Wireless Netw (2017) 23:1301–1314
DOI 10.1007/s11276-016-1224-0

EMGGR: an energy-efficient multipath grid-based geographic routing protocol for underwater wireless sensor networks

Faiza Al Salti[1] · N. Alzeidi[1] · Bassel R. Arafeh[1]

Published online: 16 February 2016

Abstract Routing in underwater wireless sensor networks (UWSN) is an important and a challenging activity due to the nature of acoustic channels and to the harsh environment. This paper extends our previous work [Al-Salti et al. in Proceedings of cyber-enabled distributed computing and knowledge discovery (CyberC), Shanghai, pp 331–336, 2014] that proposed a novel multipath grid-based geographical routing (MGGR) protocol for UWSNs. The extended work, EMGGR, viewed the network as logical 3D grids. Routing is performed in a grid-by-grid manner via gateways that use disjoint paths to relay data packets to the sink node. The algorithm consists of three main components: (1) a gateway election algorithm; responsible for electing gateways based on their locations and remaining energy level (2) a mechanism for updating neighboring gateways' information; allowing sensor nodes to memorize gateways in local and neighboring cells, and (3) a packet forwarding mechanism; in charge of constructing disjoint paths from source cells to destination cells, forwarding packets to the destination and dealing with holes (i.e. cells with no gateways) in the network. The performance of EMGGR has been assessed using Aqua-Sim, which is an NS2 based simulator for UWSNs. Results show that EMGGR is an energy efficient protocol in all simulation setups used in the study. Moreover, EMGGR can also maintain good delivery ratio and end-to-end delay.

Keywords Underwater wireless sensor networks · Grid-based routing · Cell-disjoint paths · Gateway election

1 Introduction

The ocean covers more than 70 % of the Earth's surface. This enormous environment is considered a major source for food production and for providing natural resources. Thus, there is a need to explore this environment in order to make use of its valuable treasures. Underwater wireless sensor networks (UWSNs) are considered as a potential candidate for investigating the ocean. This kind of networks consists of a collection of sensor nodes that are responsible for sensing information from the surrounding and then forwarding the gathered data to sink nodes at the surface level. UWSNs can be used in a variety of applications such as environmental monitoring, undersea exploration, disaster prevention, and assisted navigation [1].

Acoustic waves are the only feasible physical layer technology for underwater networks communication. In fact, electromagnetic waves propagate for a very short range in underwater due to high attenuation and absorption at high frequency. In seawater, the absorption of an electromagnetic signal is approximately $(45\sqrt{f})dB/km$, where f is the frequency in hertz [2]. This is three orders of magnitude higher than the absorption of the acoustic signal in water. Although an electromagnetic wave can propagate for a reasonable distance at low frequency, it requires high transmission power and large antenna, which make it impractical for dense deployment of UWSNs. On the other

✉ Faiza Al Salti
m081851@student.squ.edu.om

N. Alzeidi
alzidi@squ.edu.om

Bassel R. Arafeh
arafeh@squ.edu.om

[1] Department of Computer Science, Sultan Qaboos University, Muscat, Oman

hand, optical links are not good for use in water for many reasons. First, the absorption of the optical signal in water is very high and hence, can propagate for short distances (i.e. less than 5 m [2]). Second, it suffers from scattering. Third, it requires an accurate positioning for narrow beam optical transmitter, which is difficult to provide in underwater environments. Although acoustic communication is the only suitable medium in underwater environment, it is affected by many factors which pose challenges in designing UWSNs [1, 3]. These factors include: multipath, path loss, noise, high propagation delay and Doppler spread. They make underwater acoustic channels temporally and spatially variable, and make bandwidth of the channel limited and dependent on both communication range and frequency. Specifically, less bandwidth is achieved when the communication range increases.

In addition to the above-mentioned challenges in the communications of UWSNs, acoustic channel consumes more energy than radio channel. Moreover, sensors are equipped with batteries of limited power. These batteries cannot be easily replaced or recharged due to the harsh underwater environment. Therefore, there is a need to design an energy-efficient routing protocol to prolong the network lifetime. Although, several routing protocols have been proposed [4–6] specifically for UWSNs, they suffer from high number of broadcasts which may be a burden on the sensors' energy. Grid-based routing is used widely in MANETs [7–9] mainly to overcome the broadcast storm problem that consumes high power and degrades the performance of the protocols. However, to the best of our knowledge, our previous work,MGGR, [10] is the first grid-based routing proposed for UWSNs. Although several other techniques are applied to terrestrial WSNs to conserve energy such as constructing minimum paths between nodes to transmit data [11, 12], such technique may not be applicable for underwater networks due to the use of acoustic channels and also to the dynamic nature of the environment which results in continuous topology changes.

The main aim of this paper is to extend MGGR and evaluate the extended version. Basically, the extension includes:

1. Providing a more detailed description about the proposed routing protocol.
2. Modifying the gateway election mechanism to further reduce the overhead and preserve energy.
3. Modifying the routing paths.
4. Presenting performance comparison results between EMMGR and Vector Based Forwarding (VBF) [4]. The performance evaluation metrics include studying the effects of network density, traffic load and nodes mobility on packet delivery ratio, end-to-end delay and energy consumption.

The rest of this paper is organized as follows. Some location-based routing protocols proposed for underwater wireless sensor networks are summarized in Sect. 2. Section 3 describes the proposed EMGGR protocol and Sect. 4 presents the experimental analysis of the protocol. Finally, Sect. 5 concludes the paper.

2 Related work

Underwater sensor network has been under research for several decades and researchers tackle different network layers. For example, [13, 14] proposed models for underwater acoustic propagation loss. Whereas [15, 16] and [17, 18] tackle the MAC and application layers, respectively.

Regarding to the routing layer, which is the main focus in this paper; several routing protocols for underwater wireless sensor networks have been proposed. Those protocols can be classified into two categories namely, location-based and location-free protocols. Location-based routing protocols are based on the assumption that sensor nodes are equipped with a localization service to obtain location information. On the contrary, location-free routing protocols are those that do not depend on any location information to route packets. In this section, we investigate some of the location-based routing protocols.

In [4], a vector-based forwarding (VBF) routing protocol is proposed. In this protocol, data packets are forwarded in a pipe with a given radius where the pipe is halved by a vector from the source node to the sink node. Only, nodes within the pipe are responsible for relaying data packets to the sink. Although the protocol limits the number of forwarders of the data packets to those in the routing pipe which reduces the overall overhead, its performance is highly affected by the width of the pipe. If the width is small, then, there might be no nodes in the pipe to forward the packets. If it is large, then the number of forwarders increases which increases the number of retransmissions of the packets that might interfere with each other. Moreover, the energy consumption increases by increasing the density of the network and/or the radius of the routing pipe due to the increase in the number of nodes involved in packet forwarding. Enhancement of the protocol is proposed in [19] in which a virtual pipe per hop is used instead of a single pipe from the source to the sink. Nodes located in the pipe from the previous forwarder to the sink are considered as candidates for relaying the data packets. The approach produces better results in sparse networks than VBF. However, for dense networks, the number of candidate nodes for forwarding the packets increases which increases the traffic and the overhead in the network, and increases the energy consumption. In addition, the performance, as in VBF, depends on the width of the routing pipe which needs to be carefully selected.

Jornet et al. [5] proposed focused beam routing (FBR) protocol. FBR assumes that for each node, there is a finite number of increasing power levels P_1 through P_N. The lowest power level is used for the initial transaction. The current forwarder should first multicast a *REQUEST_TO_SEND* (*RTS*) control packet containing its location and the location of the destination at the lowest power level P_1. Nodes located within a cone of angle $\Theta/2$ originating from transmitter towards the destination are considered as candidate nodes and hence, each should respond with CLEAR_TO_SEND (*CTS*) control packet. If a node does not hear any response after waiting a round trip time, it will send the *RTS* with an increased power level P_2. This process continues until the node receives a response or all power levels exhausted. If all power levels ran out with no response, it shifts its cone and looks for candidate left and right of the main cone. However, if it receives response(s), it will send the data packet to the closest node to the destination. The receiver of the packet will follow the same process to deliver the data packet to the next hop and so on until the packet reaches its destination. The cone aperture Θ plays an important role in the performance of FBR. In a dense network, having a small Θ is a better choice to reduce the number of nodes that will be involved in forwarding packets. However, in a sparse network, large Θ is preferable. The exchange of the control packets at each hop of the forwarding process consumes high energy and can degrade the performance of the protocol under high traffic conditions.

Chirdchoo et al. [6] proposed a sector-based routing with destination location prediction (SBR-DLP). The area around the current forwarder is divided into a number of sectors where the first sector is halved by the vector from the forwarder to the destination. The other sectors are labeled according to their angular differences from that vector. When the current forwarder has a packet to send, it broadcasts a *Chk_Ngb* packet. Nodes closest to the destination than the current forwarder, reply with a *Chk_Ngb_Reply* packets. Upon receiving the *Chk_Ngb_Reply* packets, the node discards those nodes that might move out of its range before being able to acknowledge the reception of its data packet. Then, the closest node to the destination from those remaining candidates is selected as the next forwarder. Although the protocol prevents flooding of the data packet over the whole network by selecting a single candidate in each hop which reduces the energy consumption, the control packets exchanged during the forwarder selection can incur extra overhead in dense or high traffic network. In addition, SBR-DLP assumes that the destination's movement plan is known to all nodes, which reduces the flexibility of the network since the destination might deviate from its scheduled movement due to water currents.

In [20], a directional flooding-based routing (DFR) protocol is proposed to improve the reliability of UWSNs. DFR controls the flooding zone (i.e. zone in which a packet should be broadcasted) using the link quality of the neighboring nodes of each forwarder. To achieve this, it employs two types of angles namely the *CURRENT_ANGLE* and the *REFERENCE_ANGLE*. The *CURRENT_ANGLE* of a node F (CA_F) is the angle formed by the vectors $\overrightarrow{FS}$ and $\overrightarrow{FD}$ where S and D are the source and the sink, respectively. The *REFERENCE_ANGLE* is an angle specified by the previous forwarder P (RA_P) and used by the receiving node to determine whether it is a suitable forwarder candidate. When a source node S has a packet to send, it includes an initial reference angle (RA_S) equal to a predefined minimum value *A_MIN* and broadcasts the packet. A node (e.g. F) upon receiving this packet calculates its CA_F and compares it with the RA_S included in the packet. If the CA_F is smaller than the RA_S, it simply discards the packet. Otherwise, it computes a new *REFERENCE_ANGLE* (RA_F) based on the average link quality to its neighbors and retransmits the packet. This process continues until the packet reaches its destination. The protocol limits the void problem (i.e. when there is no candidate for forwarding the data packet to the next step) by ensuring that at least one forwarder is selected in each step. However, its performance depends significantly on a number of predefined values (e.g. A_MIN) which need to be carefully selected.

4-chain, 2-chain and single-chain based routing schemes for cylindrical networks is proposed in [21] to improve network lifetime and throughput. The protocol consists of two main phases namely; initialization phase and protocol operation phase. In the initialization phase, nodes broadcast their location information to be used in forming chains and the optimal paths. In the protocol operation phase the chains are formed based on the location information starting from the farthest node from the sink. Local optimal path between nodes in each chain is constructed. Then, each chain is connected to the nearest node in the next chain forming a global optimal path. Finally, data packets are transmitted through the global optimal path. Simulation results show that the 4-chain routing scheme performs better than its counterparts. However, the proposed protocol is not suitable for mobile underwater networks because nodes continuously change their position, and hence, location information need to be rebroadcasted over and over again in order to reconstruct the chains and optimal path which incurs extra overhead and drains the energy.

EMGGR is proposed to reduce the energy consumption caused by the broadcast storm and by the overhead of exchanging control packets in each hop of relay selection. The protocol is described in the next section.

3 EMGGR: the proposed approach

The proposed EMGGR routing protocol assumes that the geographic area of the network is partitioned into 3D logical grids as shown in Fig. 1. Each grid or cell is a cube of size d^3, where d is the length of the cell side. The grid cells are identified by their XYZ-coordinates. The sensor nodes are assumed to be deployed at different depths and are either attached to surface buoys or anchored at the bottom of the ocean. Without loss of generality, we assume that there is only one sink node and it is fixed at the top surface of the grid. Deploying more sinks will not affect the design of the protocol as will be explained later. Data packets are forwarded in a grid-by-grid manner via gateways through disjoint paths.

Each sensor node is identified by a unique number (*NID*) and equipped with a localization service from which it can obtain its current location. It also knows the cell in which the sink node is located. From the location information, each node is able to obtain the grid coordinates it belongs to which is then mapped to a unique number (*CID*). The *CID* of a cell is calculated as follow. The grid origin is assumed to be at the bottom left corner of the grid and the cell numbering adopted in the protocol starts from that position (The origin in Fig. 1 is at the right corner of the Grid for easy visualization). The *CID* of the cell at the origin (i.e. with x = 0, y = 0, and z = 0) is assumed to be equal to 0. We then move right in the X direction and increment the number. After consuming all the cells in the X direction, we move to the next cell in the Y direction and so on. After consuming the cells in the X and Y directions, we move up to the next Z and we do the same thing. We do this until we number all the cells in the monitored volume. For example, assume that the xyz-Grid coordinates of a cell i is x_i, y_i and z_i, respectively, then, its CID_i can be computed as given in (1):

$$CID_i = x_i + k * (y_i + k * z_i) \tag{1}$$

where k is the number of cells in each dimension.

Any cell y that has a common face, side or vertex with another cell x is considered a neighbor cell to x. Therefore, for each cell x there are up to 26 neighboring cells (eight cells in the same z-level as x, nine cells in the upper z level and nine cell in the beneath z level). The farthest distance between two points in two neighboring cells is when they meet in two opposite corners, and to ensure a communication between nodes in these points, the value of d (i.e. the length of the cell side) as depicted in Fig. 2 is selected such that it satisfies the condition $d \leq R/(2\sqrt{3})$, where R is the transmission range of sensor nodes.

EMGGR consists of three main components: (a) the gateway election mechanism (b) the updating gateways' information mechanism and (c) the packet forwarding mechanism which will be described in the following subsections.

3.1 Gateway election mechanism

Gateway Election Mechanism deals with electing sensor nodes as gateways to be responsible for relaying data packets to the destinations. At any point of time, there is at most one gateway in each cell. Since energy is one of the most critical resources in underwater wireless sensor networks and because transmission is the most expensive activity performed by sensor nodes compared to the computation cost, reception cost and idling cost [22], it is very

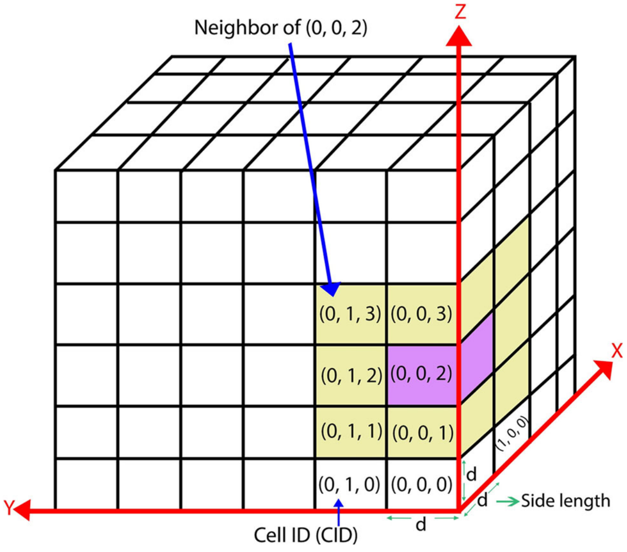


Fig. 1 A 3D logical grid view

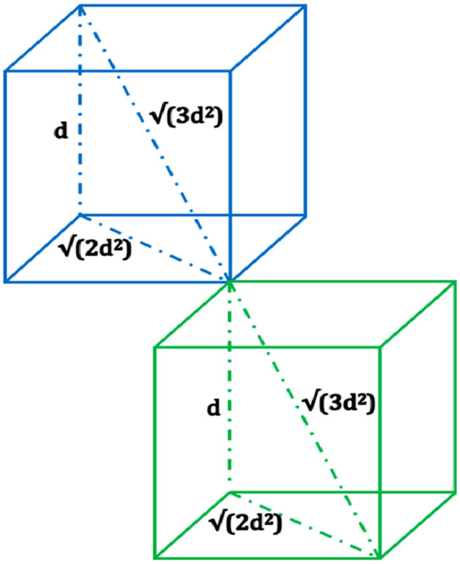


Fig. 2 Finding the length of cell side

important to design an energy efficient gateway election algorithm so that a gateway remains in a cell for a long time while balancing the energy among nodes in a cell. Although keeping the gateway longer can cause fast depletion of its energy, it reduces the election overhead. As nodes move in the network and their energy levels change, non-gateways will eventually take turns in acting as gateway nodes and hence, that reduces the risk of depleting the energy of the gateway nodes if they remain as gateways for long time. The gateway election is based on a weighted value called an election weight. Every node i in the network keeps track of its election weight w_i which is calculated as given in (2).

$$w_i = \alpha\left(\frac{maxDist - dist_i}{maxDist}\right) + (1-\alpha)\left(\frac{E_i}{maxEnergy}\right) \qquad (2)$$

where $dist_i$ is the distance from the center of the cell to node i, E_i is the remaining energy level of node i, and α is a predefined factor (between 0 and 1 inclusive) to decide the weight of each term of the above expression. Choosing small values for α will give more importance to the energy level while choosing large α values means that being close to the center is the dominating factor in the election of the gateway nodes. The node with the highest election weight will be elected as the cell gateway. If there is a tie, then the node with highest *NID* is selected. *maxEnergy* is the initial energy given to the sensor node and *maxDist* is the farthest distance that the node can be from the center of the cell and it is calculated as

$$maxDist = \frac{d\sqrt{3}}{2} \qquad (3)$$

The gateway election algorithm consists of an initialization phase and a maintenance phase as described below.

3.1.1 Gateway initialization phase

- Every node i broadcasts a local *GW_Election(NID, CID, w_i)* packet, where *NID* is the node ID, *CID* is the ID of the cell where node i is located, and w_i is the election weight of node i. During the election process, nodes keep track of the current possible gateway by storing its *NID* and election weight in a variable called *current_candidate*. When a node sends a *GW_Election* packet, it sets itself as the current candidate. In addition, nodes that send that packet should wait for a predefined amount of time called an *election period* to receive replies from nodes with higher weights.
- Upon receiving a *GW_Election* packet from a local node, nodes check if there is a candidate in the *current_candidate* variable. If there is a candidate, then the nodes compare the election weight stored in the received packet with the election weight of the *current_candidate*, and the variable gets updated if the received weight is greater than the weight stored in it. If there is no candidate's information stored yet, then nodes compare their election weight with the received weight. If a node has a higher weight, it sets itself as a possible candidate and replies with a *GW_Candidate (NID, CID, w_i)* packet, and it should wait for an *election period* to receive replies from nodes with higher weights. Otherwise, it sets the *NID* in the received packet as the possible candidate. If two nodes have the same election weight, then the node with the highest *NID* is elected. Algorithm 1 shows the actions taken by a node upon receiving a gateway election packet.
- When the *election period* of a node expires, it checks the *NID* stored in the *current_candidate*. If the *NID* is its own ID, then it declares itself as the gateway by storing its information in its gateway table and broadcasting a *GW_New(NID, CID, w_{max})* packet, where w_{max} is its election weight. Also, it should set a timer called *GW_Update_Timer* to periodically inform other nodes about its existence.
- Upon receiving a *GW_New* packet, local and neighboring nodes store the gateway information in their gateway tables. Also, local nodes should set timers called *GW_Existence_Timers* to periodically check the existence of the elected gateway in the cell.

```
Procedure recvGwElectionPkt(Packet* p<NID, cid, weight>) {

  if the packet is from a local cell  then
    myWeight = calculateWeight()

    if currentGateway == NULL then
      if I'm a better candidate than NID then
        currentGateway = myself
        send GW_CandidatePacket
        wait for better candidate
        if there is no better candidate then
          send gateway new packet to
            elect myself as a gateway for the coming period
      else
        currentGateway = NID

  else
    if NID is a better candidate than currentGateway then
      currentGateway = NID
}
```

Algorithm 1: Actions taken by a node upon receiving a GW_Election packet

3.1.2 Gateway maintenance phase

- The current gateway periodically checks its remaining energy. If its remaining energy is greater than or equal to a predefined value called *energy_threshold*, it calculates its new election weight and broadcasts a *GW_Update(NID, CID, w_{max})* packet. In addition, it continues to serve as a gateway for the grid cell for the coming period. If, however, the remaining energy is less than the threshold, it silently removes itself from the gateway table without sending any packet. Algorithm 2 summarizes the actions performed by a gateway when its gateway period expires.

- Local and neighboring nodes upon receiving a *GW_Update* packet update their gateway tables. Moreover, local nodes reset the *GW_Existence_Timer* for the next period.
- If no *GW_Update* packet was received after the *GW_Existence_Timer* expires, nodes start a new election process by broadcasting *GW_Election(NID, CID, w_i)* packets as described in the initialization phase.
- If a gateway node roams out of its grid cell, it broadcasts a *GW_Exit (NID, CID, w_{max})* packet. Upon receiving the *GW_Exit* packet, local and neighboring nodes remove the gateway record of that gateway from their gateway tables. In addition, local nodes start a new election process as described in the initialization phase.

```
Procedure GwTimeout(){

   if energy_ < energy_thresh then
      weight_ = 0.0;
   else
     myWeight = calculateWeight()

    if myWeight _>0.0 then
        sendGwUpdatePkt();
        reschedule gwTimeoutTimer_
    else
           currentGateway = NULL
           removeMyRecordfrom GwTable
}
```

Algorithm 2: Actions taken by a gateway when GwPeriod expires

3.2 Updating gateway's information

This part of the protocol deals with how nodes update gateways' information in their local and neighboring cells. Every sensor node in the network maintains a table of gateways (*GW_Table*) in the local and neighboring grid cells. Every record in that table stores information about a single gateway and it includes the node's identifier (*NID*), its election weight (w_{max}), and the ID of the cell (*CID*) it is located in. The table gets updated in the following cases:

- A *GW_Existence_Timer* expires without receiving a *GW_Update* packet from the local gateway. In this case, local nodes should remove the record from their tables and start a new election process. The cell is considered as a hole (i.e. without a gateway) in the network until a new node is elected.
- Upon receiving *GW_New(NID, CID, w_{max})* packet or *GW_Update(NID, CID, w_{max})* packet, local and neighboring nodes should add a record for the gateway in their tables.
- Upon receiving *GW_Exit(NID, CID, w_{max})* packet, local and neighboring nodes should remove the record from their tables and this cell will be a hole in the network until another gateway is elected.

3.3 Packet forwarding mechanism

The packet forwarding mechanism is further subdivided into two parts, namely, the construction of disjoint paths and forwarding of the data packets through those paths as described below.

3.3.1 Construction of disjoint paths

Cell disjoint paths are those paths that join only in the source and destination cells. They can be used to gain high reliability by sending multiple copies of the same packet in parallel, to reduce the end-to-end delay by dividing a large message into small messages and then sending these messages over the disjoint paths, or to find alternative paths when a current path is broken [23, 24]. The proposed EMGGR protocol uses disjoint paths to find alternative paths to route data packets when a hole (i.e. a cell with no gateway) is encountered in the current path. Disjoint paths for 2D and 3D mesh networks are well defined and well proven in the literature [23, 24].

In [23], it is assumed that each cell has at most eight neighboring cells corresponding to the eight possible starting moves (i.e. right, left, up, down, right up, right down, left up, and left down). Therefore, there are at most eight possible disjoint paths that can be constructed from any cell. The paths are constructed based on the geographic information. Please refer to [23] for more information on how these paths are constructed. Although, the paths for 3D grid have been constructed in [24], the construction, maintenance and storage of these paths are complicated, since every node will have 26 neighbors in three dimensions. Thus, we have developed a new technique for constructing the disjoint paths based on those constructed for 2D grid [23] as described below and illustrated with Fig. 3:

- *Phase 1:* In the first phase, the destination cell is projected to a virtual cell in XY-surface with z value equal to that of the source cell.
- *Phase 2:* In the second phase, data packets are routed from the source cell to a cell just before the virtual destination (i.e. the neighbors of the virtual destination) using disjoint paths defined for 2D grid as proposed in [23]. Note that there can be three to eight disjoint paths between any two cells of a 2D mesh depending on the coordinates of the cells. We however, will limit the number of disjoint paths to a maximum of five in order

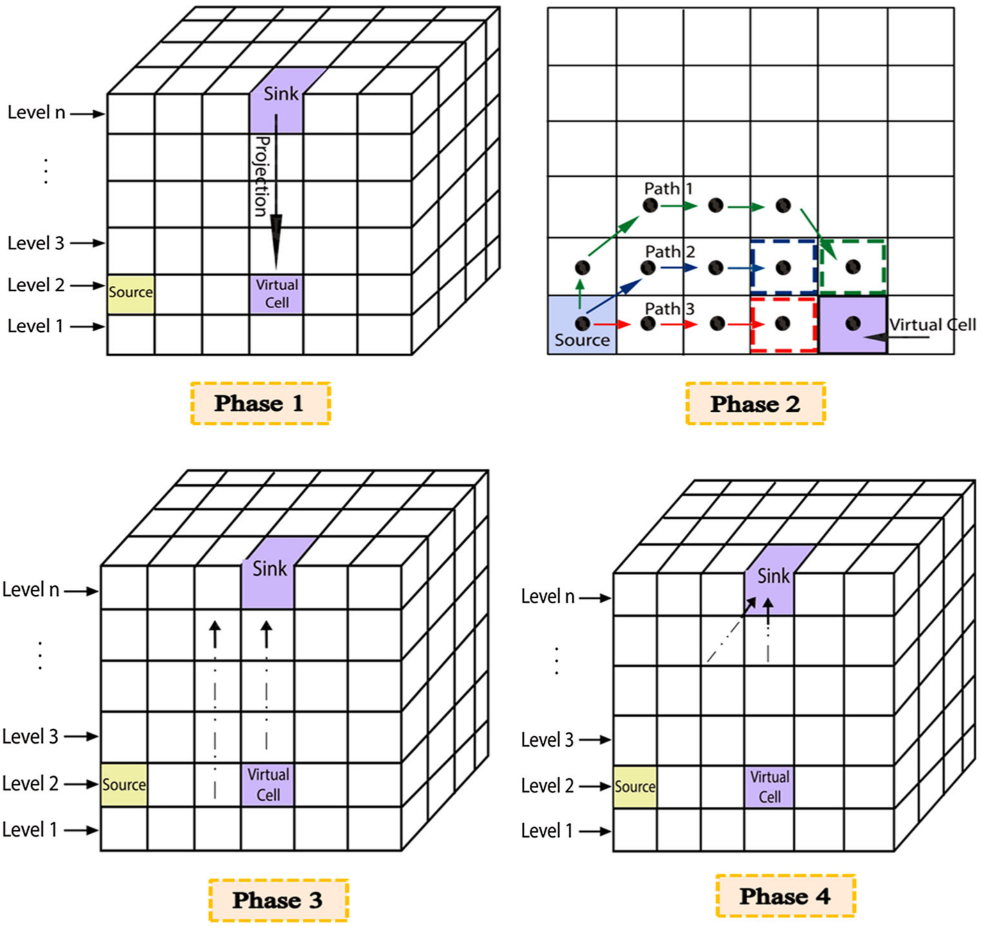


Fig. 3 Packet forwarding mechanism

to avoid long paths that may substantially increase the end-to-end delay.

- *Phase 3:* The packet is then routed vertically (i.e. in the same column reached in the previous phase) until it reaches one of the neighboring cells of the sink node in the z level beneath the surface level.
- *Phase 4:* Finally, the packet is forwarded from that cell to the actual destination cell through a one-hop move.

Since only gateways are responsible for forwarding data packets, the paths are constructed only by gateways, and once they are constructed, they will be stored in the gateway as long as the node acting as a gateway for the corresponding cell. The paths are stored in a table called *Path_Table* along with an indication of the path validity (i.e. 1 indicates a valid path, 0 indicates an invalid path). For example, *1: 20 23 30 40 58*, is a path crosses the cells 20, 23 30, 40 and 58, and it is a valid path. The proposed protocol uses source routing in which the entire path is stored in the packet's header, and the next hop can be easily determined by looking at the packet's header. Also, it is worth mentioning that the construction of paths does not pose any constraints about the exact position of the sink in the surface, and the number of sinks used. Thus, any number of sink nodes can be deployed at the surface depending on the requirements of the application.

3.3.2 Forwarding data packets

When a source node generates a data packet to be sent to the destination, it reads its gateway table to look for a local gateway to forward the packet. If there is no gateway in the local cell, it drops the packet. Otherwise, if the gateway ID is its own ID, then it considers itself as the source gateway of the packet. If the address is not its ID, it forwards the packet to that gateway and will be considered as the source gateway. The source gateway will select a path from the available valid paths in a round robin fashion. After finding a valid path, the source gateway adds the path (sequence of cell IDs) to the packet header and forwards the packet to the next hop. Then, the gateway in the next hop checks if there is a gateway in the following hop, and if so it forwards the packet to that gateway. Otherwise, it acts as the source gateway of the packet and tries to forward the packet through one of its valid paths. If all paths are invalid, it sends a negative acknowledgment to the previous hop which tries to forward the packet from its cell to the destination. If however, no valid path is found from the intermediate cells, the packet will be eventually propagated back to the source gateway which indeed marks the path as invalid and searches for another path to forward the data packet. If all paths in the source cell are invalid, the current data packet is dropped.

Figure 4 demonstrates further the forwarding process. The example shows a 2D grid for easy visualization, but the idea applies to 3D grid. Assume that the source gateway S has a data packet to be transmitted to the destination D. S searches its path table and finds that the path $\pi_1 = 1, 2, 3, 4, 5$ is the valid path to be used next. The node S includes the path in the packet header and transmits it to the gateway in cell 2. The different cases are listed below:

- The perfect case is when there is a gateway in every cell (i.e. cells 2, 3 and 4) that the path crosses. The packet propagates through this path until it reaches the destination as shown in the left grid of Fig. 4.
- The packet reaches the gateway in cell 2 and it tries to forward the packet to cell 3, unfortunately, there is no gateway in cell 3. The gateway in cell 2 tries to look for a path from its table; however, all paths are invalid. In this case, it should send a negative acknowledgment back to the source gateway S in cell 1. S upon receiving the negative acknowledgment marks the path π_1 as an invalid path, and searches for another path. If all paths are invalid, it drops the packet; otherwise, it forwards the packet through the new path.
- Assuming now that the packet reaches cell 3 successfully, the gateway in that cell has information that cell 4 is a hole, it acts as the source gateway and searches for another path in its table and finds that the path $\pi_2 = 3, 6, 7, 5$ as shown in the right grid of Fig. 4 is valid. It replaces the old path in the packet header with this new path and forwards the packet to the gateway in cell 6. If everything goes fine with this path, the packet will arrive to the destination. If a negative acknowledgment is sent back to the gateway in cell 3 and there is no other valid path, the packet gets dropped.

4 Performance evaluation

4.1 Simulation setting

We have used Aqua-Sim [25] as the simulation package to evaluate our approach and compare with the basic VBF [4] routing protocol. We assume that the network size is 264 × 264 × 264 m^3 and the number of nodes varies between 100 and 500 nodes. One sink node is positioned at the top middle of grid surface. The simulation type adopted in this evaluation is based on the terminating state, where each run lasted for 1000 s. Results from the first 150 s and the last 100 s are discarded to minimize the warm-up effect. Since most underwater applications assume 2D horizontal movements of sensor nodes [4], the simulation study in this research assumes that nodes are randomly deployed in the topology and they follow the 2D random walk [26, 27] mobility model which is one of the most widely used mobility models [26]. The batch means method was used to collect simulation results where each point is obtained from the average of 25 simulation runs. In each run, 10 randomly selected sensor nodes where used to inject exponentially distributed traffic into the network. 95 % confidence intervals were calculated for all collected results.

A more comprehensive list of simulation parameters is presented in Table 1. The values of the general parameters are similar to those used in previous studies [28, 29].

4.2 Performance metrics

The performance of the protocol is assessed by investigating the following performance metrics:

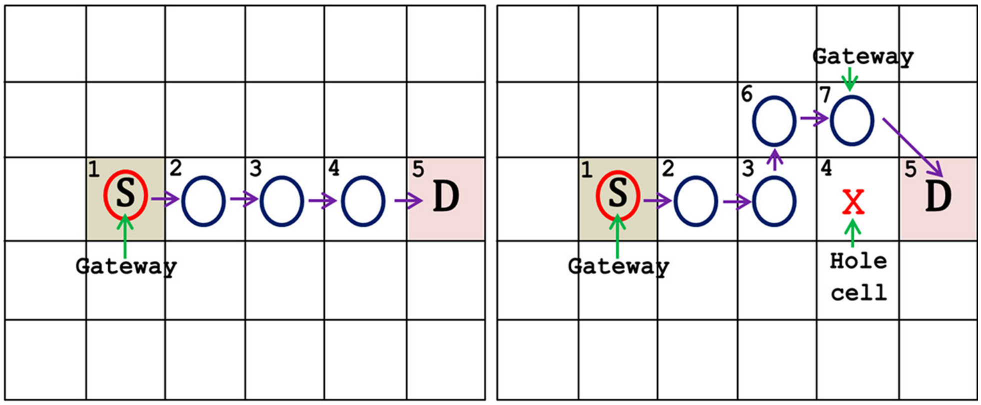


Fig. 4 Different states that the packet may get into while travelling to the destination

Table 1 System parameters

Parameter	Value (s)
Channel	Underwater channel
Propagation	Underwater propagation
MAC type	Broadcast MAC
Initial energy	10,000 J
Cost of transmission	2.0 W
Cost of reception	0.75 W
Idle power	0.01 W
Transmission range	150 m
Spreading type	Spherical spreading
Packet size	200 bytes
Control packet size	20 bytes
The value of d	44 m
k (cells in each dimension)	6
Nodes position update interval	0.3 s
Gateway update period	(44 & 66) s
Checking node roaming period	0.3 +(5 × 10^{-9}) s
Waiting for qualified node period	7.3 s
Resetting fault path period	22 s
Gw_Existance_Timer	50 s
Energy_Threshold	20 J
Bandwidth	5 × 104 bps
Width of routing pipe for VBF	50 m

- *Packet Delivery Ratio (PDR):* PDR is the ratio of the number of distinct data packets that are successfully delivered to the sink nodes to the total number of data packets generated at the source nodes [30]. Formally, it can be written as:

$$PDR = \frac{\sum Number\ of\ packets\ received}{\sum Number\ of\ packets\ send}$$

- *Average End to End Delay:* The average time taken by a data packet to arrive to the destination [30]. It is computed from the time a packet is generated until it reaches the destination. Only the data packets that were successfully delivered to destinations are counted. Mathematically, it can be calculated as:

$$Delay = \frac{\sum_{i=1}^{pkts} Arrival\ time_i - Send\ time_i}{pkts}$$

where *pkts* is the number of packets successfully delivered to destination.

- *Energy Consumption:* The total energy consumed by all nodes during the simulation [30]. It includes the transmission power, the reception power and the idling power consumed by all nodes:

$$Total\ Energy = \sum_{i=1}^{N} tranmission\ power_i + reception\ power_i + idling\ power_i$$

where N is the total number of nodes used during the simulation.

It is obvious to notice that higher packet delivery ratio, lower end-to-end delay and lower energy consumption are desirable performance measures that routing protocols strive to achieve.

4.3 Simulation results

The above performance metrics (packet delivery ratio, end-to-end delay and energy consumption) have been assessed under different operating conditions which are network density, traffic load and node's mobility.

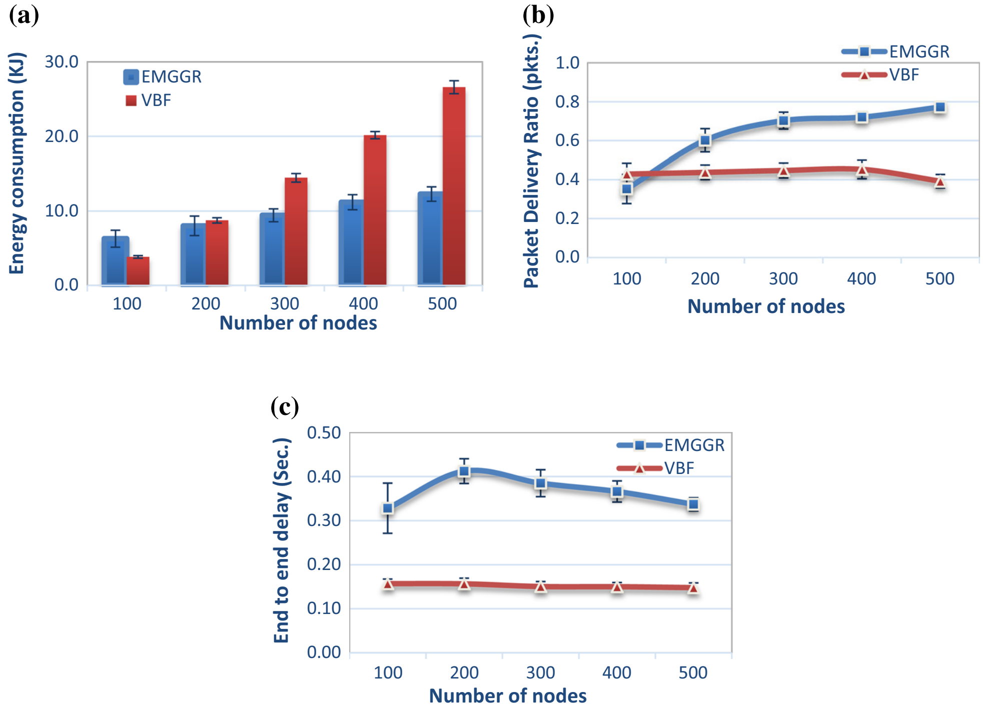
4.3.1 Effect of network density

In this set of simulations, we investigate the effect of network density on both EMGGR and VBF protocols. We vary the number of nodes from 100 to 500. Nodes are assumed to be stationery. In this work, 10 randomly selected sources are used to inject traffic, where the traffic injection time is exponentially distributed with mean 0.1 s. The value of alpha used in calculating the election weights of the nodes (see expression 2) is set to 0.5.

Figure 5(a) shows the effect of network density on the total energy consumption. As shown in the figure, the two protocols exhibit a similar trend with increasing the number of nodes. As the number of nodes increases, there is an increase in the energy consumption. However, VBF consumes more energy because as the number of nodes increases in the network, the number of qualified nodes to forward the packets increases and hence, the number of broadcast packets increases. Also, the number of nodes being able to receive the transmitted packets increases. On the contrary, in EMGGR, data packets are forwarded only through disjoint paths and only gateways can participate in the process. Specifically, when the number of nodes is 400, EMGGR can save almost 45 % of the energy better than VBF.

Figure 5(b) shows the effect of network density on packet delivery ratio. Unsurprisingly, we observe a high increase in the packet delivery ratio as the number of nodes increases in EMGGR. As the number of nodes increases in the network, the probability that each cell contains a gateway increases and hence, the probability of path breakage decreases and data packets can be successfully delivered to the destination. For the VBF, as the number of

Fig. 5 Effect of network density on the: **a** total energy consumption, **b** PDR, **c** average end-to-end delay



nodes increases, there is a slight increase in the data packets that are successfully delivered to the destination since more nodes are located within the routing pipe of the VBF. We even observe a decrease in the packet delivery ratio of the VBF when the nodes exceed 400. This is due to the expected high number of packet collisions that result from the increase in the number of packets propagated in the network. When the number of nodes is 400, EMGGR is better than VBF in delivering data packets by 27 %. Also, it can be seen from the figure that the delivery ratio of EMGGR reaches 77 % when the network density approaches an average of 2.3 nodes per grid cell.

Figure 5(c) shows the effect of the network density on the average end-to-end delay. The figure shows that the VBF routing protocol causes less average end-to-end delay than EMGGR, because of the higher number of hops in EMGGR than VBF. In EMGGR, a data packet is first forwarded horizontally and then moved upward, while in VBF, a packet is forwarded in a vertical pipe between the source and destination. Generally, as the number of nodes increases the average end-to-end delay decreases in EMGGR since cells are more likely to contain gateways that are responsible for forwarding data packets to the destination. The delay is less in EMGGR when the number of nodes is 100 because the probability that the cell is filled with gateways is very small and hence path breakage is discovered earlier and the packets get dropped. Then, that probability starts increasing with increasing network density. When the number of nodes is 200, the delay reaches its largest, then it starts decreasing because the cells are filled with gateways and packets can get delivered to the destination without the need to re-route the packets. In VBF, however, the difference in the delay is insignificant as the number of nodes increases because packets are forwarded in the same routing pipe.

4.3.2 Effect of traffic load

The impact of traffic load is assessed by varying the mean of the exponential distribution from 0.05 to 0.2 s. 400 nodes are randomly deployed in the area and they are assumed to be stationary. The value of alpha used in calculating the election weights of the nodes (see expression 2) is set to 0.5.

Figure 6(a) shows the effect of traffic load on the total energy consumption. The figure shows that the energy consumption is proportional to the traffic load in both protocols. VBF protocol seems to consume higher energy than EMGGR because of the increased packets that are propagated in the network which lead to more delay and hence, more energy consumption. Specifically, EMGGR saves on average 62 % of the energy better than VBF over all used traffic. This is again due to the efficient forwarding mechanism employed by EMGGR.

Figure 6(b) shows the effect of traffic load on the packet delivery ratio. From the figure, it is observed that the packet delivery ratio decreases with the increase in the packet generation rate in both EMGGR and VBF. By

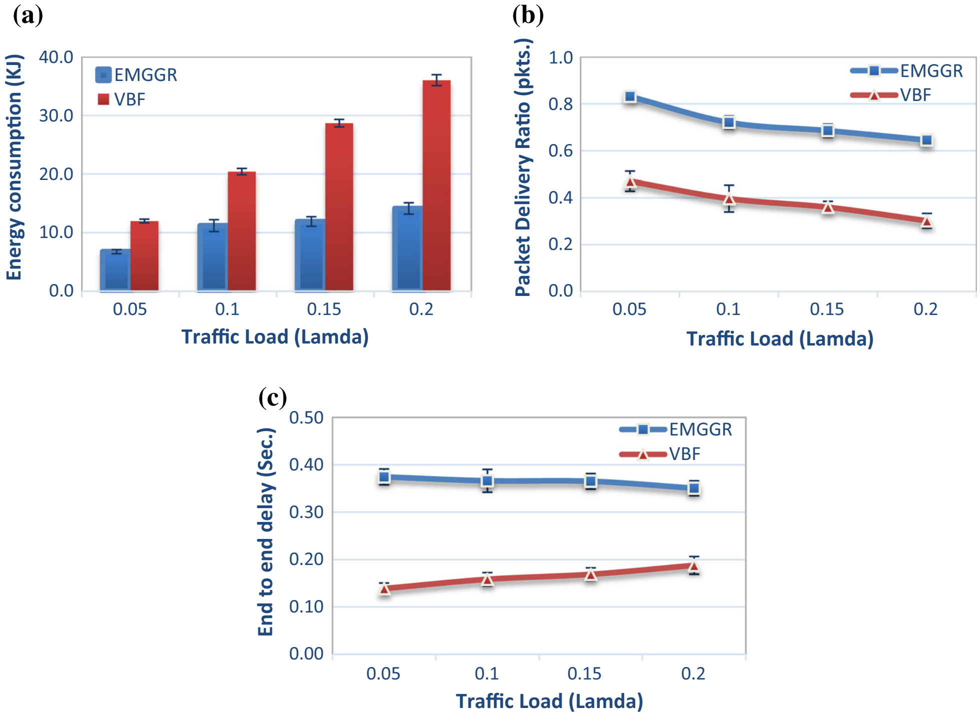


Fig. 6 Effect of traffic load on the: **a** total energy consumption, **b** PDR, **c** average end-to-end delay

increasing the packet generation rate, more packets are propagated in the network and hence, more packets collide and get dropped. Nevertheless, EMGGR outperforms VBF under all traffic rates. This is mainly due to the availability of the disjoint paths, and the ability of the algorithm to deal with holes and to prolong the paths lifetime. It can be seen from the figure that EMGGR improves the packet delivery ratio by an average of 34 % over all used traffic rates compared to VBF.

Figure 6(c) shows the effect of traffic load on the average end-to-end delay. The figure illustrates that the two protocols exhibit different trends and EMGGR incurs higher delay than VBF because of the higher number of hops in EMGGR than VBF. For VBF, as the traffic load increases, the packets propagated in the network increase and hence, packets get delayed. In EMGGR, the change in the average end-to-end delay is insignificant because packets are routed on different paths.

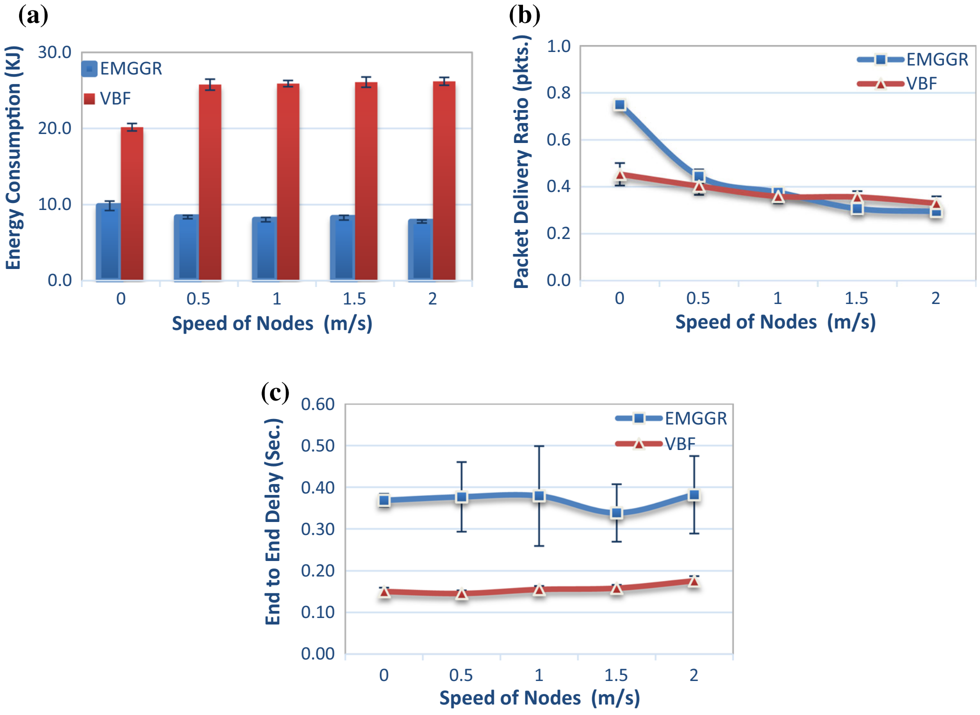
4.3.3 *Effect of node mobility*

The effect of node mobility is assessed by varying the maximum speed of the nodes from 0.0 to 2.0 m/s. The minimum speed is fixed to 0.0 m/s. The number of nodes used in this set of simulation is 400 nodes and are deployed randomly in the network. The mean of the exponential distribution is set to 0.1 s. The value of alpha used in calculating the election weights of the nodes (see expression 2) is set to 0.8 to favor gateways that are close to the center of the cells.

Figure 7(a) shows the effect of node mobility on the total energy consumption. The figure shows that as the speed of nodes increases, the total energy consumed in EMGGR slightly decreases because of the increased number of holes in the network which results in dropping packets earlier. In VBF, the energy consumption increases significantly once nodes start moving. The reason is that, when nodes are static, the number of nodes that remain idle is large compared to the number once nodes start moving and the idling state consumes less power. The figure shows that VBF consumes almost three times more energy than EMGGR (when speed is 1 m/s, VBF consumes 25.898 KJ while EMGGR consumes only 8.045 KJ).

Figure 7(b) shows the effect of node mobility on the packet delivery ratio. The figure shows that the packet delivery ratio is inversely proportional to the nodes' mobility. In EMGGR, when the maximum speed of the nodes changes from 0 to 0.5 m/s, the packet delivery ratio is sharply decreased by 41 % because the period that the gateways stay in the same cells is short and hence, the election starts over again. The sharpness decreases as the speed increases. In VBF, there is a little decrease in packet delivery ratio compared to EMGGR because nodes continue to leave and enter the routing pipe and because any node in the pipe can forward data packets. EMGGR has better packet delivery ratio until the speed reaches 1 m/s.

Fig. 7 Effect of node mobility on the: **a** total energy consumption, **b** PDR, **c** average end-to-end delay



When the speed is greater than 1 m/s, VBF gives slightly better packet delivery ratio than EMGGR.

Figure 7(c) shows the effect of node mobility on the average end-to-end delay. Generally, in both protocols, the average end-to-end delay increases marginally by increasing the nodes' mobility except in EMGGR when the maximum speed is set to 1.5 m/s. The increase in the delay in EMGGR is because packets are propagated through longer paths due to the increase in the number of holes in the network.

5 Conclusion

In this paper, we proposed a multipath grid-based geographic routing protocol, called EMGGR, for UWSNs. EMGGR consists of three main components: a gateway election mechanism, a mechanism for updating gateway's information and a data packet forwarding mechanism. Data packets are transmitted through disjoint paths via gateways. These paths help in increasing the reachability of the network. The performance of the protocol has been assessed via simulation using the Aqua-Sim simulator. A number of experiments have been conducted under various operating conditions (network density, traffic load and mobility). Generally, the simulation results show that EMGGR is an energy efficient protocol compared to the Vector Based Forwarding (VBF). In addition, EMGGR can maintain a good delivery ratio while exhibiting higher delay compared to VBF. Therefore, EMGGR is more suitable for applications that care about energy consumption and packet delivery ratio.

EMGGR can be considered as a base for grid-based routing for UWSNs in which several improvements can be applied to achieve better performance. For example, the gateway election algorithm can be enhanced to reduce the overhead and improve power consumption and end-to-end delay. Furthermore, implementing the constructed paths for 3D mesh that are available in the literature [24] and forward data packets through those disjoint paths directly, instead of the four-phases technique used in EMGGR can help in reducing the average end-to-end delay caused by the long paths in the current technique. Moreover, data aggregation technique which has been employed in terrestrial WSNs [31–33] for energy efficiency can be applied to the protocol.

Acknowledgments This work is supported by The Research Council (TRC) of the Sultanate of Oman under the open research grant number RC/SCI/COMP/15/02. This paper is an extension of our conference paper appeared in CyberC-2014 conference [10].

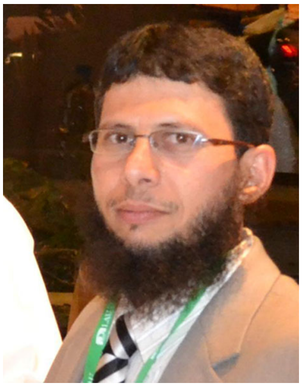
References

1. Akyildiz, I., Pompili, D., & Melodia, T. (2005). Underwater acoustic sensor networks: research challenges. *Ad Hoc Networks, 3*(3), 257–279.
2. Ayaz, M., Baig, I., Abdullah, A., & Faye, I. (2011). A survey on routing techniques in underwater wireless sensor networks.

Journal of Network and Computer Applications, 34(6), 1908–1927.

3. Lanbo, L., Shengli, Z., & Jun-Hong, C. (2008). Prospects and problems of wireless communication for underwater sensor networks. *Wireless Communications and Mobile Computing, 8*(8), 977–994.
4. Xie, P., Cui, J. -H., & Lao, L. (2006). VBF: vector-based forwarding protocol for underwater sensor networks. In *Proceedings of IFIP networking'06*, 15–19 May 2006, Coimbra, pp. 1216–1221.
5. Jornet, J. M., Stojanovic, M., & Zorzi, M. (2008). Focused beam routing protocol for underwater acoustic networks. In *Proceedings of the 3rd ACM international workshop on wireless network testbeds, experimental evaluation and characterization (WuWNeT'08)*, 15–September 2008, San Francisco, CA, pp. 75–82.
6. Chirdchoo, N., Soh, W. -S., & Chua, K. (2009) Sector-based routing with destination location prediction for underwater mobile networks. In *Proceedings of international conference on advanced information networking and applications workshops (WAINA '09)*, 26–29 May 2009, Bradford, pp. 1148–1153.
7. Liao, W.-H., Sheu, J.-P., & Tseng, Y.-C. (2001). GRID: A fully location-aware routing protocol for mobile ad hoc networks. *Telecommunication Systems, 18*(1–3), 37–60.
8. Al-Maqbali, H., Day, K., Ould-Khaoua, M., Touzene, A., & Alzeidi, N. (2014). A grid-based MANET routing protocol with simple cell-head election. *International Journal of Wireless and Mobile Computing, 7*(2), 159–170.
9. Al-Maqbali, H., Day, K., Ould-Khaoua, M., Touzene, A., & Alzeidi, N. (2013). Efficient grid-based MANET routing protocol. *International Journal of Computer Science and Telecommunications, 4*(7), 14–22.
10. Al-Salti, F., Alzeidi, N., & Arafeh, B. (2014). A new multipath grid-based geographic routing protocol for underwater wireless sensor networks. In *Proceedings of cyber-enabled distributed computing and knowledge discovery (CyberC)*, 13–15 October 2014, Shanghai, pp. 331–336.
11. Yao, Y., Cao, Q., & Vasilakos, A. V. (2013) EDAL: An energy-efficient, delay-aware, and lifetime-balancing data collection protocol for wireless sensor networks. In *Proceedings of 2013 IEEE 10th international conference on mobile ad-hoc and sensor systems (MASS)*, 14–16 October 2013, Hangzhou, pp. 182–190.
12. Yao, Y., Cao, Q., & Vasilakos, A. V. (2015). EDAL: An energy-efficient, delay-aware, and lifetime-balancing data collection protocol for heterogeneous wireless sensor networks. *IEEE/ACM Transactions Networking, 23*(3), 810–823.
13. Smith, K. B. (2001). Convergence, stability, and variability of shallow water acoustic predictions using a split-step fourier parabolic equation model. *Journal of Computational Acoustics, 09*(01), 243–285.
14. Llor, J., Stojanovic, M., & Malumbres, M. P. (2011). A simulation analysis of large scale path loss in an underwater acoustic network. In *Proceedings of the 2011 IEEE on OCEANS,* 6–9 June 2011, Santander, pp. 1–5.
15. Xie, P., & Cui, J. -H. (2007). R-MAC: An energy-efficient MAC protocol for underwater sensor networks. In *Proceedings of the international conference on wireless algorithms, systems, and applications (WASA)*, 1–3 August 2007, Chicago, IL, pp. 187–198.
16. Xiao, Y., Peng, M., Gibson, J., Xie, G. G., Du, D.-Z., & Vasilakos, A. V. (2012). Tight performance bounds of multihop fair access for MAC protocols in wireless sensor networks and underwater sensor networks. *IEEE Transactions Mobile Computing, 11*(10), 1538–1554.
17. Felemban, E., Shaikh, F. K., Qureshi, U. M., Sheikh, A., & Bin Qaisar, S. (2015). Underwater sensor network applications: A comprehensive survey. *International Journal of Distributed Sensor Networks*, No. ID 896832, 1–14.
18. Ribeiro, F. J. L., de Castro, A., Pedroza, P., & Costa, L. H. M. K. (2014). Underwater monitoring system for oil exploration using acoustic sensor networks". *Telecommunication Systems, 58*(1), 91–106.
19. Nicolaou, N., See, A., Xie, P., Cui, J. -H., & Maggiorini, D. (2007). Improving the robustness of location-based routing for underwater sensor networks. In *Proceedings of OCEANS 2007—Europe*, 18–21 June 2007, Aberdeen, pp. 1–6.
20. Shin, D., Hwang, D., & Kim, D. (2012). DFR: An efficient directional flooding-based routing protocol in underwater sensor networks. *Wireless Communications and Mobile Computing, 12* (17), 1517–1527.
21. Javaid, N., Jafri, M. R., Khan, Z. A., Alrajeh, N., Imran, M., & Vasilakos, A. (2015). Chain-based communication in cylindrical underwater wireless sensor networks. *Sensors, 15*(2), 3625–3649.
22. Jiang, Z. (2008). Underwater acoustic networks—issues and solutions. *International Journal of Intelligent Control and Systems, 13*(3), 152–161.
23. Day, K., Touzene, A., Arafeh, B., & Alzeidi, N. (2011). Parallel routing in mobile ad-hoc Networks. *International Journal of Computer Networks and Communications, 3*(5), 77–94.
24. Day, K., & Al-Ayyoub, A. E. (1997). Fault diameter of k-ary n-cube networks. *IEEE Transactions on Parallel and Distributed Systems, 8*(9), 903–907.
25. Xie, P., Zhou, Z., Peng, Z., Yan, H., Hu, T., Cui, J. -H., Shi, Z., Fei, Y., & Zhou, S. (2009). Aqua-Sim: An NS-2 based simulator for underwater sensor networks. *Proceedings of MTS/IEEE Biloxi—marine technology for our future: Global and local challenges (OCEANS 2009)*, 26–29 October 2009, Biloxi, MS, pp. 1–7.
26. Davies, V. (2000). *Evaluating mobility models within an ad hoc network*. Master's thesis, Colorado School of Mines, 2000.
27. Camp, T., Boleng, J., & Davies, V. (2002). A survey of mobility models for ad hoc network research. *Wireless Communications and Mobile Computing, 2*(5), 483–502.
28. Masoumeh Alimohammadi, M. A., & Alirezaee, S. (2014). Distance-based underwater routing protocol: DBURP. *International Journal of Computer Science and Network Solutions, 2*(4), 37–47.
29. AL-Zubydi,A. M., & Abdallah, M. N. (2013). Performance assessment of MAC layer protocols in pollution monitoring system based on underwater wireless sensor networks. *Journal of Global Research in Computer Science*, *4*(3), 40–45.
30. Xu, M., Liu, G., & Wu, H. (2014). An energy-efficient routing algorithm for underwater wireless sensor networks inspired by ultrasonic frogs. *International Journal of Distributed Sensor Networks*, Article ID 351520, 12.
31. Xiang, L., Luo, J., & Vasilakos, A. (2011). Compressed data aggregation for energy efficient wireless sensor networks. In *Proceedings of 2011 8th annual IEEE communications society conference on sensor, mesh and ad hoc communications and networks*, 27–30 June 2011, Salt Lake City, UT, pp. 46–54.
32. Liu, X.-Y., Zhu, Y., Kong, L., Liu, C., Gu, Y., Vasilakos, A. V., & Wu, M.-Y. (2015). CDC: Compressive data collection for wireless sensor networks. *IEEE Transactions on Parallel Distributed Systems, 26*(8), 2188–2197.
33. Xu, X., Ansari, R., Khokhar, A., & Vasilakos, A. V. (2015). Hierarchical data aggregation using compressive sensing (HDACS) in WSNs. *ACM Transactions on Sensor Networks (TOSN), 11*(3), 1–25.

Faiza Al Salti received her B. Sc. and M.Sc. Degrees in computer science from the Sultan Qaboos University (Oman) in 2012 and 2015, respectively. She is currently a Ph.D. candidate in the Sultan Qaboos University. Her research interest includes communication protocols, terrestrial and underwater wireless sensor networks.



Nasser Alzeidi received his Ph. D. degree in Computer Science from the University of Glasgow (UK) in 2007. He is currently an assistant professor of computer science and the director of the Center for Information Systems at Sultan Qaboos University, Oman. His research interests include performance evaluation of communication systems, wireless networks, interconnection networks, System on Chip architectures and parallel and distributed computing. He is a member of the IEEE.

Bassel R. Arafeh received the B.Sc. in Electrical Engineering from Cairo University, Egypt, in 1974; and the M.E. degree in Electrical Engineering and the Ph.D. degree in Computer Science from Texas A & M University, College Station, Texas in 1980 and 1986, respectively. Dr. Arafeh is currently an associate professor in the Department of Computer Science, Sultan Qaboos University, Muscat, Oman. His research interests include wireless ad hoc and sensor networks, and parallel and distributed computing systems. He is a member of the ACM, the IEEE and IEEE Computer Society.