DBR: Depth-Based Routing for Underwater Sensor Networks

Hai Yan, Zhijie Jerry Shi, and Jun-Hong Cui

Department of Computer Science and Engineering
University of Connecticut, Storrs, CT 06269-2155
E-mails: {hay04002,zshi,jcui}@engr.uconn.edu

Abstract. Providing scalable and efficient routing services in underwater sensor networks (UWSNs) is very challenging due to the unique characteristics of UWSNs. Firstly, UWSNs often employ acoustic channels for communications because radio signals do not work well in water. Compared with radio-frequency channels, acoustic channels feature much lower bandwidths and several orders of magnitudes longer propagation delays. Secondly, UWSNs usually have very dynamic topology as sensors move passively with water currents. Some routing protocols have been proposed to address the challenging problem in UWSNs. However, most of them assume that the full-dimensional location information of all sensor nodes in a network is known in prior through a localization process, which is yet another challenging issue to be solved in UWSNs. In this paper, we propose a depth-based routing (DBR) protocol. DBR does not require full-dimensional location information of sensor nodes. Instead, it needs only local depth information, which can be easily obtained with an inexpensive depth sensor that can be equipped in every underwater sensor node. A key advantage of our protocol is that it can handle network dynamics efficiently without the assistance of a localization service. Moreover, our routing protocol can take advantage of a multiple-sink underwater sensor network architecture without introducing extra cost. We conduct extensive simulations. The results show that DBR can achieve very high packet delivery ratios (at least 95%) for dense networks with only small communication cost.

1 Introduction

Wireless sensor networks have been used extensively in many land-based applications. Recent several years have also seen a rapidly growing trend towards the application of sensor networks in underwater environments, i.e., building underwater sensor networks (UWSNs) [1] [4] [6] [13] [14] [16]. Among many research issues in this new and promising area, delivering packets from a source node to a destination, namely routing, is one of the fundamental problems that need to be studied for constructing the UWSN protocol stack.

In fact, providing scalable and efficient routing service in UWSNs is very challenging due to the unique characteristics of underwater sensor networks. First of all, radio does not work well in water because of its rapid attenuation. Thus acoustic communications are usually adopted in underwater environments. Acoustic channels often feature low bandwidths and long propagation delays. Thus a routing protocol with long end-to-end delays or high bandwidth requirements is not a good choice. Secondly, most nodes in a UWSN can move passively with water currents (except that some gateway nodes are fixed at the water surface or anchored at the bottom), resulting in highly dynamic network topology. To handle dynamic networks, existing routing protocols for land-based (static) sensor networks need to update routing information periodically, which introduces significant communication overhead. Thirdly, similar to land-based sensor nodes, underwater sensor nodes are usually powered by batteries, which are even harder to recharge or replace in hash underwater environments. Thus, energy efficiency is another important concern for UWSN routing. In short, a routing protocol designed for UWSNs should consider all these factors: long propagation delays, low communication bandwidths, dynamic topology, and energy efficiency.

Recently, there are a couple of proposals for UWSN routing, such as VBF [17] and HHVBF [11]. All of them belong to geographic-based routing protocols and thus require full-dimensional location information of all sensor nodes. This is a tough requirement since localization in UWSNs is another challenging research issue yet to be solved.

In this paper, we propose a novel protocol, called **depth-based routing (DBR)**, for underwater sensor networks. DBR well utilizes the general underwater sensor network architecture: data sinks are usually situated at the water surface. Thus based on the depth information of each sensor, DBR forwards data packets greedily towards the water surface (i.e., the plane of data sinks). In DBR, a data packet has a field that records the depth information of its recent forwarder and is updated at every hop. The basic idea of DBR is as follows. When a node receives a packet, it forwards the packet if its depth is smaller than that embedded in the packet. Otherwise, it discards the packet. Obviously, if there are multiple data sinks deployed at the water surface, as in the multiple-sink underwater sensor architecture [4] [15], DBR can naturally take advantage of them. Packets reach any of the sinks are treated as successfully delivered to the final destination since these water-surface sinks can communicate with each other efficiently through radio channels, which have much higher bandwidths and much lower propagation delays.

To summarize, the main advantages of DBR are as follows. 1) It does not require full-dimensional location information. 2) It can handle dynamic networks with good energy efficiency. 3) It takes advantage of multiple-sink network architecture without introducing extra cost. We will show the performance of DBR using extensive simulations.

The rest of the paper is organized as follows. Section 2 briefly reviews some related work. Section 3 presents the DBR protocol in detail. Section 4 evaluates the performance of DBR through simulations. Finally, Section 5 concludes the paper and discusses some future work.

2 Related Work

In this section, we review several existing routing protocols for wireless sensor networks and underwater sensor networks, which are categorized roughly into two classes.

Routing Protocols for Land-based Sensor Networks Many routing protocols [3] [7] [8] [18] [19] have been proposed for land-based sensor networks. However, most of these protocols are proposed for static or low dynamic networks. For example, in Directed Diffusion [8], routing is initialized by a sink. A sink first floods its interest for data across the network. Upon receiving an interest, a sensor node responds with data along all possible paths. Then an optimal path is enforced from the sensor node to the sink. In SPIN [7], routing is initialized by sensor nodes. When a node wants to send data, it first broadcasts a description of the data. A neighboring node decides whether to request the data based on its local resource. Clearly, a direct application of these routing protocols proposed for static sensor networks in UWSNs with dynamic topologies requires frequent routing information updates, which lead to significant communiation overhead, low energy efficiency, and high end-to-end delays.

Geographic-based Routing Protocols Geographic-based routing protocols [2][5][9] leverage the location information of sensor nodes to forward packets from a source node to a destination node. For example, GPSR [9], is a typical geographic-based routing protocol for land-based sensor networks. The basic idea is that given the location information of all nodes, a sensor node can forward packets to a proper next hop node. A GPSR protocol has two components: a greedy algorithm, and a recovery algorithm. In the greedy algorithm, a forwarding node tries to forward packets to one of its one-hop neighboring nodes which has the shortest distance to the destination node. It is possible that a forwarding node itself has the shortest distance to the destination, resulting in a "void" zone between the forwarder and the destination in which none of the forwarders' neighbors resides. In such cases, a recovery algorithm is performed to route the packet around the "void" zone until the routing can continue in the greedy mode.

VBF [17] is a geographic-based routing protocol designed for underwater sensor networks. In VBF, the packet delivery is guided by the vector from the source to the destination. Only those sensor nodes that are within a range R of the vector will forward packets, where R is a system parameter that can be tuned. The forwarding process of VBF can be seen as to build a routing pipe between the source node and the destination node so that packets are delivered through the nodes in the pipe. To achieve high energy efficiency, multi-path elimination based on a desirableness factor is considered. Namely, an intermediate node only forwards a packet to one of its neighboring nodes, which is the closest to the vector.

VBF works well for underwater sensor networks. However, it makes the assumption that the location information of each sensor node can be obtained through a localization service, which is another difficult issue in UWSNs.

The main difference between our proposal, DBR, and geographic-based routing protocols is that DBR does not need full-dimensional location information. Instead, only local depth information of each node is required in packet forwarding.

3 Depth-Based Routing Protocol

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

3.1 Network Architecture

As mentioned earlier, DBR can naturally take advantage of the multiple-sink underwater sensor network architecture [4] [15]. An example of such networks is illustrated in Fig. 1. In the network, multiple sinks equipped with both radio-frequency (RF) and acoustic modems are deployed at the water surface. Underwater sensor nodes with acoustic modems are distributed in the interested 3-D area, with each likely to be a data source. They can collect data and also help relay data to the sinks. Since all the sinks have RF modems, they can communicate with each other very efficiently via radio channels. Hence, if a data packet arrives at any sink, we assume it can be delivered to other sinks or remote data centers efficiently. This assumption can be easily validated by the fact that sound propagates (at a speed of 1.5×10^3 m/s in water) five orders of magnitudes slower than radio (with a propagation speed of 3×10^8 m/s in air). To be more focused, we do not consider communications between surface sinks in this paper. Instead, we assume that a packet reaches the destination as long as it is successfully delivered to one of the sinks.

Furthermore, we assume that each underwater node knows its depth information, namely the vertical distance from itself to the water surface. In practice, depth information can be obtained easily with a depth sensor. In comparison, obtaining full-dimensional location information is much more difficult.

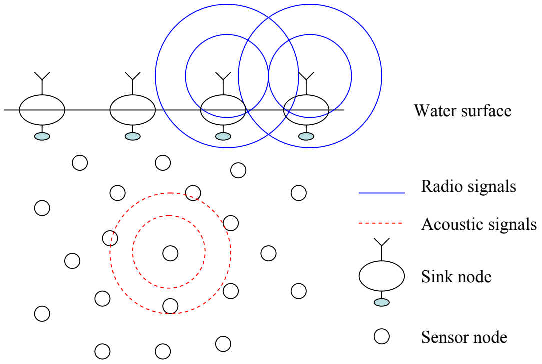


Fig. 1. Multiple-sink underwater sensor network architecture

3.2 Protocol Overview

DBR is a greedy algorithm that tries to deliver a packet from a source node to sinks. During the course, the depth of forwarding nodes decreases while the packet approaches the destination. If we reduce the depth of the forwarding node in each step, a packet can be delivered to the water surface (if no "void" zone is present). In DBR, a sensor node distributively makes its decision on packet forwarding, based on its own depth and the depth of the previous sender. This is the key idea of DBR.

In DBR, upon receiving a packet, a node first retrieves the depth d_p of the packet's previous hop, which is embedded in the packet. The receiving node then compares its own depth d_c with d_p. If the node is closer to the water surface, i.e., $d_c < d_p$, it will consider itself as a qualified candidate to forward the packet. Otherwise, it just simply drops the packet because the packet comes from a (better) node which is closer to the surface. It is not desirable for the receiving node to forward the packet.

It is very likely that multiple neighboring nodes of a forwarding node are qualified candidates to forward a packet at the next hop. If all these qualified nodes try to broadcast the packet, high collision and high energy consumption will result. Therefore, to reduce collision as well as energy consumption, the number of forwarding nodes needs to be controlled. Moreover, due to the inherited multiple-path feature of DBR (in which each sensor node forwards packets in a broadcasting fashion using an omnidirectional acoustic channel), a node may receive the same packet multiple times. Consequently, it may send the packet multiple times. To improve energy efficiency, ideally a node needs to send the same packet only once. We will address the techniques of suppressing redundant packets in the next section.

3.3 Protocol Design

Packet Format The packet format in DBR is illustrated in Fig. 2. The packet header consists of three fields: Sender ID, Packet Sequence Number, and Depth. "Sender ID" is the identifier of the source node. "Packet Sequence Number" is a unique sequence number assigned by the source node to the packet. Together with Sender ID, Packet Sequence Number is used to differentiate packets in later data forwarding. "Depth" is the depth information of the recent forwarder, which is updated hop-by-hop when the packet is forwarded.

Redundant Packet Suppression To save energy as well as reduce collision, redundant packets need to be suppressed. There are two major reasons for redundant packets. One is

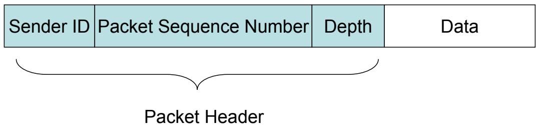


Fig. 2. DBR packet format

that multiple paths are naturally used to forward packets. The other is that a node may send a packet many times. Although multiple paths in DBR can not be completely eliminated, we use a *priority queue* to reduce the number of forwarding nodes, and thus control the number of forwarding paths. To solve the second problem, a *packet history buffer* is used in DBR to ensures that a node forwards the same packet only once in a certain time interval.

In DBR, each node maintains a *priority queue* $Q1$ and a *packet history buffer* $Q2$. An item in $Q2$ is a unique packet ID, which is composed of Sender ID and Packet Sequence Number. When a node successfully sends out a packet, it inserts the unique ID of the packet into $Q2$. When $Q2$ is full, the new item will replace the Least Recently Accessed (LRA) item. In other words, $Q2$ maintains a recent history of the packets the node has sent.

An item in $Q1$ includes two components: a packet and the scheduled sending time for the packet. The priority of an item in $Q1$ is represented by the scheduled sending time. More specifically, an item with earlier sending time has a higher priority. When a node receives a packet, instead of sending the packet immediately, it first holds the packet for a certain amount of time, called *holding time*. The scheduled sending time of a packet is computed based on the time when the packet is received and the holding time for the packet.

At a node, an incoming packet is inserted into $Q1$ if it has not been sent by the node before (i.e., its unique ID is not in $Q2$) and it was sent from a lower node (i.e., a node with a larger depth, $d_p > d_c$). If a packet currently in $Q1$ is received again during the holding time, the packet will be removed from $Q1$ if the new copy is from a node with a smaller or similar depth ($d_p \leq d_c$), or its scheduled sending time will be updated if the new copy is from a lower node ($d_p > d_c$). After a node sends out a packet as scheduled, the packet is removed from $Q1$ and its unique ID inserted into $Q2$.

Holding Time Calculation As mentioned earlier, a node uses *holding time* to schedule packet forwarding. At a node, the holding time for a packet is calculated based on d, the difference between the depth of the packet's previous hop and the depth of the current node. Nodes with different depths will have different holding times even for the same packet. In order to reduce the number of hops along the forwarding paths to the water surface, DBR tries to select the neighboring node with the minimal depth to be the first one to forward a packet. It also tries to prevent other neighboring nodes from forwarding the same packet to reduce energy consumption.

Fig. 3 shows an example. Node S is the sender, and nodes n_1, n_2, and n_3 are all its one-hop neighboring nodes. The solid line circle represents the transmission range of node S. When node S broadcasts a packet, all neighboring nodes will receive this packet. Node n_3 is below S so it discards the packet. Although nodes n_1 and n_2 are both qualified forwarding nodes, node n_1 is preferred to forward the packet. The forwarding of node n_2 is prevented if it receives the packet from n_1 before its own scheduled sending time for the packet.

Based on the above analysis, we can see that the holding time must satisfy two conditions: (1) the holding time should decrease with the increase of d; and (2) the difference between the holding times of two neighboring nodes should be long enough so that the forwarding

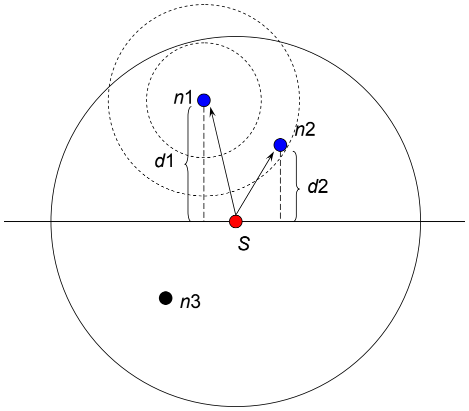


Fig. 3. Forwarding node selection

of the node with the smaller depth can be heard by the other node timely (before the lower node starts its own packet forwarding).

Let us still take Fig. 3 as an example to show how we calculate the holding time. We express the holding time using a linear function of d as follows, where d is the depth difference of the current node and the previous one.

$$f(d) = \alpha \cdot d + \beta \tag{1}$$

Let d_1 and d_2 be the depth difference at nodes n_1 and n_2, respectively. Let us assume that n_1 receives a packet from S at time t_1, n_2 receives the packet at time t_2, and t_{12} is the propagation delay between n_1 and n_2. Then the two conditions can be represented by the following inequalities:

$$f(d_1) < f(d_2), \tag{2}$$

and

$$t_1 + f(d_1) + t_{12} \leq t_2 + f(d_2). \tag{3}$$

Substituting $f(d)$ with our linear expression, we have

$$\alpha \leq \frac{(t_2 - t_1) - t_{12}}{d_1 - d_2}, (\alpha < 0).$$

Here α is negative. As long as $|\alpha| \geq \frac{(t_1-t_2)+t_{12}}{d_1-d_2}$, both conditions can be met. Considering the worst positions for n_1 and n_2, we can choose $|\alpha| = \frac{2\tau}{d_1-d_2}$, where $\tau = R/v_0$ is the maximal propagation delay of one hop (R is the maximal transmission range of a sensor node and v_0 is the sound propagation speed in water).

The value of α depends on $(d_1 - d_2)$, the depth difference of nodes n_1 and n_2. For a node's one-hop neighbors, α can vary between 0 and R, the maximal transmission range of a sensor node. When $(d_1 - d_2)$ approaches 0, $\alpha \to -\infty$. It shows that we cannot find a constant α to make condition (2) always satisfied. Instead, we use a global parameter δ to replace $(d_1 - d_2)$ for the holding time calculation. Therefore $\alpha = -\frac{2\tau}{\delta}$. We guarantee that node n_1 will forward a packet first and prevent the forwarding of node n_2 if $d_1 - d_2 \geq \delta$.

Let the node with the minimal depth have holding time 0. We can compute β by solving the following equation:

$$-\frac{2\tau}{\delta} R + \beta = 0$$

Substituding α and β in the linear function (1), we have the definition of $f(d)$ as follows:

$$f(d) = \frac{2\tau}{\delta} \cdot (R - d), \delta \in (0, R]. \tag{4}$$

When a small δ is chosen, nodes have longer holding times (if their depth differences with the previous forwarder are not exactly R). This may result in longer end-to-end delays. At the same time, the forwarding at these nodes is more likely to be suppressed by the forwarding from a neighboring node closer to the water surface, which results in lower energy consumption. In Section 4, we will study the impact of δ on DBR's performance through simulations.

Depth Threshold In order to further control the number of nodes involved in packet forwarding, we introduce another global parameter, Depth Threshold d_{th}, in DBR. A node will forward a packet only if the difference between the depth of the packet's previous hop d_p and the depth of the current node d_c is larger than the threshold d_{th}. d_{th} can be positive, 0, or negative. If d_{th} is set to 0, all the nodes with smaller depths than the current node are qualified forwarding candidates. If d_{th} is set to $-R$, where R is the maximal transmission range of a sensor node, DBR becomes the flooding protocol.

Clearly, the depth threshold represents the tradeoff between packet delivery ratio and energy consumption. With a larger threshold, less nodes will be involved in packet forwarding, and thus less energy is consumed; but the packet delivery ratio will be lower. On the other hand, with a smaller threshold, more nodes will participate in packet delivery, and thus a higher packet delivery ratio can be achieved; but the energy consumption will be higher.

Summary We summarize the packet forwarding algorithm in Fig. 4.

Upon receiving a packet, a node first checks if it is a qualified forwarder for the packet based on the depth information and the depth threshold d_{th}. If it is not a qualified forwarder, it searches the packet in $Q1$ and removes the packet with the same packet ID since another (better) node has already forwarded the packet. If the node is a qualified forwarder, it searches the packet in the packet history buffer $Q2$. If the packet is found in $Q2$, it is dropped as it has been forwarded recently. Otherwise, the node calculates the sending time for the packet based on the current system time and the holding time and inserts the packet into the priority queue $Q1$. Note that if the packet is already in $Q1$, the sending time is updated to the earlier time. Later, the packets enqueued in $Q1$ will be sent out according to their scheduled sending times.

4 Performance Evaluation

In this section, we evaluate the performance of DBR and compare it with the Vector Based Forwarding (VBF) protocol[17].

4.1 Simulation Settings

All simulations are performed using the Network Simulator (ns2) [12] with an underwater sensor network simulation package (called Aqua-Sim) extension. In our simulations, sensor nodes are randomly deployed in a $500m \times 500m \times 500m$ 3-D area. Multiple sinks are randomly deployed at the water surface. While we assume that the sinks are stationary once deployed, the sensor nodes follow the random-walk mobility pattern. Each sensor node randomly selects

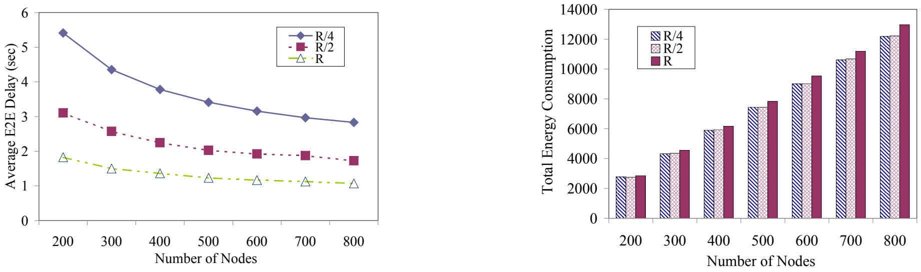
```
Algorithm 1 Algorithm for packet forwarding in DBR
ForwardPacket(p)
Get previous depth $d_p$ from p
Get node's current depth $d_c$
Compute $\Delta d = (d_p - d_c)$
IF $\Delta d$ < Depth Threshold $d_{th}$ THEN
    IF p is in $Q1$ THEN
        Remove p from $Q1$
    ENDIF
    Drop p
    return
ENDIF
IF p is in $Q2$ THEN
    Drop p
    return
ENDIF
Update p with current depth $d_c$
Compute holding time $HT$
Compute sending time $ST$
IF p is in $Q1$ THEN
    Get previous sending time of p $ST_p$
    Update p's sending time with $min(ST, ST_p)$
ELSE
    Add the item <p, $ST$> into $Q1$
ENDIF
```

Fig. 4. DBR packet forwarding algorithm

a direction and moves to the new position with a random speed between the minimal speed and maximal speed, which are 1 m/s and 5 m/s respectively unless specified otherwise. Although the source node can be anywhere in the network, for easy simulations, we place it at a random position at the bottom layer in our experiment. The data generating rate at the source node is one packet per second, with a packet size of 50 bytes. The communication parameters are similar to those on a commercial acoustic modem, LinkQuest UWM1000 [10]: the bit rate is $10k$ bps; the maximal transmission range is 100 meters (in all directions); and the power consumption in sending, receiving, and idling mode are $2w$, $0.1w$, and $10mw$, respectively. The same broadcast Media Access Control (MAC) protocol as in [17] is used in our simulations. In this MAC protocol, when a node has a packet to send, it first senses the channel. If the channel is free, it continues to broadcast the packet. Otherwise, it backs off. The packet will be dropped if the maximal number of backoffs have been reached.

We use the following metrics to evaluate the performance of routing protocols.

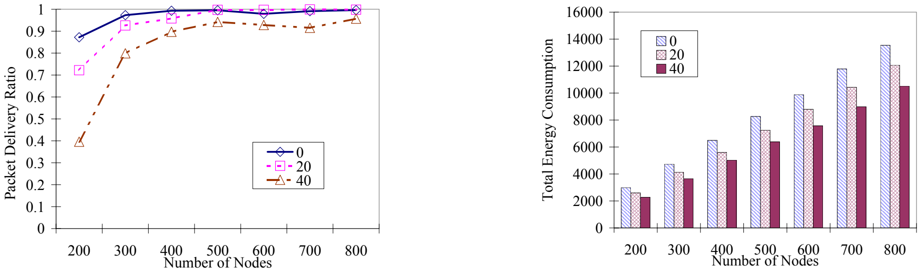
- *Packet Delivery Ratio* is defined as the ratio of the number of distinct packets received successfully at the sinks to the total number of packets generated at the source node. Although a packet may reach the sinks multiple times, these redundant packets are considered as only one distinct packet.
- *Average End-to-end Delay* represents the average time taken by a packet to travel from the source node to any of the sinks.



(a) Average end-to-end delay

(b) Energy consumption

Fig. 5. DBR's performance with different δ values



(a) Packet delivery ratio

(b) Energy consumption

Fig. 6. DBR's performance with different depth thresholds

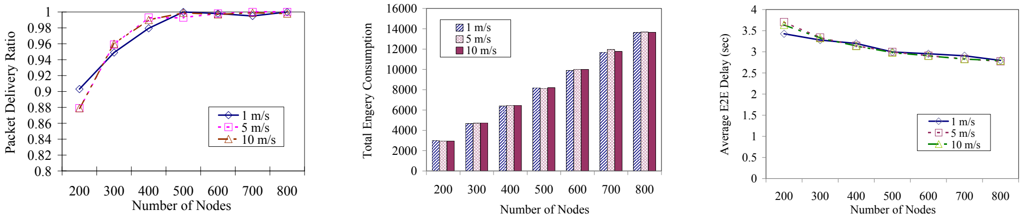
- *Total Energy Consumption* represents the total energy consumed in packet delivery, including transmitting, receiving, and idling energy consumption of all nodes in the network.

4.2 Impact of Design Parameters

Now we examine how the performance of DBR is affected by algorithm parameters and network settings: parameter δ, depth threshold d_{th}, node mobility, and number of sinks.

Parameter δ. The parameter δ decides the holding time of packets at each node. With a larger δ, each node has a shorter holding time and hence the average end-to-end delay will be reduced; but more nodes will forward the same packet, which results in more energy consumption. On the other hand, with a smaller δ, a node will have longer holding time and hence the average end-to-end delay is increased; but less nodes will forward the same packet, as helps to reduce energy consumption.

Fig. 5 shows how the energy consumption and average end-to-end delay change with three different values of δ: $R/4$, $R/2$, and R, where R is the maximal transmission range. In this set of simulations, the number of sinks is set to 5 and the depth threshold is set to 0. From Fig. 5(a), we observe that the average end-to-end delay when $\delta = \frac{R}{4}$ is about three times of the delay when $\delta = R$. Fig. 5(b) shows that the energy consumption does increase as δ is lifted, but not significantly (less than 5%). Hence, we choose $\delta = R$ in later simulations.



(a) Packet delivery ratio

(b) Energy consumption

(c) Average end-to-end delay

Fig. 7. DBR's performance with different node mobility

Depth Threshold. Fig. 6 shows how the depth threshold, d_{th}, affects the packet delivery ratio and energy consumption. In this set of simulations, the number of sinks is set to 5.

From Fig. 6, we see that when the depth threshold increases, both the packet delivery ratio and the total energy consumption decrease. This is because increasing the depth threshold has a similar effect to reducing the number of available nodes in the network. Therefore, the number of forwarding nodes will decrease. Consequently, the packet delivery ratio decreases and less energy is consumed.

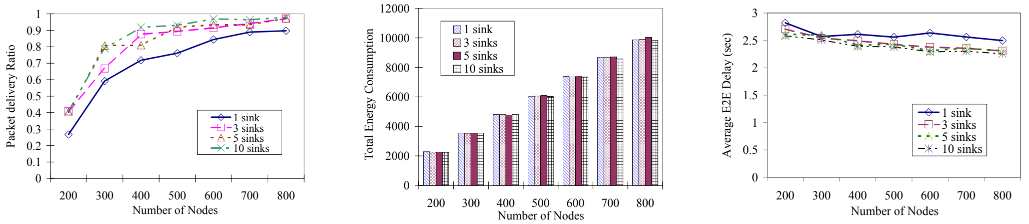
Node Mobility. To evaluate how node mobility affects the performance of DBR, we simulate DBR with different fixed node speeds at 1 m/s, 5 m/s, and 10 m/s. The depth threshold is set to 0 and the total number of sinks deployed is 5. The simulation results are plotted in Fig. 7.

From Fig. 7, we observe that the packet delivery ratio, total energy consumption, and average delay do not change much with node speed. The reason is that all routing decisions in DBR are made locally based on a node's depth information. No topology or route information needs to be exchanged among neighboring nodes. Therefore, DBR can handle dynamic network topologies well. For example, Fig. 7(a) shows that DBR successfully delivers at least 88% of the packets in all cases. When the total number of nodes is larger than 400, almost all packets can be successfully delivered.

From Fig. 7 we can also see how node density affects the performance of DBR. The figure shows that the packet delivery ratio and the energy consumption increase as the number of nodes increases. The average end-to-end delay, however, decreases. This is because when the number of nodes increases, the number of forwarding nodes also increases, which leads to a higher packet delivery ratio and higher energy consumption. At the same time, the probability that a forwarding node has a shorter distance to the sinks increases as more nodes participate in packet forwarding. As a result, the average end-to-end delay decreases.

Number of Sinks. Fig. 8(a) shows how the packet delivery ratio changes with different number of sinks. DBR with multiple sinks has a better packet delivery ratio than DBR with only one sink. Since DBR is a greedy algorithm trying to deliver data packets to the water surface, increasing the number of sinks at the water surface will increase the chance that a packet is received by a sink. This explains the higher delivery ratio when multiple sinks are deployed.

The total energy consumption for different number of sinks is shown in Fig. 8(b). We observe that the energy consumption is almost the same for different number of sinks. The reason is that in DBR, the number of sinks does not affect the forwarding process. Therefore, all different settings have almost the same energy consumption.



(a) Packet delivery ratio

(b) Energy consumption

(c) Average end-to-end delay

Fig. 8. DBR's performance with different number of sinks

From Fig. 8(c), we observe that DBR with multiple sinks has a slightly better average end-to-end delay than DBR with one sink. This is because in the multiple-sink case, a packet is considered successfully delivered whenever it reaches any of the sinks.

4.3 Comparison with VBF

Now we compare DBR with VBF [17]. In this set of simulations, we consider two settings for DBR: one sink and multiple sinks. For one-sink DBR, the depth threshold is set to 0. In multiple-sink DBR, 5 sinks are randomly deployed at the water surface and the depth threshold is set to 20 meters. For VBF, we set the routing pipe radius to 100 meters, which is the maximal transmission range.

Fig. 9 compares DBR and VBF with respect to the three metrics. Fig. 9(a) shows that for the one-sink setting, DBR achieves a similar packet delivery ratio to VBF. With the multi-sink setting, however, DBR can achieve a much better delivery ratio, especially for sparse networks. For example, in a network with 200 nodes, DBR with five sinks has a packet delivery ratio of around 70%, which is more than four times larger than 15%, the delivery ratio of VBF.

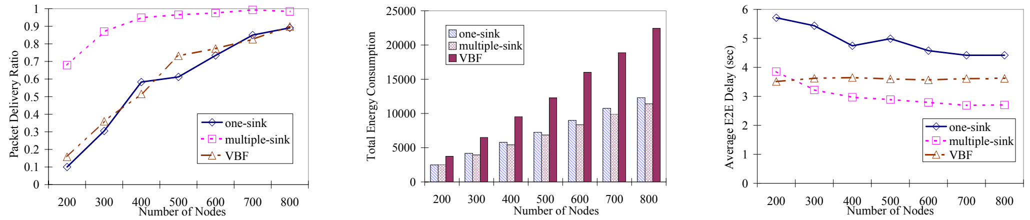
Fig. 9(b) shows that DBR has better energy efficiency compared with VBF. In all cases, the total energy consumption of DBR is about half that of VBF. This is mainly due to the the redundant packet suppression techniques adopted by DBR. The two-queue mechanism in DBR has greatly reduced the redundant packet transmissions in the network. It should be noted, however, that this comparison is made based on the basic settings of VBF. VBF may get better energy efficiency when advanced adaptation algorithms are applied and optimal parameters are chosen.

Fig. 9(c) shows that the best end-to-end delay is achieved in multiple-sink DBR while VBF has a better end-to-end delay than one-sink DBR. This is because VBF tries to find the shortest path from the source node to the sink along the virtual vector between them. Thus the delay in VBF is shorter than that in one-sink DBR. In multiple-sink DBR, however, packets can be delivered to any sink, instead of a fixed sink as in VBF.

It should be noted that DBR and VBF target different network settings and have quite different network assumptions. For example, VBF is designed for networks with a single sink. Although DBR can work in one-sink networks, it has better performance in multiple-sink settings.

4.4 Summary

The simulation results show that DBR protocol works well for dense networks. However, the delivery ratio in sparse networks is relatively low. The reason is that DBR has only a



(a) Packet delivery ratio (b) Energy consumption (c) Average end-to-end delay

Fig. 9. Comparison of DBR and VBF

greedy mode. The greedy method alone is not able to achieve high delivery ratios in sparse networks. We will investigate recovery algorithms for DBR in the future.

The DBR protocol requires more memory in sensor nodes to maintain two buffers. Since the underwater sensor nodes normally are equipped with more resources than land-based sensor nodes, the memory overhead is not significant in most systems. Moreover, the applications for underwater sensor networks have relatively low data rate so only small buffers need to be maintained. For example, we observed in our simulations that the average number of packets in the sending queue of each node is less than 10. As a result, the cost of extra memory is affordable for underwater sensor nodes.

5 Conclusions and Future Work

In this paper, we presented Depth Based Routing (DBR), a routing protocol based on the depth information of nodes, for underwater sensor networks. DBR uses a greedy approach to deliver packets to the sinks at the water surface. Different from other geographical-based routing protocols that require full-dimensional location information of nodes, DBR only needs the depth information, which can be easily obtained locally at each sensor node. Further, DBR can naturally takes advantages of the multiple-sink underwater sensor network architecture without introducing extra cost. Our simulation results have shown that DBR can achieve high packet delivery ratios (at least 95%) for dense networks, with reasonable energy consumption.

To achieve better performance for sparse networks, recovery algorithms need to be explored to avoid the "void" areas where the greedy strategy fails. Advanced recovery algorithms should be developed for higher energy efficiency and delivery ratio.

DBR provides a distributed routing protocol based on local information of the sensor nodes. Besides the depth information, other information, such as the residual energy level and estimated distance to neighboring nodes, could also be useful in making routing decisions that can further reduce energy consumption and extend the network's life time.

For multiple-sink network settings, we only consider some simple cases in which the sinks are randomly, uniformly deployed on the water surface. Given the routing protocol and the node deployment model, we may find better deployment locations for the multiple sinks to achieve better performance.

References

1. I. F. Akyildiz, D. Pompili, and T. Melodia. Challenges for efficient communication in underwater acoustic sensor networks. *ACM SIGBED Review*, 1(1), July 2004.

2. S. Basagni, I. Chlamtac, V. R. Syrotiuk, and B. A. Woodward. A distance routing effect algorithm for mobility (dream). *MOBICOM98*, 1998.
3. D. Braginsky and D. Estrin. Rumor routing algorithm for sensor networks. *WSNA02*, September 2002.
4. J. H. Cui, J. Kong, M. Gerla, and S. Zhou. Challenges: Building scalable mobile underwater wireless sensor networks for aquatic applications. *Special Issue of IEEE Network on Wireless Sensor Networking*, May 2006.
5. M. Grossglauser and M. Vetterli. Locating nodes with ease: Mobility diffusion of last encounters in ad hoc networks. *IEEE INFOCOM03*, March 2003.
6. J. Heidemann, W. Ye, J. Wills, A. Syed, and Y. Li. Research challenges and applications for underwater sensor networking. *IEEE Wireless Communications and Networking Conference*, April 2006.
7. W. R. Heinzelman, J. Kulik, and H. Balakrishnan. Adaptive protocols for information dissemination in wireless sensor networks. *Mobicom99*, August 1999.
8. C. Intanagonwiwat, R. Govindan, and D. Estrin. Directed diffusion: A scalable and roust communication paradigm for sensor networks. *MOBICOM00*, August 2000.
9. B. Karp and H. T. Kung. Gpsr: Greedy perimeter stateless routing for wireless networks. *Proc. Mobicom*, 2000.
10. Linkquest. http://www.link-quest.com/.
11. N. Nicolaou, A. See, P. Xie, J.-H. Cui, and D. Maggiorini. Improving the robustness of location-based routing for underwater sensor networks. *Proceedings of IEEE OCEANS'07*, June 2007.
12. The ns mannual. http://www.isi.edu/nsnam/ns/doc/index.html, 2002.
13. J. Partan, J. Kurose, and B. N. Levine. A survey of practical issues in underwater networks. *Proc. of ACM International Workshop on UnderWater Networks (WUWNet)*, pages 17–24, September 2006.
14. J. Proakis, E.M.Sozer, J. A. Rice, and M.Stojanovic. Shallow water acoustic networks. *IEEE Communications Magazines*, pages 114–119, 2001.
15. W. K. Seach and H. X. Tan. Multipath virtual sink architecture for underwater sensor networks. *OCEANS*, May 2006.
16. G. G. Xie and J. Gibson. A networking protocol for underwater acoustic networks. *Technical Report TR-CS-00-02*, 2000.
17. P. Xie, J.-H. Cui, and L. Lao. Vbf: Vector-based forwarding protocol for underwater sensor networks. *Proceedings of IFIP Networking*, May 2006.
18. F. Ye, H. Luo, J. Cheng, S. Lu, and L. Zhang. A two-tier data dissemination model for large-scale wireless sensor networks. *MOBICOM02*, September 2002.
19. F. Ye, G. Zhong, S. lu, and L. Zhang. Gradient broadcast: A robust data delivery protocol for large scale sensor networks. *ACM Wireless Networks(WINET)*, 11(2), 2005.