forwarding nodes can hear each other and suppress duplicate transmissions which leads to decrease in energy consumption and congestion. In order to prevent energy wasting in a high-density forwarding set, the number of receiving nodes can be appropriately adjusted. Simulation results demonstrate that our protocol increases packet delivery chance in each transmission and inherently excludes the paths lead to a void area.

The rest of this paper is organised as follows: In Section II, we review the related work in this field. In Section III, the details of OVAR protocol is presented after introducing network architecture and void problem. Section IV evaluates the performance of OVAR through simulations. In Section V, we conclude the paper and discuss future work.

II. Related Work

In this section, we review some geographic routing protocols in the UWSN and how they take advantage of the opportunistic data forwarding to deal with the void and channel fading. It is worth mentioning that GPS does not work in the underwater environment; however, some studies still assume that underwater nodes can obtain their 3D geographic coordinates with the aid of the localisation service which is reported to be a challenging issue in the underwater environment [10].

Some routing protocols such as VBF, HHVBF, and AHH-VBF [4], [11], [12] are location-based greedy routing in which forwarding nodes are selected within a virtual pipeline faced toward the destination. These protocols consider a desirableness factor to select forwarding candidates among the nodes inside the pipe. In order to reduce the latency, the nodes are selected in a way that a packet is forwarded using the longest possible hop from the transmitter while maintaining its closeness to the routing vector. However, in underwater environment, the likelihood of bit error increases with increasing the traversed distance. They try to compensate this defect by increasing the radius of the pipe and involving more forwarding nodes which causes higher collisions and hence waste of energy. However, increasing the radius of pipe does not help to resolve the problem of void area which is mostly occurred in sparse networks. The transmitter cannot utilise the nodes outside the pipe to bypass a void area located in the pipe. Moreover, these protocols suffer from hidden terminal nodes because the neighbouring nodes of the sender can be out of range of each other (e.g. be placed in different directions of the pipe).

In another group of studies, depth information is employed to route the packets towards their destinations (one of the sonobuoys on the water surface). DBR [10] is the first depth-based routing protocol proposed for UWSNs. Nevertheless, forwarding set selection is not performed in an optimal way (having duplicated packets problem) in DBR. On the other hand, HydroCast [7] and VAPR [13] represent the pressure-based routings which are enhanced by opportunistic data forwarding and void handling. These protocols try to select a subset of forwarding candidates with maximum advancement towards the destination and also addressing the hidden terminal problem. However, HydroCast relies on the use of 2D surface flooding method to discover a recovery path for local maxima nodes in the near-surface layer. More importantly, void areas can appear in deeper regions of the water which is not considered in this protocol.

VAPR tries to bypass void areas by holding information of up to two-hop neighbouring nodes which impose high overhead to the system. Moreover, the beaconing procedure in VAPR (for multi-sink architecture) is not properly utilised in a way that beacons carry additional useful information in addition to the hop count. For this reason, in VAPR, each node is forced to periodically measure the distance to its neighbours and broadcast the measured information to its one-hop neighbours. As another problem, packet can only be forwarded up or down depending on the selected direction which cannot utilise subsets of nodes in the horizontal direction (including nodes with lower depth and higher depth together in the forwarding set).

III. Opportunistic Void-Avoidance Routing Protocol

In this section, we present our OVAR protocol in detail.

A. System Model

We assume that each node knows its current depth (i.e. vertical distance from each node to the water surface) by using an embedded depth sensor [10]. Moreover, nodes can obtain their hop count distance to the sink with the aid of distributed beaconing [13]. Nodes randomly move in the horizontal direction because of the water current and their small vertical movements are negligible. The batteries are energy supplier of the underwater sensor nodes. Nodes are homogeneous in terms of energy consumption and transmission range. The Thorp model is used for designing the underwater acoustic propagation and adjusting the transmission power [4], [12]. Moreover, we consider a lossy channel in which path loss and bit error depends on the traversed distance and signal frequency. The path loss or attenuation over distance d with the signal frequency f is defined as follows [6]:

$$A(d,f) = A_0 d^k \alpha(f)^d \tag{1}$$

where A_0 represents a unit-normalizing constant, and k is the geometric spreading factor which is set to 1.5 for practical scenarios. Furthermore, the absorption coefficient $\alpha(f)$ is defined by the Thorp formula. The ratio of the signal power which contains meaningful data to the unwanted signal power (i.e. noise) is defined as the signal-to-noise ratio (SNR). By considering the attenuation formula, the signal-to-noise ratio over distance d with the signal frequency f can be expressed as follows [6]:

$$SNR(d,f) = \frac{PR(f)}{A(d,f)PN(f)} \tag{2}$$

where $PR(f)$ and $PN(f)$ indicate transmission power of forwarding node with frequency f, and the underwater environment noise, respectively. In order to decode the received signal without error, SNR at the receiver should be higher than a detection threshold. The ambient noise in the underwater environment includes four main components of turbulence $PN_t(f)$, shipping $PN_s(f)$, waves $PN_w(f)$ and thermal energy $PN_{th}(f)$ which can be expressed as [4]:

$$PN(f) = PN_t(f) + PN_s(f) + PN_w(f) + PN_{th}(f) \tag{3}$$

These noises are dominant in the different frequency regions which can affect the communication channel throughput.

B. OVAR Overview

Having a single permanent destination, in the single-sink model, or a number of destinations, in the multi-sink model, is a unique useful feature in developing void-aware routing protocols for UWSNs, which has been perhaps neglected in most routing protocol developments in this field. Using this feature, the process of establishing a void-avoidance route for all the nodes in the network to their destination(s) can be initiated by the sink(s) and cascaded down by intermediate nodes, similar to the route establishment phase of some distance vector routing protocols in wireless ad hoc networks. In order to obtain reachability information and neighbouring nodes discovery, each node periodically broadcasts a beacon which includes the hop count information (proximity of nodes to the sink) and also some neighbouring information for updating the routing tables. The beaconing mechanism has already implemented and utilised by some MAC protocols [14] for neighbouring nodes discovery. This mechanism can be augmented to support the hop count information required by OVAR without imposing new overhead.

OVAR employs a hop-by-hop forwarding set selection manner to deliver packets to the sink. Each packet holder uses local information of hop distance and packet advancement to determine its own forwarding set. In addition, the forwarding set should prevent hidden terminal problem which is caused by including the nodes that are out of range of each other. In order to manage the energy, the number of collaborative nodes can be adjusted according to the density of the network. Afterwards, in order to prioritise the multiple forwarding nodes, each node considers its depth as the second metric to set a relaying timer. The node with the highest priority (lowest depth) transmits the packet earlier and other low priority nodes can drop the packet after hearing the transmission. This suppression mechanism along with the selecting a path with lower hop count leads to more energy saving and higher delivery ratio. By employing hop-by-hop forwarding set selection, OVAR is highly scalable to be used in large underwater sensor networks. Finally, OVAR automatically excludes all the routes leading to void areas and therefore does not need to switch any high overhead recovery mode for void bypassing.

C. Beaconing Model

We consider S as a single sink on the surface for collecting the information. Other nodes including relay nodes and anchored nodes can be shown by $V = \{R_1, R_2, ..., R_m\}$. Let $m = |V|$ denote the number of nodes in V. We define $N(R_i)$ as the set of R_i neighbouring nodes. Based on R_i members hop count values, $N(R_i)$ can be partitioned as follows:

$$N(R_i) = L(R_i) \bigcup E(R_i) \bigcup H(R_i) \tag{4}$$

where $L(R_i)$, $E(R_i)$, and $H(R_i)$ indicate disjoint neighbouring sets of R_i with lower, equal, and higher hop count value, respectively. Each node in V locally holds a table about its neighbouring nodes and classifies them based on the partitioning criteria expressed in Equation 4.

At the beginning of beaconing process, all the nodes in V are isolated from each other and their hop count value is set to a maximum value, to show no connectivity with S. Node S is the final destination on the surface and accordingly its hop count number is set to 0. In our beaconing model, node S along with all the nodes in V periodically propagate a beacon including their ID, depth, hop count value, and all neighbouring nodes in subset $E(R_i)$ (neighbouring nodes with the same hop count value as the sender). Nodes with the maximum hop count value are exempted from the beaconing until they find a path to the sink. The sink node initiates the beaconing process and gradually is cascaded down to the network. The beacon interval for each node is considered as T_{update}.

Depending on the hop count value of the beacon, receiving node decides how to deal with it. Upon receiving a beacon with lower hop count, receiving node updates its hop count value and holds the sender's ID in its subset L and attaches its depth, and all other existing IDs within the beacon to the sender entry in the table. If a node receives other beacons with the same lower value, it will also add them to the table in the same manner. Since, all nodes periodically broadcast a beacon, receiver knows all of its neighbours with lower hop count (all next available nodes to relay the packets during the packet forwarding stage). On the other hand, when a node receives a beacon with the same hop count value as its own value, it only holds sender's ID in its subset E and broadcasts it with the next beacon. Furthermore, receiving node drops all the beacons with the higher hop count value.

According to the information extracted from the table, each node can form its own adjacency graph. Only nodes in the subset L are considered to be included in the graph and other nodes which are inaccessible from the sender can be removed from the table. It can simply be realized by checking that beacons from which neighbours directly received by the node.

Upon changing the hop count value in each node (e.g. finding a shorter path), the table is updated based on the current available path and it sends out a beacon with new hop count value, and also resets the beacon timer. Moreover, nodes employ $T_{invalid}$ which shows that how long a path is valid at each node. If node cannot sense any neighbouring node with lower hop count in its vicinity in this time interval, it should determine a new hop count value based on the recently received, and still valid, beacons. The routing performance depends on the assigned value for T_{update} in a way that higher value leads to invalidity of the vicinity information and lower value imposes high communication overhead. According to the mobility pattern and speed of underwater nodes, T_{update} should be carefully determined.

D. Routing Algorithm

In OVAR routing algorithm, we select a forwarding set based on two metrics: packet delivery probability and packet advancement. In this section, we first explain how packet delivery probability can be estimated from receiving beacons.

We then specify how packet advancement is modelled in our routing algorithm.

OVAR routing algorithm is divided into 4 phases. First, an adjacency graph is constructed at every node and using a heuristic some clusters, i.e. clique sub-graphs, are created to ensure that hidden nodes are removed from forwarding sets. Second, the best forwarding set is selected using Expected Packet Advancement to maximise the chance of successful delivery of a packet. Third, the number of forwarding nodes in the forwarding set is adjusted to make a trade-off between reliability and energy consumption. Finally, the holding time is calculated at each candidate node before forwarding the packet. In order to illustrate the protocol, we consider a local OVAR scenario like the one in Fig. 1.

Relationship between Packet delivery probability and transmission distance: Assuming node R_i intends to send a packet to the sink S, and $L(R_i) = \{n_1, n_2, .., n_c\}$ shows the available candidates of node R_i (neighbouring nodes with lower hop count values) which are ordered increasingly based on their depth values. Let $c = |L(R_i)|$ denote the number of candidates in $L(R_i)$. Node R_i is aware of the packet delivery probability of its neighbours. For instance, if R_i has received a beacon from n_k, $1 \leq k \leq c$, can calculate pairwise distance $Dist(R_i, n_k)$ based on the receiving signal power from the beacon. In this way, node R_i can calculate all pairwise distances between itself and its neighbouring nodes, and add them to its neighbouring table. Thus, all nodes in $L(R_i)$ can be associated with a packet delivery probability P_{ik} $(1 \leq k \leq c)$ which can be calculated, as explained later in this section, based on the distance from node R_i to n_k. Node n_k is a neighbouring node of R_i when $P_{ik} > P_T$, where P_T represents a probability threshold. Otherwise, the sent packet cannot be decoded free of error. Moreover, we assume that the packet delivery probability on each acoustic link is independent.

According to the model used in [7], the bit error probability over distance d can be calculated as follows:

$$P_e(d) = \frac{1}{2}\left(1 - \sqrt{\frac{SNR_{avg}(d,f)}{1 + SNR_{avg}(d,f)}}\right) \tag{5}$$

where $SNR_{avg}(d,f)$ is the average signal-to-noise ratio over distance d. The bit error probability increases by increasing the distance due to channel fading. Moreover, a packet (from node i to node j) with size n bits can be delivered over distance d with the probability P_{ij} [7]:

$$P_{ij} = (1 - P_e(d))^n \tag{6}$$

Let F denotes R_i's forwarding set including all the nodes used in the opportunistic data forwarding. Let $r = |F|$ denote the number of nodes in F. Now, our first goal is to select the subset F from $L(R_i)$ in a way that it can maximize packet delivery probability and resolve the hidden terminal problem in the lossy underwater environment.

Obviously, a packet transmission obtains more chance of delivery if more forwarding nodes are involved in the packet forwarding. With $r = 1$, only one node from $L(R_i)$ is selected for packet forwarding and therefore the successful delivery chance is limited to the packet delivery probability of a single node. For instance, in Fig. 1, if we just select node n_1, the delivery probability is equal to P_{i1}. A traditional routing protocol without opportunistic routing might ideally achieve $max(P_{i1}, P_{i2}, ..., P_{ic})$ packet delivery in each step towards destination which is not suitable for the lossy underwater acoustic channel. On the other hand, by maximizing the forwarding set size, i.e. $r = c$, all neighbouring nodes with lower hop count take part in the packet delivery. Although, this certainly increases the chance of packet delivery but also increases the energy consumption and also the network congestion. Moreover, involving nodes without considering the hidden terminal problem may result in redundant paths and packet collisions. Three-dimensionality of underwater environment makes the hidden terminal problem even worse due to the existence of some neighbouring nodes in different directions.

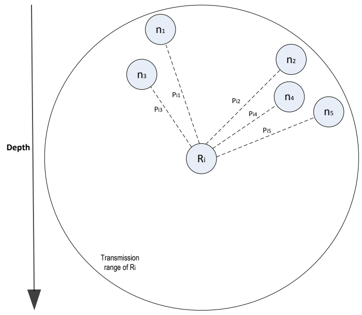


Fig. 1: Example in which node R_i is forwarding a packet

Packet Advancement: To specify the priority of relaying nodes, we define a fitness factor, α, which represents the depth difference between sender's depth, D_s, and receiver's depth, D_r, in a normalised value as follows:

$$\alpha = \frac{D_s - D_r}{R} \quad (-1 \leq \alpha \leq 1) \tag{7}$$

where R is the transmission range of sensor nodes. According to the fitness factor, a relay node with lower depth has higher priority to relay the packets as it is closer to the surface where the sink is located. Negative value of the fitness factor indicates that the receiver node is located below the sender, perhaps due to the presence of a void area in the routing path. In contrast to the majority of greedy routing protocols, OVAR gives these kinds of nodes (with higher depth than sender and maybe with higher geographical distance to the sink) the chance of participating in the packet forwarding to bypass the void areas. However, these nodes still can be prioritised based on the lower depth due to the fact that the packet most likely should be relayed upward over the next step to become closer to the final destination on the surface. In order to use this value in our calculations, we further normalise this value to be placed in the range [0,1] as follow:

$$\beta = \frac{1}{2}(\alpha + 1) \quad (0 \leq \beta \leq 1) \tag{8}$$

Now, we can explain four phases of our routing algorithm: forming the adjacency graph and then dividing them into some clusters, selecting the best forwarding set, adjusting the number of forwarding nodes in the forwarding set and finally time holding calculation. Details of all these phases are presented in Algorithm 1.

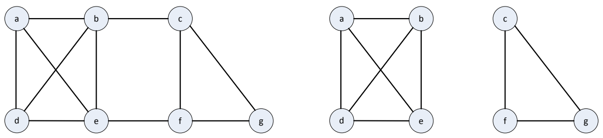
Algorithm 1 OVAR Routing Algorithm

```
procedure RECEIVEPACKET($R_i$, P)
    if Packet has not already been received and
            $R_i \in$ forwarding set of P then
        Calculate $\alpha$ and $T_{hold}$
        Set forwarding timer
    else
        Drop P
    end if
end procedure

procedure FORWARDPACKET($R_i$, P)
    if forwarding timer expired then
        $F(R_i) = \emptyset$
        $L(R_i) = \{n_k | 1 \le k \le c\}$
        $G(R_i) = AdjacencyGraph\big(L(R_i), Table(R_i)\big)$
        $\Psi(G(R_i)) = CliqueSubGraphs(G(R_i))$
        for all $\Phi_z \in \Psi(G(R_i))$ do
            Calculate $EPA(\Phi_z)$
        $F(R_i) = \Phi_z$ with maximum EPA value
        for $j = 1$ to $r$ do
            Calculate $EEPA(F, j)$
        $j_{max} = \arg\max_j EEPA(F, j)$
        for all $j > j_{max}$ do
            $F(R_i) = F(R_i) - n_j$
        $P.ForwardingList \leftarrow F(R_i)$
        Forward P
    else
        Drop P
    end if
end procedure
```

1) Adjacency graph construction and clustering: In the first step of OVAR execution, each forwarding node, R_i, constitutes its adjacency graph $G(R_i)$ with the aid of information provided by beaconing. In order to remove the possibility of having hidden nodes in a forwarding set, the forwarding node extracts all clique sub-graphs (clusters) from $G(R_i)$. After creating all possible hidden-node-free clusters, the forwarding node should select the best cluster with the best packet advancement and energy efficiency among others to forward the packet.

Transforming any graph to a set of clique sub-graphs (in which all vertices are directly connected to each other) is a NP-hard problem [15]. Thus, we apply a heuristic which is computationally more efficient for converting $G(R_i)$ to a set of sub-graphs with no hidden node in each sub-graph. Transforming an adjacency graph to a number of clique sub-graphs can be done by removing some edges from the adjacency graph to result in some clusters with the same node-degree for all the nodes in each cluster. To this, forwarding node independently builds an adjacency sub-graph for each of its neighbouring nodes and converts it to a clique sub-graph. Our heuristic method first checks the degree of all nodes in a sub-graph. If all nodes have equal degree, sub-graph is a clique sub-graphs and can be considered as a candidate forwarding set. Otherwise, it starts by removing the node with the lowest degree from the sub-graph. If two nodes have the smallest degree at the same time, the node with lower packet probability and lower depth is removed (delivery probability * β). This process continues until all nodes in the sub-graph have the same degree; a cluster with no hidden node. Similarly, forwarding node will perform this procedure for all neighbouring nodes to obtain all possible clusters in the neighbourhood. Figure 3 shows how an example of an adjacency graph can be transformed to some clique sub-graphs.



(a) Original graph (b) Clique graph

Fig. 2: Transforming original graph into a clique graph

2) Forwarding set selection: Afterwards, based on our criteria, OVAR benefits from a cluster which can simultaneously maximise packet advancement and packet delivery probability. Let $\Psi(G(R_i)) = \{\Phi_1, \Phi_2, ..., \Phi_t\}$, $1 \le t \le c$, as a set of all created clusters with no hidden node from $G(R_i)$ which are decreasingly ordered based on their number of members. Moreover, the members of each cluster Φ_z are decreasingly ordered based on the value of β presented in Equation 8. First, we define the Expected Packet Advancement (EPA) to estimate the advancement of each packet which is relayed by a set of nodes. We propose EPA for each cluster $\Phi_z = \{m_1, m_2, ..., m_l\}$, $1 \le l \le c$, created by forwarding node R_i as follows:

$$EPA(\Phi_z) = \sum_{k=1}^{l} \beta_{ik} P_{ik} \prod_{y=0}^{k-1} \overline{P}_{iy} \qquad (9)$$

where $\overline{P}_{iy} = 1 - P_{iy}$ and $P_{i0} = 0$. EPA is calculated for all clusters in $\Psi(G(R_i))$. Then, each cluster which can maximise this value will be selected as the forwarding set (subset $F(R_i)$). By utilising this approach, loss probability is decreased and duplicate transmission paths can be disappeared efficiently without imposing high cost to the system.

3) Reliability and energy consumption trade-off: Sometimes, specially, in a dense network, there are too many nodes in a cluster resulting in wasting energy. Hence, we introduce a new metric, Expected Energy and Packet Advancement (EEPA) to balance energy efficiency and routing efficiency. To consume energy more efficiently, it is assumed that nodes can only listen to a transmitted packet if the packet is destined for them. This is achieved by equipping the nodes with low power receiver to wake them up to participate in the packet forwarding only by checking the header of the packets. Thus, the receiving energy consumption can be reduced by decreasing the number of receivers in the forwarding set. For instance, if E_{rx} is considered as the receiving energy at each node, by removing n nodes from F, we can save $n * E_{rx}$ units energy.

Let $EPA(F, j)$ and $E(F, j)$ be the expected packet advancement and energy consumption of the forwarding set, respectively, when j nodes participate in the packet forwarding. The maximum value for EPA and energy (EPA_{max} and E_{max}, respectively) can be obtained by involving all the nodes in the forwarding set, i.e. $EPA(F, r)$ and $E(F, r)$, where $r = \mid F \mid$. In this way, by selecting j forwarding candidates from F, EEPA can be defined as follow:

$$EEPA(F, j) = \mu \frac{EPA(F, j)}{EPA_{max}} - \rho \frac{E(F, j)}{E_{max}} \tag{10}$$

where μ and ρ are defined as the weighting coefficients for EPA and energy, respectively. These coefficients can be set according to the desired criteria and density of the network. For instance, if the network is more interested in the energy saving rather than packet delivery, it can increase the ρ against the μ, or vice versa.

Forwarding set should be checked for different number of members to achieve the maximum possible value for EEPA. This can be done by examining EEPA for $j = 1, ..., r$ and finally to pick the set with the largest value and accordingly remove other extra nodes from the forwarding set, if required. In this way, we start from an empty set and add nodes (ordered by their advancement) to the forwarding set one by one. Eventually, the optimal set is selected to relay the packet. In a sparse network all nodes are held in the forwarding set to increase the reliability; however, in a dense network some nodes are removed to control the energy dissipation.

4) Holding time of forwarded packets: Eventually, node R_i locally selects the forwarding set $F(R_i)$ based on our criteria and broadcasts the packet. The packet header contains all IDs of members of $F(R_i)$. Receiver node should be in the forwarding set of the sender to accept the packet; otherwise it drops the packet. Upon receiving a packet by a forwarding candidate, it sets a forwarding timer proportional to its fitness factor (Equation 7). A node with the highest priority has the lowest forwarding timer value among forwarding candidates and if the packet is relayed by this node, other lower priorities candidates should discard the packet after hearing the packet transmission. A low priority candidate can become a forwarding node if all the nodes in the forwarding set with higher priority failed to receive or relay the packet which can be recognised by listening to the channel. This procedure with the aid of timer scheduling is repeated until the packet is successfully relayed to the next hop. By using this mechanism, redundant transmissions are prevented which leads to more energy saving for the whole network.

Each candidate node calculates its forwarding timer, T_{hold}, as follows:

$$T_{hold} = \frac{1}{2}(1 - \alpha)T_{Delay} + \frac{R - |\vec{SC}|}{\nu_{sound}} \tag{11}$$

where T_{Delay} is the predefined maximum delay which should be set in a way that all forwarding candidates are able to hear the transmission of higher priority nodes before relaying the packet. R and ν_{sound} are the transmission range of node, and the propagation speed of sound in the water, respectively. $|\vec{SC}|$ indicates the relative distance between the sending node S to the candidate node C which can be estimated based on the received signal strength. The first part of equation ensures that candidate nodes hold the packet based on their priorities (the greater fitness factor value, the shorter timer) and the second part of the equation is used to compensate the receiving delays between a sending node and its multiple candidate nodes.

IV. Experimental Results

The details of our simulation study and also the performance results are presented in this section. We use Aqua-Sim [16], an NS-2 based simulating software for underwater acoustic networks, to develop the simulation models of OVAR and two other recently proposed routing protocols, VBF and HHVBF.

A. Simulation setup

In our simulations, we consider the channel model described in Section III-A to simulate a lossy underwater environment. Similar to the majority of studies in this field, we use CSMA MAC protocol without using its RTS/CTS and ACK mechanism. The transmission power is set to 95 $dB\ re\ \mu\ Pa$ and transmission range for all nodes is considered as 100 meters. The data generation rate is set to 1 packet per second which can effectively prevent the interference of two continuous packets. The channel bit rate is 10 kbps and the propagation speed of acoustic signal in underwater environment is 1500 m/s. The size of packets varies by changing the number of forwarding candidates, but its average value is less than 150B. The coefficients of EPA and energy (μ,ρ) in Equation 10 are considered equal to balance energy and routing efficiency. We set T_{Delay} in Eq. 11 as 1 second based on our model. In the beaconing procedure, T_{update} is set to 30 seconds and $T_{invalid}$ is considered as 75 seconds (with considering the random jitters to prevent synchronization).

The relay nodes (ranging from 400 to 1200) are randomly deployed in a $500m \times 500m \times 1000m$ 3D field. Relay nodes can move horizontally at the speed of 2 m/s by following a Random Walk 2D mobility model (moving in X-Y plane) which is mostly used in the underwater environment. Also, we consider a single sink at location (100; 100; 0) to collect the information and a source node at location (400; 400; 1000) to generate the packets to be transferred to the sink node. Sink and source are intentionally placed in opposite corners of the field to have better assessment about routing protocols. We consider the maximum pipeline radius for VBF and HHVBF as 100 meters (equals to maximum transmission range) in which they have highest performance in packet delivery. All the results are averaged over 20 runs for randomly generated topologies with the 95% confidence interval. The simulation time for each run is set to 1000 seconds.

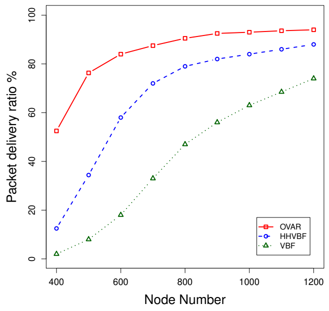


Fig. 3: Packet delivery ratio vs node density

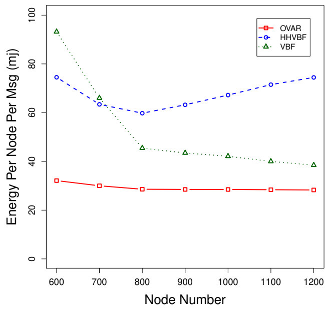


Fig. 4: Energy consumption per message vs node density

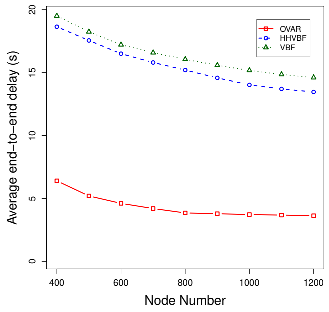


Fig. 5: Average end-to-end delay vs node density

B. *Results and Analysis*

In this section, we evaluate the performance of OVAR against those of VBF, and HHVBF in terms of packet delivery ratio, energy tax, and end-to-end delay.

Packet Delivery Ratio (PDR): It is defined as the ratio of the number of packets successfully received by the sink node to the number of packets generated by the source. The results for the packet delivery ratio in different node density are shown in Fig. 3. PDR is increased by increasing the number of nodes because it reduces the size of void areas and also the number of them. In a dense network, more forwarding nodes have this chance to be placed in the routing path and consequently PDR of routing protocols converge to a high value. On the contrary, majority of nodes in sparse networks are disconnected which leads to lower PDR. OVAR always has higher PDR than that of other routing protocols (especially in the sparse network) because it inherently excludes all the routes leading to a void area and enhance the packet delivery probability in each step towards the destination. However, in VBF and HHVBF protocols, packet failure is increased when the void area appears in their routing pipes. Also, these protocols do not take into account the packet delivery probability as a criteria for forwarding nodes selection.

Energy tax: We measure the energy tax in milli joule (mj) in terms of energy spent per node and per message to route a packet towards destination (including energy for sending mode, receiving mode and idle mode). Fig. 4 plots the energy tax for each protocol versus the number of nodes. As can be seen, OVAR consumes lower energy than other protocols for delivery of each packet to the sink. This is due to the fact that OVAR confines the forwarding candidates in a cluster (without hidden nodes) which can prevent redundant packet transmissions and collisions. However, the radius of pipe in VBF and HHVBF has a great impact on the total energy consumption and packet delivery ratio. Selecting a large radius can involve more nodes in packet forwarding; however, it increases duplicated packets which leads to more energy waste. On the other hand, lower radius causes more packet failures. In contrast to our approach, using pipelines for opportunistic routing is not able to achieve an appropriate trade-off between lower energy consumption and higher packet delivery ratio. In sparse networks, energy tax of VBF and HHVBF is high due to the low packet delivery ratio of them. In terms of normalised energy consumption, OVAR is highly efficient and achieves high delivery ratio over its consumed energy. In dense networks (while the delivery ratio has almost reached the maximum), increasing the number of nodes has little contribution on the packet delivery ratio but wastes the energy. However, OVAR mitigates this energy waste by using a principled approach to modify the forwarding set. To achieve this, in dense networks, the number of forwarding nodes is slightly increased or hold constant by OVAR, to control the energy dissipation.

Average end-to-end delay: This criteria measures the average delay time taken from the moment of creation of packets at the source until to be received by the sink for all the successfully received packets. We take into account the propagation delay, transmission delay, and holding time of packets for calculating the end-to-end delay. The average end-to-end delay for each protocol is plotted in Fig. 5. The average end-to-end delay for all protocols decreases by increasing the number of nodes because the forwarding node can find more qualified nodes in its neighbourhood. The latency of OVAR is very small in comparison to other protocols because packets almost use the optimal path towards the sink with the least possible transmissions. However, in VBF and HHVBF, nodes with better progress towards the sink may be located at the outside of the pipe and ignoring them can increase the latency. Moreover, VBF and HHVBF only give higher priority to the nodes which are close to the virtual vector (which is drawn from the source or sender to the sink), and not necessarily the nodes with lower hop count distance to the sink. Furthermore, in OVAR, each node can hold a packet with the less average holding time (by setting less amount of T_{Delay}) due to the fact that candidate nodes are closer to each other on average. However, the desirableness factor (a predefined maximum delay) of VBF and HHVBF are obviously longer than that of our method because of the different way of selection and prioritization of the forwarding nodes. In OVAR, the number of collisions and retransmissions reach the least amount possible and this improves the packet delivery time. However, in VBF and HHVBF, forwarding candidates may be located in different sides of the pipe and because of the hidden nodes, collisions will be increased at the receiver. As a result, only the packets which avoid the collisions (by using the back off process) can successfully be

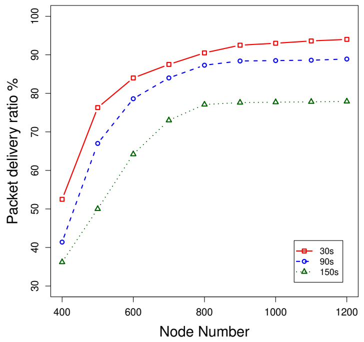


Fig. 6: Impact of beacon intervals on packet delivery ratio

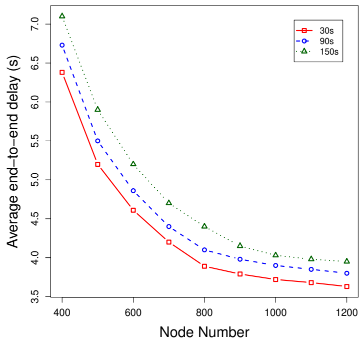


Fig. 7: Impact of beacon intervals on average end-to-end delay

delivered to the sink. Thus, the latency of VBF and HHVBF is increased by increasing the number of retransmissions due to the existence of hidden nodes in the pipe.

Impact of beaconing interval: In order to evaluate the impact of beacon intervals on OVAR performance, we conduct extensive simulations at varied beacon intervals of 30s, 90s, and 150s under the same operational condition as before. The impacts of beacon intervals on the packet delivery ratio, and average end-to-end delay are shown in Figs. 6 and 7, respectively.

As can be seen in Fig. 6, by increasing the beacon interval, packet delivery ratio is decreased because the routing and neighbourhood information gradually becomes outdated by passing the time. Furthermore, by considering the longer intervals, all the estimates about the packet delivery probabilities can become obsolete due to node movement. Thus, beacon interval should be set in a way that packet delivery ratio reaches to the highest possible value without imposing high overhead to the network. On the other hand, late updating can potentially increase the latency of received packets. This is due to the fact that packets are relayed over the non-optimal paths because the forwarding decisions are partially based on the outdated information.

V. Conclusion

In this paper, we investigated opportunistic routing in UWSNs and how it can overcome the drawback of unreliable acoustic transmission by taking advantage of intermediate nodes collaboration to relay packets. We have proposed OVAR, an opportunistic routing protocol, to minimise the number of dropped packets by efficiently bypassing void areas and also to maximise the transmission reliability where there exist serious ambient noises and channel fading. OVAR exploits the local information obtained from the periodic beaconing, construct the adjacency graph, forms the neighbouring clusters by removing the possibility of having hidden nodes in each cluster after applying a low-cost heuristic solution, and finally selects the best forwarding set. In contrast to the most of protocols reported in the field, which route packets only toward the surface, OVAR can route packets in any direction to guarantee smoothly bypassing of any type of void areas. Our simulation results have demonstrated that OVAR significantly decreases packet loss, energy consumption, and end-to-end delay in sparse to dense scenarios.

References

[1] J. Heidemann, W. Ye, J. Wills, A. Syed, and Y. Li, "Research challenges and applications for underwater sensor networking," in *Wireless Communications and Networking Conference, IEEE*, vol. 1, pp. 228–235, 2006.

[2] I. F. Akyildiz, D. Pompili, and T. Melodia, "Underwater acoustic sensor networks: research challenges," *Ad hoc networks*, vol. 3, no. 3, pp. 257–279, 2005.

[3] P. Xie, Z. Zhou, Z. Peng, J.-H. Cui, and Z. Shi, "Void avoidance in three-dimensional mobile underwater sensor networks," in *Wireless Algorithms, Systems, and Applications*, pp. 305–314, Springer, 2009.

[4] H. Yu, N. Yao, and J. Liu, "An adaptive routing protocol in underwater sparse acoustic sensor networks," *Ad Hoc Networks*, 2014.

[5] R. W. Coutinho, A. Boukerche, L. F. Vieira, and A. A. Loureiro, "A novel void node recovery paradigm for long-term underwater sensor networks," *Ad Hoc Networks*, 2015.

[6] M. Stojanovic, "On the relationship between capacity and distance in an underwater acoustic communication channel," *ACM SIGMOBILE Mobile Computing and Communications Review*, vol. 11, no. 4, pp. 34–43, 2007.

[7] U. Lee, P. Wang, Y. Noh, F. Vieira, M. Gerla, and J.-H. Cui, "Pressure routing for underwater sensor networks," in *INFOCOM, Proceedings IEEE*, pp. 1–9, 2010.

[8] M. Zorzi, P. Casari, N. Baldo, and A. F. Harris, "Energy-efficient routing schemes for underwater acoustic networks," *Selected Areas in Communications, IEEE Journal on*, vol. 26, no. 9, pp. 1754–1766, 2008.

[9] H. Liu, B. Zhang, H. T. Mouftah, X. Shen, and J. Ma, "Opportunistic routing for wireless ad hoc and sensor networks: Present and future directions," *Communications Magazine, IEEE*, vol. 47, no. 12, pp. 103–109, 2009.

[10] H. Yan, Z. J. Shi, and J.-H. Cui, "Dbr: depth-based routing for underwater sensor networks," in *NETWORKING Ad Hoc and Sensor Networks, Wireless Networks, Next Generation Internet*, pp. 72–86, Springer, 2008.

[11] P. Xie, J.-H. Cui, and L. Lao, "Vbf: vector-based forwarding protocol for underwater sensor networks," in *Networking technologies, services, and protocols; performance of computer and communication networks; mobile and wireless communications systems*, pp. 1216–1221, Springer, 2006.

[12] N. Nicolaou, A. See, P. Xie, J.-H. Cui, and D. Maggiorini, "Improving the robustness of location-based routing for underwater sensor networks," in *OCEANS-Europe, IEEE*, pp. 1–6, 2007.

[13] Y. Noh, U. Lee, P. Wang, B. S. C. Choi, and M. Gerla, "Vapr: void-aware pressure routing for underwater sensor networks," *Mobile Computing, IEEE Transactions on*, vol. 12, no. 5, pp. 895–908, 2013.

[14] K. Chen, M. Ma, E. Cheng, F. Yuan, and W. Su, "A survey on mac protocols for underwater wireless sensor networks," *IEEE Communications Surveys and Tutorials*, vol. 16, pp. 1433–1447, Mar 2014.

[15] J. Abello, P. Pardalos, and M. Resende, "On maximum clique problems in very large graphs," *AT&T labs Reserrch Technical Report: TR98*, vol. 32, 1998.

[16] P. Xie, Z. Zhou, Z. Peng, H. Yan, T. Hu, J.-H. Cui, Z. Shi, Y. Fei, and S. Zhou, "Aqua-sim: an ns-2 based simulator for underwater sensor networks," in *OCEANS, MTS/IEEE Biloxi-Marine Technology for Our Future: Global and Local Challenges*, pp. 1–7, 2009.