

# Adaptive Routing in Underwater Delay/Disruption Tolerant Sensor Networks

Zheng Guo\*, Gioele Colombi<sup>†</sup>, Bing Wang\*, Jun-Hong Cui\*, Dario Maggiorini<sup>†</sup>, Gian Paolo Rossi<sup>†</sup>

\*Computer Science & Engineering Department, University of Connecticut, Storrs, CT, 06269, USA  
Emails: {guozheng, bing, jcui}@engr.uconn.edu

<sup>†</sup>Department of Information Communication, University of Milano, 20135 Milano, Italy  
Emails: gioele.colombi1@studenti.unimi.it, {dario, rossi}@dico.unimi.it

**Abstract**—As an emerging technique, Underwater Sensor Network (UWSN) will enable a wide range of aquatic applications. However, due to the adverse underwater environmental conditions as well as some system constraints, an underwater sensor network is usually viewed as an Intermittently Connected Network (ICN) (or Delay/disruption Tolerant Network (DTN)), which requires specialized routing protocols. Moreover, applications may have different requirements for different types of messages, as demands a smart protocol to handle packets adaptively. In this paper, we propose a novel routing protocol where routing is performed adaptively based on the types of messages and application requirements. This is obtained by exploiting message redundancy and resource reallocation in order to achieve different performance requirements. We demonstrate through simulations that our protocol can satisfy different application requirements and achieve a good trade-off among delivery ratio, end-to-end delay and energy consumption.

## I. INTRODUCTION

The earth is a water planet: over 71% of the surface is covered by the ocean, most of which has not been explored. As a promising solution to aquatic environmental monitoring and exploration, Underwater Sensor Network (UWSN) has attracted significant attention recently from both academia and industry [1], [2], [3], [4], [5]. On the one hand, it can enable a wide range of aquatic applications, such as oceanographic data collection, pollution monitoring, offshore exploration, disaster prevention, assisted navigation and tactical surveillance applications [2]. On the other hand, adverse environmental conditions pose a range of challenges to underwater communication and networking. Firstly, acoustic communication is usually employed in underwater sensor networks, and acoustic channels feature high error rate, long propagation delay and limited bandwidth. Secondly, underwater sensor nodes are usually mobile with water currents. Furthermore, compared to terrestrial sensor nodes, underwater sensor nodes are usually much bigger, more energy-consuming, harder to recharge and more expensive [4]; thus it is very costly to deploy dense underwater sensors in a large area.

Due to node mobility and sparse deployment, an underwater sensor network can be easily partitioned. That is, there may not

be an available persistent route from a source to a destination. Therefore, an underwater sensor network can be viewed as an Intermittently Connected Network (ICN), which is also referred to as Delay/Disruption Tolerant Network (DTN) [6]. For DTN, traditional routing protocols are usually not practical since packets will be dropped when no routes are available.

In addition to intermittent connectivity, an underwater sensor network often needs to provide differentiated packet delivery according to various application requirements. For instance, in water pollution surveillance, a packet that reports pollution should be delivered as quickly as possible, while a packet that reports normal conditions (such as conductivity, temperature, and depth) can tolerate a long end-to-end delay. Thus it is desirable to design a smart routing protocol that could handle these different requirements adaptively.

Motivated by the above considerations, in this paper, we propose an adaptive routing protocol for underwater sensor networks. The goals of our protocol are (1) to satisfy different application requirements; and (2) to achieve a good trade-off among delivery ratio, average end-to-end delay and energy consumption for all packets. Our key idea is to exploit message redundancy and resource reallocation, so that “more important” packets can obtain more resources than other packets so that they could be delivered within shorter delays. We will elaborate our protocol design, and conduct extensive simulations to evaluate its performance. The results will show that our adaptive routing protocol provides differentiated packet delivery according to application requirements and achieves a good performance trade-off among delivery ratio, average end-to-end delay and energy consumption.

The rest of this paper is organized as follows. We discuss related work in Section II. We introduce the problem setting and present our adaptive routing protocol in Sections III and IV respectively. In Section V, we evaluate the performance of our proposed protocol through simulations. Finally, we conclude and point out future directions in Section VI.

## II. RELATED WORK

Several routing protocols have been proposed for underwater sensor networks. Vector Based Forwarding (VBF) protocol [7] aims to solve the problem of high error probability

This work is supported in part by the US National Science Foundation under CAREER Grant No. 0644190 and Grant No. 0709005, and the Italian Ministry of University and Research under Contract No. 2006091851

in dense underwater sensor networks. It defines a routing pipe from the source to the sink and floods inside the pipe. By introducing a self-adaptation approach to suppress redundant transmissions, VBF can significantly improve energy efficiency. In [8], Pompili et al. introduce a two-phase resilient routing solution for long-term monitoring missions based on centralized planning of network topology and data paths. In [9], the same authors propose a routing protocol for delay-insensitive and delay-sensitive applications. In this protocol, cross-layer interaction between routing functions and underwater characteristics is exploited. Different from our work, all of the aforementioned routing protocols are designed for dense and connected underwater sensor networks, which are dramatically different from ICN/DTN, the target network scenarios of this paper.

Delay/Disruption Tolerant Network (DTN) is an emerging type of networks, where routing is very challenging [6]. Traditional forward-or-drop routing protocols are not practical in this type of networks because packets will not be delivered due to the lack of connectivity.

In [10], Zhang surveys a large number of store-and-forward routing protocols proposed for DTN, such as Epidemic, SWIM, Spraying, PROPHET, and MRP. Depending on the amount of message redundancy, we can map these routing protocols to a routing spectrum. One extreme is the Epidemic routing protocol proposed in [11], which is basically a flooding approach. In this routing protocol, each node replicates a packet to every encountered node, thus it can utilize every opportunity to deliver a packet to the destination. Clearly, the Epidemic routing protocol can maximize successful delivery ratio and minimize average end-to-end delay in unconstrained networks (i.e., in networks with plenty of resources such as buffer, available energy and bandwidth). However, this routing protocol consumes too many resources, as is not desirable in resource constrained networks (such as underwater sensor networks). The other extreme is the Single-copy routing protocol designed in [12]. In this protocol, for any packet, only one copy is kept in the network all the time during delivery. Obviously, this routing protocol can help to reduce energy consumption significantly. However, the average end-to-end delay is usually very long. Many other protocols such as [13], [14], [15], [16], [17], and [18] fall in between these two extremes. In these protocols, for any packet, several copies are forwarded to selected nodes encountered based on either flooding [14], [17], [18] or gossiping [13], [15], [16]. Some statistic profiles are utilized to evaluate the benefits earned from sending a copy to current neighbors. Though different protocols use different evaluation schemes, one common idea in these protocols is that multiple copies are exploited to increase the probability that any one copy can reach the destination quickly.

All the aforementioned DTN routing protocols treat every packet equally and aim at a single optimization goal, e.g., minimizing average end-to-end delay, maximizing delivery rate or energy efficiency. None of them is suitable for the applications where packets are of different importance and

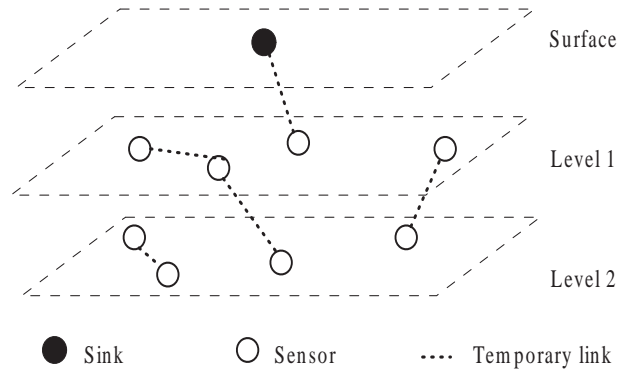


Fig. 1. Illustration of network model.

have various requirements. The novelty of our work is that we propose an adaptive routing protocol by using different number of message copies according to the characteristics of the packets and the network. We will show that our routing solution can satisfy different application requirements and achieve a good trade-off among delivery ratio, average end-to-end delay and energy consumption.

### III. PROBLEM SETTING

In this paper, we consider the application of water pollution surveillance. For this application, underwater sensor nodes are randomly deployed in a target underwater area and periodically report the detected quality of water to the data sink at the water surface. The data sink can be a boat, buoy or other vehicle equipped with radio to communicate with the on-shore control center. According to the quality of water, the data packets generated by sensors can be classified into different emergency levels with corresponding requirements: if the quality of water is good or it only changes in a certain interval, the generated data packets are processed as *ordinary* packets with the lowest emergency level, which only require delivery to the sink with an acceptable delay and as little energy consumption as possible; if some pollution events are detected, then *emergent* data packets are generated with the highest emergency level, which are required to be delivered to the sink as quickly as possible. Otherwise, packets will be generated with an *intermediate* emergency level depending on the quality of the water, which require moderate delay and energy consumption.

Fig. 1 illustrates our target network model: a layered 3D underwater sensor network with sparse node density. Underwater sensor nodes are deployed at different depths using buoyancy control [4], and they can freely move in the horizontal 2D plane, slightly fluctuating in the vertical direction. In this way, sensor nodes at the same depth form a layer. A data sink sits in the center of the water surface. Furthermore, we assume all sensors know their 3D positions through a certain localization service [19], and follow a basic geographic routing scheme, similar to that implemented in [20].

Recall that the goals of our adaptive routing protocol are (1) to provide different services for different types of packets, i.e., provide high energy efficiency to ordinary packets and

low latency to emergent packets; and (2) to achieve a good trade-off among delivery ratio, average delay and energy consumption for all packets. In order to achieve these goals, when a node needs to decide whether to forward a packet or not (we call this situation as “ a packet in transit at a node”), it is not sufficient to only consider the packet emergency level. In our protocol, for each packet in transit at a node, we assign a priority to this packet based on the packet characteristics, such as emergency level and age (i.e., residence time in the network), and the node status, such as residual battery and the density of the neighborhood (see Section IV for detailed definitions). In this way, the calculated priority represents the importance of a packet at a specified node at a certain time. We allow a packet with a higher priority to forward more copies, and routing decisions are made based on the packet priority afterwards.

#### IV. ADAPTIVE ROUTING PROTOCOL

In this section, we present our adaptive routing protocol.

##### A. Packet Types and Node Actions

In our routing protocol, there are two types of packets: *HELLO* packets and *data* packets. *HELLO* packets are used for sensor nodes to exchange information and discover neighborhood, while each *data* packet includes the ordinary payload and a simple protocol overhead with two fields: *emergency level* and *packet generating time* (see Section IV-C for detailed definitions), which are used in packet priority calculation.

Our protocol does not have high requirements on the computational and storage capabilities of sensor nodes. Besides the basic system parameters, such as buffer size, residual battery, and received packet queue, the main additional information each node needs to maintain is its neighborhood status. And the major actions performed on each sensor node are: *neighbor discovery*, *priority calculation* and *routing decision*. In the following, we describe these three actions.

##### B. Neighbor Discovery

For a node in a DTN, one critical issue is to find which nodes are in its transmission range (or called “in contact”) at a certain time. The contacts represent the forwarding opportunities in the DTN. In our adaptive routing protocol, we solve this problem through *neighbor discovery*. More specifically, each node in the sparse underwater sensor network periodically broadcasts *HELLO* packets. Each *HELLO* packet consists of node location, node ID and available buffer size. Node location and ID are used for geographic routing; available buffer size is employed to avoid buffer overflow. Depending on the frequency of *HELLO* packet broadcast, there is a clear trade-off between the routing performance and energy consumption: the more frequent the *HELLO* packets are broadcasted, the more forwarding opportunities can be found, but the more energy will be consumed in neighbor discovery. We will explore this trade-off through simulations in Section V.

**Epidemic ACKs:** In order to suppress redundant packets still existing in the network, we introduce *Epidemic ACKs*. More

specifically, ACKs for successfully received packets at the sink are first broadcasted from the sink node using the Epidemic routing approach proposed in [11]. Any node who receives the ACKs will delete the corresponding packets and further broadcast the received ACKs to the rest of the network. To simplify our protocol design, we let each node (including the sink node) piggy-back ACKs in its periodically broadcasted *HELLO* packets. In this way, a node receiving *HELLO* packets becomes aware of the locations of its neighbors, updates its spatial-temporal density (see Section IV-C) and drops those packets whose ACKs are included. To eliminate duplicate packets more responsively, we let each node maintain all the ACKs it has received. As such, whenever a node receives a packet, it could decide whether to keep or drop the packet.

##### C. Priority Calculation

As mentioned earlier, the priority of a packet should consider the characteristics of the packet and the conditions of the network. More specifically, packet priority is calculated based on an *information vector*. Considering a packet at a node, we define an information vector consisting of *packet emergency level*, *packet age*, *node spatial-temporal density*, and *node battery level*. We describe each element as follows.

- *Packet Emergency Level:* When a packet is generated, its emergency level is *ordinary*, *intermediate* or *emergent*, indicating the detected water quality. This is the main factor to calculate the packet priority.
- *Packet Age:* This element can be retrieved from the packet generating time in the overhead and represents the time the packet spent in the network. To avoid potential long delay for the packet, we will increase the priority when the packet gets order.
- *Node Spatial-Temporal Density:* This element is defined as the number of different neighbors the node encountered in the past certain period of time (called *history window*). It is updated every time the node receives a *HELLO* packet. When a node has low spatial-temporal density, most probably it is in a very sparse area. For this case, we can increase the priorities of the packets at this node to route them quickly out of this sparse area to avoid long delays.
- *Node Battery Level:* This element defines the residual battery energy at the node. Clearly, node battery is one of the most critical resources in underwater sensor networks. To fully utilize the energy, when the node battery level is low, we can decrease the priorities of the packets in transit so that the number of copies to replicate for each packet is reduced. In this way, we can use limit energy resource to deliver more distinct packets rather than more copies of certain packets.

The packet priority value is calculated as follows. We define the range of the priority to be  $[0, 100]$ : 0 means the lowest priority and 100 means the highest priority. Further, we design a mapping function for each element in the information vector so that the mapped values fall into the same range as the priority. More specifically, let  $E, A, D, B$  denote respectively

the mapped values for the packet emergency level, packet age, node spatial-temporal density and node battery level. The priority  $P$  is calculated as a linear combination as follows:

$$P = \alpha_1 E + \alpha_2 A + \alpha_3 D + \alpha_4 B \quad (1)$$

where

$$\alpha_1 + \alpha_2 + \alpha_3 + \alpha_4 = 1, \quad \alpha_1, \alpha_2, \alpha_3, \alpha_4 \in [0, 1]. \quad (2)$$

The weights  $\alpha_i$  ( $0 \leq i \leq 4$ ) are chosen depending on the application requirements. For instance, in water pollution surveillance, the goals are to quickly react to emergent packets while achieve energy-efficiency for ordinary packets. Thus the packet emergency level should be explicitly differentiated and the weight,  $\alpha_1$ , should be large enough. In other scenarios, we may want to assign a large weight to  $\alpha_2$  if the goal is to minimize end-to-end delay.

**When to Calculate Packet Priority:** When a node receives a packet, it does not calculate the priority of the packet immediately. The node only calculate the priority of the packet when it encounters a new neighbor (i.e., a potential forwarding opportunity). This approach is called *per contact routing* [16] and can help us to utilize the latest information to construct the information vector. Once the priority of the packet is calculated, the node can determine whether the new encountered neighbor is a proper “next hop” node (see Section IV-D for decision routing). If yes, the node replicates one copy of the packet to this neighbor, and uses the just calculated priority to decide the left number of copies to be forwarded<sup>1</sup>. After the node forwards the required number of copies, it will discard the packet.

During the delivery of the packet in the network, all elements in the information vector keep changing except the emergency level. Thus, for any packet, it can have different priorities at different nodes.

#### D. Routing Decision

After a node calculates the priority for a packet, it needs to make routing decisions accordingly to forward the packet toward to the sink. In our adaptive routing protocol, we divide the whole routing spectrum into 4 routing states. More specifically, we evenly partition the priority scale into 4 intervals  $[0, 25]$ ,  $(25, 50]$ ,  $(50, 75]$  and  $(75, 100]$  which correspond to the 4 routing states. A node choosing routing state  $i$  for a packet can forward  $i$  copies of the packet ( $1 \leq i \leq 4$ ). The reason we use a maximum 4 copies in our adaptive routing protocol is that we find 4 copies are enough (in our network scenarios) to approach the performance of the Epidemic routing protocol (see Section V).

With the assistance of a localization service, we use geographic routing in our protocol. In Fig. 2, we illustrate the

<sup>1</sup>For simplicity, we use the same priority for all left copies. In reality, however, because the packet age and node properties may change significantly when a node forwards different copies of the same packet, it can recalculate the priority for each copy. Together with previous priorities of this packet, the node can estimate the average number of copies to be forwarded until it satisfies the requirement.

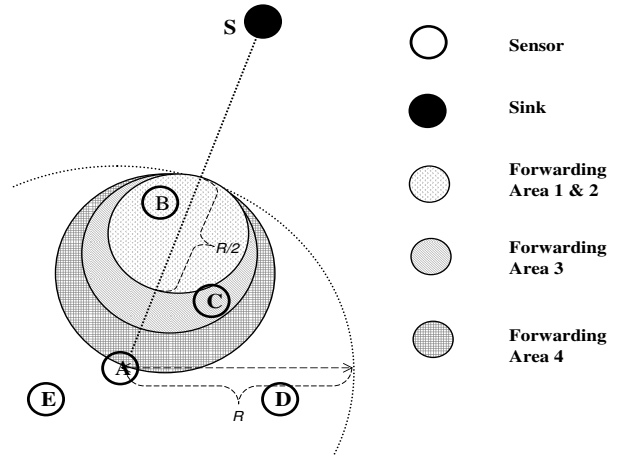


Fig. 2. Illustration of routing decision procedure.

routing decision procedure at a node, specifically showing how the four routing states are integrated with a simple geographic routing scheme. Note that Fig. 2 is a 2-D plot (the front view) of the 3-D network space. In this figure, node  $A$  is the node which needs to make routing decisions based on the routing states of its packets in transit; node  $B$ , node  $C$ , node  $D$ , and node  $E$  are its neighbors; node  $S$  is the sink; and  $R$  denotes the transmission range of sensor nodes. Here, we introduce a new concept, called *forwarding area*. For a packet at routing state  $i$ , it has a corresponding forwarding area  $i$ , which is a sphere tangent to the transmission sphere of node  $A$  at the intersection point of  $\overline{AS}$  (the forwarding vector from the current node to the sink) and the transmission sphere of node  $A$ . The size of each forwarding area is predetermined. In our design, we set the diameter of forwarding area 1, 2, 3, 4 to  $R/2$ ,  $R/2$ ,  $3R/4$ , and  $R$  respectively (due to the sparsity of underwater sensor networks, we choose the same size for forwarding areas 1 & 2). Clearly, in general, a higher priority (i.e., a bigger routing state) corresponds to a larger forwarding area. Now assuming node  $A$  has a packet with a priority of 45. Then this packet has routing state 2. Thus node  $A$  will forward one copy of the packet to a neighbor in the forwarding area 2, that is node  $B$ , and wait until the next neighbor moving into the forwarding area 2 to forward another copy. If there are multiple eligible neighbors at different layers, a node prefers those neighbors at the upper layer since they are closer to the sink, as is in the line of the general greedy approach of geographic routing

## V. PERFORMANCE EVALUATION

In this section, we evaluate the performance of our proposed adaptive routing protocol (we use *Adaptive Routing* for short in this section), compared to the Epidemic routing protocol [11] (*Epidemic Routing* for short), and a Single-copy routing protocol (*Single-copy Routing* for short) through simulations. Single-copy Routing is geographic routing with one copy, and it can be viewed as a special case of Adaptive Routing where

all packets have the lowest priority. We first compare the performances of different protocols in unconstrained networks to demonstrate the benefits of Adaptive Routing. We then evaluate the impact of design parameters. Finally, we choose proper parameters to examine the performance of different protocols in constrained networks.

### A. Simulation Settings

We implement a 3D underwater sensor network with 5 layers. Each layer is an area of  $1km \times 1km$  and contains 16 sensor nodes with a transmission range of  $100m$ . The depth between two consecutive layers is  $60m$ . A single sink is at the center of the surface layer. Each sensor node moves in the plane of its own layer and every second randomly selects a speed between 0 and the maximum speed of  $4m/s$  in both  $x$  and  $y$  directions. When sensor nodes reach the network boundary, they are simply bounced back and continue move as usual. All data packets are generated from those sensors at the lowest layer, with a random emergency level between 1 and 4. We assume that *HELLO* packets are much shorter compared to *data* packets. We define the energy consumed for each *data* packet to be 1 *energy unit* and for each *HELLO* packet to be 0.02 unit. We also implement a simple collision protocol as follows: a node only transmits a packet when there are no other transmissions in its collision domain. We define a node *fails* when it runs out of its battery. Considering the requirements of the pollution surveillance application, we assign a high value of 0.6 to the weight of the emergency level,  $\alpha_1$ , and set the other weights  $\alpha_2$ ,  $\alpha_3$  and  $\alpha_4$  in the priority calculation to be 0.25, 0.05 and 0.1 respectively.

**Performance Metrics:** We use three metrics to compare the performance of different protocols: *delivery ratio*, *average delay* and *normalized energy consumption*. *Delivery ratio* is defined the ratio of the successfully received packets at the sink to the totally generated packets at the sources. *Average delay* denotes the average end-to-end delay for all successfully received packets at the sink. *Normalize energy consumption* is calculated as the average energy consumed for each successfully delivered packet.

### B. Performance Comparison in Unconstrained Networks

We first examine the performance of different protocols in an ideal setting, i.e., an unconstrained network with infinite node battery, buffer size and bandwidth. In this set of experiments, each simulation lasts 4000 seconds. Totally 100 packets are generated with a rate of 1 packet per second, and each packet is generated from a node randomly selected at the lowest layer. Further, we set the interval for *HELLO* packet broadcast 10 seconds. We run 100 simulations to obtain the average.

Table I compares the delivery ratio, average delay and normalized energy consumption for different routing protocols in unconstrained networks. Because there are different types of packets with various emergency levels, we also list the delivery ratio and average delay for packets with the lowest emergency level of 1 and the highest emergency level of 4.

	Epidemic	Adaptive	Single-copy
Delivery ratio	1	1	0.74
Average delay (second)	323	501	1230
Norm. energy consumption (unit)	61.11	34.22	19.77
Delivery ratio (emergency 1)	1	1	0.75
Average delay (emergency 1)	324	596	1239
Delivery ratio (emergency 4)	1	1	0.74
Average delay (emergency 4)	322	395	1222

TABLE I  
PERFORMANCE COMPARISON IN UNCONSTRAINED NETWORKS.

From the table, we can see that without resource constraints, Epidemic Routing can deliver all packets with the lowest average delay, but consumes as much energy as 61.11 units per delivered packet. On the other hand, Single-copy Routing only delivers 74.15% packets with a much higher average delay, although the normalized energy consumption for each delivered packet is just 19.77 units. Compared to Epidemic Routing, Adaptive Routing delivers all packets with slightly longer delays, but consumes about half of the normalized energy for each delivered packet. While compared to Single-copy Routing, it delivers much more packets with substantially shorter delays using slightly more normalized energy.

Now we look at the performance for different types of packets. Table I shows that all packets in either Epidemic Routing or Single-copy Routing have similar performances. This is because these two protocols do not differentiate packets with different emergency levels. Adaptive Routing provides delivery diversity for different packets: Since packets with emergency level of 4 have higher priorities, Adaptive Routing then utilize more resources to deliver them and achieves the average delay of 395 seconds, close to 322 seconds in Epidemic Routing; For packets with emergency level of 1, at the beginning, they have low priorities, but they will be upgraded to high priority when they stay in the network for a relative long time. Thus Adaptive Routing can allocate more resources to avoid long delays and increase the delivery ratio compared to Single-copy Routing.

From the simulation results in unconstrained networks, we conclude that Adaptive Routing could achieve high delivery ratio and high energy efficiency and perform adaptively for packets with different requirements.

### C. Impact of Design Parameters

Clearly, in real underwater sensor networks, node battery, buffer size and bandwidth are always limited. Before evaluating the performance of various protocols in constrained networks, we first investigate the impact of different design parameters in Adaptive Routing. Different from the settings in unconstrained networks, in this set of experiments, each simulation keeps generating 600 packets with a rate of 1 packet every 10 seconds from the nodes at the lowest layer and continues to 8000 seconds. The interval for *HELLO* packet broadcast is still 10 seconds. Further, we assign each node a battery capacity of 300 units and a buffer that can accommodate 50 packets. The bandwidth of each node can

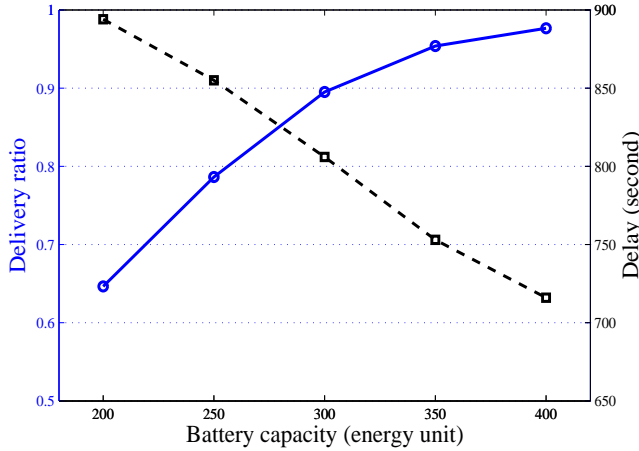


Fig. 3. Impact of battery.

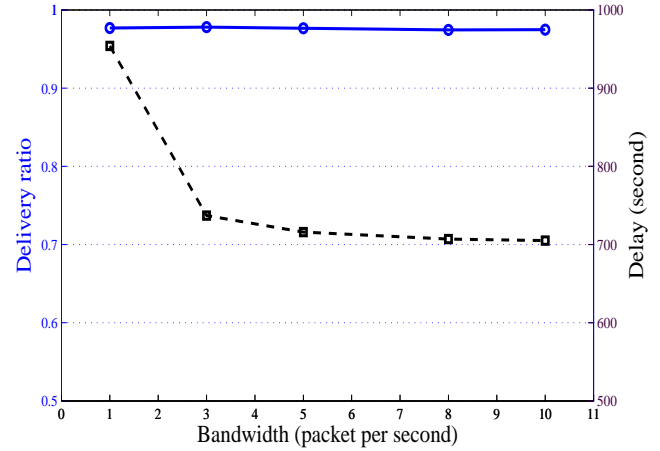


Fig. 5. Impact of bandwidth.

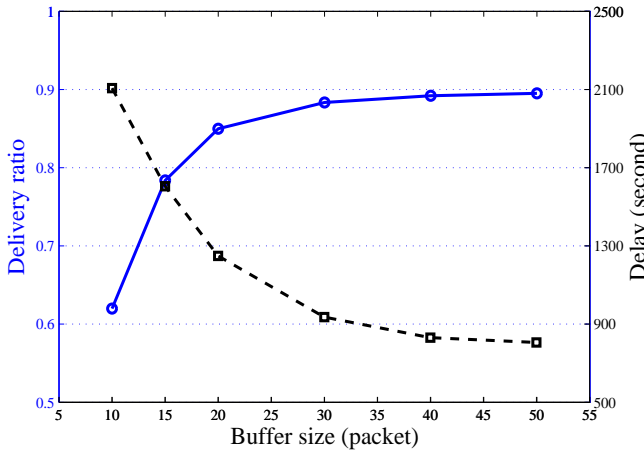


Fig. 4. Impact of buffer size.

support 5 packets per second. In the following, when we investigate one parameter, we only vary this parameter while fixing others.

1) *Impact of battery*: We first vary the battery capability for each node from 200 units to 400 units and show the results in Fig. 3. The solid line denotes the delivery ratio and the dashed line represents the average delay. It is clear that the more battery capacity each node has, the fewer nodes will fail and the more packets can be delivered. When the battery capacity increases, the delivery ratio becomes larger and approaches to 1, and the delay decreases and will approach to the optimal minimum delay, which can be obtained in unconstrained resource settings.

2) *Impact of buffer size*: Now we vary the the number of packets that the buffer at each node can accommodate from 10 to 50 and investigate the impact of buffer size. Fig. 4 shows the results of delivery ratio and average delay (the line presentations here are the same as in Fig. 3). From this figure, we can see that the delivery ratio increases and the average delay decreases as the buffer size is lifted. This is because as

other routing protocols for DTN, Adaptive Routing uses the approach of store-and-forward, which requires large buffers to store packets. If the size is too small, buffers get filled quickly. Nodes with full buffers can not receive more packets from neighbors, as leads a lower delivery ratio and a longer average delay.

Meanwhile, we investigate the impact of buffer size on Epidemic Routing and notice that compared with Adaptive Routing, Epidemic Routing is much more sensitive to buffer size, especially when buffer size is high. As an example, when the buffer size decreases from 50 packets to 30 packets, using Adaptive Routing, the delivery ratio does not change much and the delay only increases from 806 seconds to 936 seconds by 16%. Using Epidemic Routing, although the delivery ratio does not change much either, the delay dramatically increases from 399 seconds to 627 seconds by 57%.

3) *Impact of bandwidth*: In this set of simulations, we assign the battery capacity of each node to be 400 units vary the bandwidth of each node from 1 packet per second to 10 packets per second. The results for delivery ratio and average delay are plotted in Fig. 5 (the line notations here are the same as in Fig. 3). From this figure, we can observe that the performances are very similar with varying bandwidths expect for the extremely low one of 1 packet per second, where the delay is much higher. This is because our target network is sparse and traffic density is also very low. When the bandwidth is as low as 1 packet per second, the channel cannot even afford forwarding all packets stored at a node during one “contact”, as causes longer delays. In short, we can conclude that Adaptive Routing is not very sensitive to bandwidth in sparse and low traffic networks.

4) *Impact of neighbor discovery frequency*: As mentioned in Section IV, the frequency of neighbor discovery (i.e., the HELLO packet broadcast interval) impacts the capability that a node finds possible forwarding opportunities, then influences the average end-to-end delay. In this set of simulation experiments, the HELLO packet broadcast interval is changed from

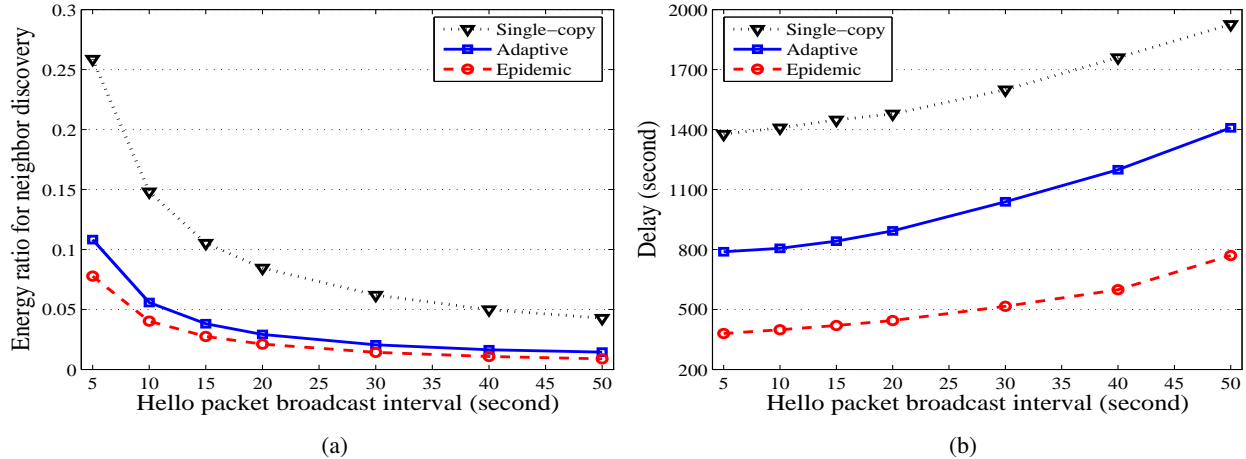


Fig. 6. Impact of *HELLO* packet broadcast interval.

5 seconds to 50 seconds. Fig. 6(a) and Fig. 6(b) show the results on energy ratio for neighbor discovery and average delay respectively. As we can see, when the interval is shorter, the neighbor discovery is more frequently executed, and nodes can find more opportunities to forward a packet. Thus the ratio of energy consumed for neighbor discovery will be higher, showed in Fig. 6(a), and the delay will be shorter, as in Fig. 6(b). Therefore, it is important to choose a proper neighbor discovery frequency to achieve a good trade-off between energy consumption and delay.

#### D. Performance Comparison in Constrained Networks

Now we evaluate the performance of various protocols in constrained networks, where the resources of battery, buffer size and bandwidth are all limited. Based on the impact of different design parameters we just investigated, we choose the battery capacity, buffer size, bandwidth and HELLO packet broadcast interval to be 300 units, 50 packets, 5 packets per second and 10 seconds respectively. Further, in this set of experiments, same as the packet generating pattern in the design parameter exploration section, each simulation keeps generating 600 packets with a rate of 1 packet every 10 seconds and continues to 8000 seconds.

	Epidemic	Adaptive	Single-copy
Delivery ratio	0.57	0.90	0.68
Average delay (second)	399	806	1408
Norm. energy consumption (unit)	48.70	29.87	17.05
Delivery ratio (emergency 1)	0.58	0.80	0.67
Average delay (emergency 1)	401	1157	1407
Delivery ratio (emergency 4)	0.58	0.94	0.68
Average delay (emergency 4)	403	642	1405

TABLE II  
PERFORMANCE COMPARISON IN CONSTRAINED NETWORKS.

1) *Overall performance comparison*: Similar to Table I for unconstrained networks, Table II summarizes the performance of different protocols in constrained networks. For all packets, we notice that the delivery ratio in Adaptive Routing is 33%

higher than that in Epidemic Routing and 22% higher than that in Single-copy Routing. Although the average end-to-end delay using Adaptive Routing is longer than that using Epidemic Routing, the normalized energy consumption is much smaller, which means Adaptive Routing is more energy efficient. Compared to Single-copy Routing, Adaptive Routing achieves a much shorter delay with slightly more energy consumption.

As we know, both Epidemic Routing and Single-copy Routing treat all packets equally. This means that all packets compete for the network resources for delivery. On the other hand, Adaptive Routing could reallocate the precious network resources based on packet priorities. Thus using Adaptive Routing, about 94% of the packets with the emergency level of 4 can be successfully delivered and the average end-to-end delay gets closer to the delay obtained through Epidemic Routing. Even for the packets with the lowest emergency level of 1, the delivery ratio still reaches 0.8 and the average end-to-end delay is much shorter compared to the delay in Single-copy Routing.

From Table II, we can conclude that Adaptive Routing can reallocate the limited network resources to favor those packets which really need them (i.e., packets with higher priorities). Therefore, in constrained networks, Adaptive Routing can provide delivery diversity to satisfy various requirements and achieve a good trade-off among delivery ratio, average end-to-end delay and energy consumption for all packets.

2) *Why better performance on delivery ratio?*: From Table II, we have observed that Adaptive Routing has a much higher delivery ratio than Epidemic Routing. Now let us use two figures to examine why Adaptive routing could achieve a better performance on delivery ratio. Fig. 7(a) and Fig. 7(b) show the cumulative number of received packets versus time and the cumulative number of failed nodes versus time respectively. From Fig. 7(a), we can see that at the beginning, from 0 second to 4000 seconds, Epidemic Routing and Adaptive Routing receive similar number of packets. During the same time interval in Fig. 7(b), we observe see that many nodes

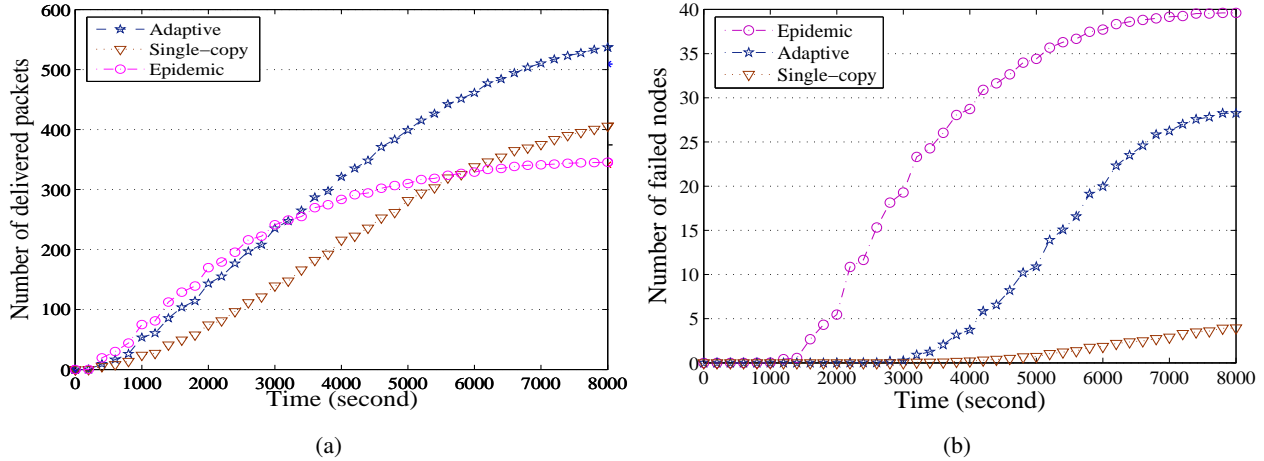


Fig. 7. Packet reception and node failure versus time.

fail in Epidemic Routing, while very few nodes fail in Adaptive Routing. This is because Epidemic Routing aggressively consumes network resources for all packets while Adaptive Routing carefully reallocates the resources to different packets, thus saving energy. After 4000 seconds, Epidemic Routing could only deliver very few packets due to the extensive early node failure, but Adaptive Routing could still deliver packets with a stable speed. Therefore, Adaptive Routing has a better performance on delivery ratio than Epidemic Routing in constrained networks.

3) *A closer look at end-to-end delay*: Since Epidemic Routing greedily delivers every packet, it exhausts the network resources quickly. As we have observed that in Epidemic Routing, most of the delivered packets are at the beginning (see Fig. 7(a)), when the network resources seem to be unconstrained. Thus, in Epidemic Routing, the received packets at the early stage have very short delays, which makes the overall average end-to-end delay to be short. For Adaptive Routing, however, the delivery ratio is much higher and many packets are successfully to the sink at later times. Now we pick a subset of the delivered packets to compare the end-to-end delay of different protocols.

Fig. 8 compares the CDF of delay distributions for different protocols calculated from a subset of received packets (from the 250<sup>th</sup> received packet to the 350<sup>th</sup> received packet). For Epidemic Routing, these packets correspond to all the packets received after 3000 seconds. For Adaptive Routing and Single-copy Routing, these packets correspond to the packets received from 3000 to 4000. Since the curves for packets with different emergency levels overlap for Epidemic Routing and Single-copy Routing, we do not plot them. From Fig. 8, we find that for the selected set of packets, Adaptive Routing achieves a similar average end-to-end delay to Epidemic Routing. Moreover, the packets with emergency level of 4 are delivered faster using Adaptive Routing than using Epidemic Routing. Besides, packets with the lowest emergency level of 1 are also delivered much faster using Adaptive Routing than using Single-copy Routing. This is because these packets become

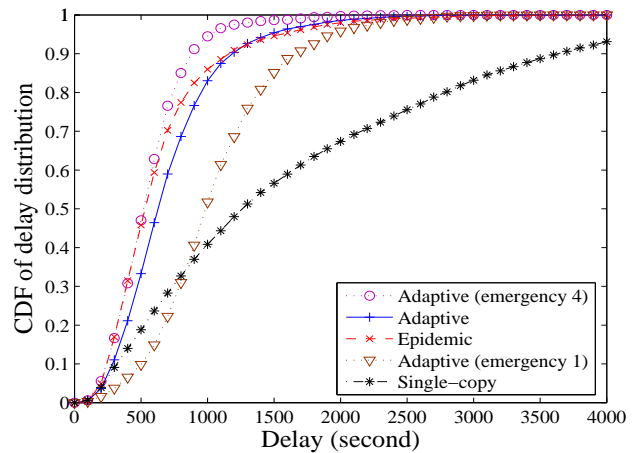


Fig. 8. Comparison of end-to-end delay for packet subset.

more “important” as they age.

## VI. CONCLUSIONS AND FUTURE WORK

In this paper, we have proposed an adaptive routing protocol for underwater delay/disruption tolerant sensor networks. This protocol considers the characteristics of both packets and the network, defines several routing states with different redundancy, and treat different types of packets adaptively by mapping the packet priority to various routing states. Through extensive simulations, we have demonstrated that our protocol can provide delivery diversity to applications with different requirements and achieve a good trade-off among delivery ratio, delay and energy consumption.

As future work, we plan to (1) use water current movement models for more realistic environments and examine the performance of our adaptive routing protocol; (2) dynamically adjust the weights in packet priority calculation according to the network conditions.

## REFERENCES

- [1] E. M. Sozer, M. Stojanovic, and J. G. Proakis, "Underwater Acoustic Networks," *IEEE Journal of Oceanic Engineering*, vol. 25, pp. 72–83, January 2000.
- [2] I. F. Akyildiz, D. Pompili, and T. Melodia, "Underwater Acoustic Sensor Networks: Research Challenges," *Ad Hoc Networks Journal (Elsevier)*, vol. 3, pp. 257–281, March 2005.
- [3] J. Heidemann, W. Ye, J. Wills, A. Syed, and Y. Li, "Research Challenges and Applications for Underwater Sensor Networking," in *Proceedings of IEEE Wireless Communications and Networking Conference*, (Las Vegas, Nevada, USA), pp. 228–235, April 2006.
- [4] J.-H. Cui, J. Kong, M. Gerla, and S. Zhou, "Challenges: Building Scalable Mobile Underwater Wireless Sensor Networks for Aquatic Applications," *IEEE Network, Special Issue on Wireless Sensor Networking*, vol. 20, pp. 12–18, June 2006.
- [5] J. Partan, J. Kurose, and B. N. Levine, "A survey of practical issues in underwater networks," in *Proceedings of ACM WUWNet'06*, (Los Angeles, CA, USA), September 2006.
- [6] S. Jain, K. R. Fall, and R. K. Patra, "Routing in a Delay Tolerant Network," in *Proceedings of ACM SIGCOMM'04*, pp. 145–158, 2004.
- [7] P. Xie, J.-H. Cui, and L. Lao, "VBF: Vector-Based Forwarding Protocol for Underwater Sensor Networks," in *Proceedings of IFIP Networking'06*, (Coimbra, Portugal), pp. 228–235, May 2006.
- [8] D. Pompili, T. Melodia, and I. F. Akyildiz, "A Resilient Routing Algorithm for Long-term Applications in Underwater Sensor Networks," in *Proceedings of Mediterranean Ad Hoc Networking Workshop (Med-Hoc-Net)*, (Lipari, Italy), June 2006.
- [9] D. Pompili, T. Melodia, and I. F. Akyildiz, "Routing Algorithms for Delay-insensitive and Delay-sensitive Applications in Underwater Sensor Networks," in *Proceedings of ACM MobiCom'06*, (Los Angeles, CA), September 2006.
- [10] Z. Zhang, "Routing in Intermittently Connected Mobile Ad Hoc Networks and Delay Tolerant Networks: Overview and Challenges," *IEEE Communications Surveys and Tutorials*, vol. 8, pp. 24–37, 1.
- [11] A. Vahdat and D. Becker, "Epidemic Routing for Partially Connected Ad Hoc Networks," tech. rep., April 2000. TR CS-200006.
- [12] A. Spyropoulos, K. Psounis, and C. Raghavendra, "Single-copy Routing in Intermittently Connected Mobile Networks," in *Proceedings of the First Annual IEEE Communications Society Conference on Sensor and Ad Hoc Communications and Networks*, October 2004.
- [13] T. Small and Z. J. Haas, "The Shared Wireless Infostation Model A New Ad Hoc Networking Paradigm (or Where there is a Whale, there is a Way)," in *Proceedings of MobiHoc'03*, (Annapolis, Maryland, USA), June 1-3 2003.
- [14] D. Nain, N. Petigara, and H. Balakrishnan, "Integrated Routing and Storage for Messaging Applications in Mobile Ad Hoc Networks," in *Proceedings of WiOpt '03: Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks*, (Sophia-Antipolis, France), March 2003.
- [15] A. Lindgren, A. Doria, and O. Schelén, "Probabilistic Routing in Intermittently Connected Networks," in *Proceedings of the First International Workshop on Service Assurance with Partial and Intermittent Resources (SAPIR'04)*, August 2004.
- [16] E. P. Jones, L. Li, and P. A. Ward, "Practical Routing in Delay-Tolerant Networks," in *Proceedings of SIGCOMM'05 Workshop on Delay Tolerant Networking (WDTN'05)*, Philadelphia, USA, September 2005.
- [17] K. Harras, K. Almeroth, and E. Belding-Royer, "Delay Tolerant Mobile Networks (DTMNs): Controlled Flooding Schemes in Sparse Mobile Networks," in *Proceedings of IFIP Networking'05*, 2005.
- [18] T. Spyropoulos, K. Psounis, and C. Raghavendra, "Spray and Focus: Efficient Mobility-Assisted Routing for Heterogeneous and Correlated Mobility," in *Proceedings of IEEE PERCOM Workshop on Intermittently Connected Mobile Ad hoc Networks (ICMAN)*, March 2007.
- [19] Z. Zhou, J.-H. Cui, and A. Bagtzoglou, "Scalable Localization with Mobility Prediction for Underwater Sensor Networks," in *Proceedings of IEEE INFOCOM'08, Mini-Conference*, (Phoenix, Arizona, USA), 2008.
- [20] [http://www.ist-runes.org/news2006\\_06-08.html](http://www.ist-runes.org/news2006_06-08.html)