

# Adaptive Routing in Underwater Delay Tolerant Sensor Networks

Zheng Guo\*, Zheng Peng\*, Bing Wang\*, Jun-Hong Cui\*, Jie Wu†

\*Computer Science and Engineering Department, University of Connecticut, Storrs, CT, 06269

Email: {guozheng, zhengpeng, bing, jcui}@enr.uconn.edu

†Computer and Information Sciences Department, Temple University, Philadelphia, PA 19122

Email: jiewu@temple.edu

**Abstract**— Underwater Sensor Network (UWSN) has attracted significant attention from both academia and industry. Different from terrestrial sensor nodes, underwater sensor nodes are usually mobile, much bigger, more energy-consuming, harder to recharge and suffer from more severe environmental conditions. Thus, an UWSN can be easily partitioned and no persistent routes from a source to a destination are available. Therefore, an UWSN can be viewed as a Delay/Disruption Tolerant Network (DTN). Moreover, an UWSN is always supposed to work for a long time to accomplish multiple tasks with various application requirements, such as delay and delivery ratio. Although many routing protocols have been proposed for DTNs, they are not suitable for UWSNs and cannot handle various application requirements. In this paper, we propose redundancy based adaptive routing (RBAR) for underwater DTNs. Through analysis and simulation, we demonstrate RBAR can achieve the best energy efficiency while satisfying different delay requirements of various packets by explicitly controlling the replication process.

## I. INTRODUCTION

As a promising solution to aquatic environmental monitoring and exploration, Underwater Sensor Network (UWSN) has attracted significant attention recently. Different from terrestrial sensor nodes, underwater sensor nodes are usually mobile, and much bigger, more energy-consuming, harder to recharge and more expensive [1]. Thus, owing to the mobility and sparse deployment, an UWSN can be easily partitioned, with no persistent routes from a source to a destination available. Therefore, an UWSN can be viewed as a Delay/Disruption Tolerant Network (DTN) [2], traditional routing protocols are usually not practical since packets will be dropped when no routes are available.

Many routing protocols have been proposed for DTNs. Most of them utilize multiple copies since obviously more copies imply more opportunities to contact the destinations and quicker delivery, such as Epidemic [3], Spray and Wait [4] and PROPHET [5]. These protocols treat all packets equivalently and try to achieve a tradeoff between energy and delay. However, in real scenarios, an UWSN often needs to provide differentiated packet delivery according to various application requirements. For instance, in water pollution monitoring, 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.

In this paper, we propose redundancy based adaptive routing (RBAR) for underwater DTNs. RBAR allows a node to hold a packet as long as possible until it has to make another copy to satisfy its delay requirement. To achieve this purpose, RBAR adopts a binary tree based forwarding procedure, through which the packet replication process can be explicitly determined. Thus RBAR can achieve the best energy efficiency while satisfying the delay requirements. We also conduct simulations to evaluate the performance of RBAR. The results show that RBAR can adaptively choose the number of copies for packets according to their delay requirements and reach desired delivery ratio for all packets with low energy consumption, while other schemes (such as Epidemic and Spray and Wait) treat all packets equivalently thus cannot achieve the best energy efficiency for all packets.

The rest of paper is organized as follows. In Section II, we introduce the related work. Then we describe the network settings and the detailed redundancy based adaptive routing (RBAR) in Sections III and IV. We compare RBAR with other schemes in Section V and conclude our work in Section VI.

## II. RELATED WORK

Most DTN routing protocols utilize multiple copies to establish multiple potential paths to the destination. According to the principles of how to replicate multiple copies, we categorize the proposed routing protocols as blind redundancy based routing and utility based routing.

### A. Blind Redundancy Based Routing

Blind redundancy based routing is the most intuitive method for DTNs since increasing the amount of copies also increase the probability that any copy can reach the destination quickly. Epidemic [3] is the first representative multi-copy routing scheme by replicating a packet to any node in the network. It is basically a flooding scheme and can optimize the end-to-end delay by taking every potential route. However, Epidemic is too aggressive to consume the network resources. Aiming to avoid unconstrained replication, Harras et. al. propose several controlled flooding schemes in [6], such as time-to-live, kill time and passive cure. Spyropoulos et. al. present spray and wait [4], in which a predetermined certain number of copies of a packet are replicated to the first encountered nodes.

The schemes above all assume that the source determines the amount of copies disseminated to the network. Actually, it is hard to obtain the network conditions and decide the suitable redundancy in DTNs. Xu et. al. raise adaptive spray mechanisms and let the relays make decisions on the spray depth independently, which can be dynamically adjusted in accordance to the network conditions [7].

These schemes simply disseminate multiple copies to the network to take the advantage of multiple delivery opportunities, thus they do not require any information from the network and incur little overhead. However, these blind replication protocols may consume huge network resources and are not practical for harsh networks, such as underwater DTNs.

### B. Utility Based Routing

Utility based routing is applicable to DTNs, where all nodes in the network have different properties at any time. For example, nodes may prefer different locations (communities) or have various behaviors and routines in social networks [8], [9], [10], while some nodes may have higher contact probability to the destination than others [5]. Therefore, it is possible to design a utility metric reflecting these properties, which is a statistic profile to evaluate the benefits earned from sending a copy to current neighbors. Burgess et. al. propose MaxProp to exploit the priorities based on the path likelihood [11]. Lindgren et. al. present PROPHET [5], in which an intermediate node only forwards a packet to the neighbors who have higher probabilities to reach the packet's destination in a short time. In [12], Balasubramanian et. al. present an intentional DTN routing protocol, Rapid, to optimize a specific routing metric. Rapid treats DTN routing as a resource allocation problem that translates the targeting metric to per-packet utility and determines the packets replication in the network. In addition, Jones et. al. utilize the contact history to find routes with minimum estimated expected delay [13]. Wu et. al. propose a scheme that forwards packets to relays with increasing utility to increase reliability [14]. The scheme by Cardei et. al. makes routing decisions based on the probabilistic trajectory prediction [15]. Liu et. al. present optimal probabilistic forwarding (OPF) to maximize the expected delivery rate while satisfying a certain constant on the number of forwardings per packet in [16].

In these schemes, utility is determined by either the special network hierarchy or the underlying mobility model. Utilities are different for different nodes, thus replication is possible along the path of increasing utility. Compared with blind redundancy based routing protocols, utility based routing protocols always have better performance regarding to certain criteria, but they also require more information from the network and incur heavy overhead to maintain and calculate the utility.

All the DTN routing protocols above, both blind redundancy based routing and utility based routing, treat all packets 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 have various requirements. The novelty of our work is that we propose a redundancy based adaptive routing protocol, RBAR, using different number of copies according to the characteristics of the packets and the network.

### III. NETWORK SETTING

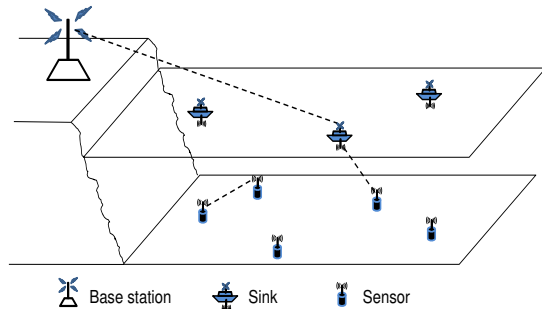


Fig. 1. System model.

We consider a UWSN as shown in Figure 1.  $N$  sensors are deployed in the bottom of the sea while  $M$  buoys float on the surface to collect data generated by the sensors. Both sensors and buoys move following the Random Waypoint model. Thus, for any pair of sensors, the inter-contact time  $t$  follows exponential distribution with intensity  $\lambda$  as  $f(t, \lambda) = \lambda e^{-\lambda t}$  [17]. This means that the process of the contact between this pair is a poisson process and the number of contacts  $k$  within the interval  $t$  follows poisson distribution as  $f(k, \lambda t) = \frac{(\lambda t)^k e^{-\lambda t}}{k!}$ . We assume every node-node pairs have the identical inter-contact pattern. Similarly, we assume the contact process of node-sink pairs is also a poisson process with intensity  $\mu$ . The node buffer is assumed to be infinite and no overflow happens, while the bandwidth is high enough to transfer the whole bundle of packets in the buffer during one contact. Since all nodes are identical (homogeneous network) and poisson process is memoryless, for a pair of nodes, the next expected contact always follows the same exponential distribution at any time. Thus, there will be no obvious better relay node for a packet, as some probability based or utility based DTN routing schemes do like PROPHET. With this assumption, the expected delay only relates to the number of nodes who hold this packet, thus the redundancy based routing schemes like Epidemic, Spray and wait will be better choices.

We consider the application of water pollution surveillance. Sensor nodes generate packets with different emergency priorities to be delivered to the sinks(buoys), and packets with different priorities can tolerant different delay. To demonstrate that our new scheme can satisfy the application requirements, we assume each packet has an explicit delay requirement associated with its emergency. The emergency level  $p$  of a

packet when generated is uniformly randomly selected in the range of  $[0, 100]$  and the delay requirement  $D$  is  $D(p) = a \times p + b$ , where  $a$  and  $b$  are pre-determined. The objective is to guarantee punctual delivery and achieve high energy efficiency at the same time.

#### IV. REDUNDANCY BASED ADAPTIVE ROUTING

In this section, we first introduce the replication procedure, then model it as a continuous Markov chain with absorbing state. With this model, each node can obtain a common replication sequence in a distributed manner. Afterwards, we analytically study the system delay distribution and energy consumption for a packet with any delay requirement.

##### A. Forwarding Procedure In RBAR

We adopt a binary tree based forwarding procedure for the copy replication. Spyropoulos *et. al.* propose a similar scheme, Spray and Wait, to randomly distribute  $n$  copies to the network, where  $n$  is pre-determined [18]. In RBAR, the number of copies required is varying over time and we want to control the global number of copies in the network at any certain time. Thus, each node should estimate the current number of copies and implicitly cooperate to be aware that which node is responsible to replicate the next copy in a distributed way.

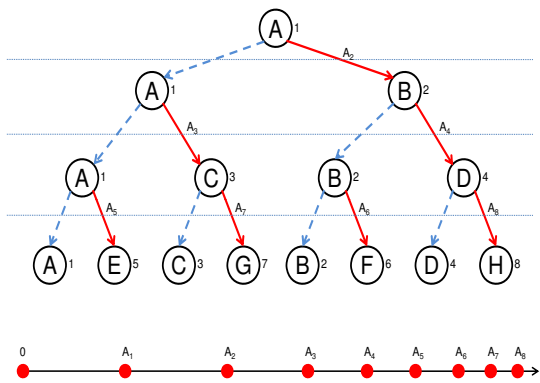


Fig. 2. The forwarding procedure of a packet

The essential part of RBAR is allowing a node to hold a packet as long as possible until very necessary to make another copy. Figure 2 shows the forwarding procedure for a certain packet, which is generated at time  $t_0$  at node  $A$ . Without loss of generality, we assume  $t_0 = 0$ . We first assume every node is aware that there exist a global time sequence  $A_i$  ( $i = 1, 2, 3, \dots, N$ ) (The estimation of the time sequence will be described in Section IV-C), which means until the age of the packet until  $A_i$ , there should be  $i$  copies in the network, otherwise, this packet cannot be delivered to the sinks with high probability. The circles with identifications in Figure 2 represent the nodes which have copies, the numbers besides circles indicate the index of the corresponding copies held

by the node. The dashed line means a node keeps holding a packet, while the solid lines with time stamps imply that when this node should replicate one copy to another node. If a node holds the  $i$ th copy, it is responsible to make the  $l$ th copy until time  $t_l$ , where  $l = 2^{(\lceil \log_2 i \rceil + j)} + i$  ( $j = 0, 1, 2, \dots, \lfloor \log_2 N \rfloor$ ,  $l \leq N$ ). After replication, the receiving node will record its copy index  $l$  and calculate its responsible time stamp to replicate.

Any node holding a packet contacts any one sink can forward this packet to the base station through this sink (buoy). Then all sinks (buoys) immediately broadcast an acknowledgement to the network.

##### B. Markov Model

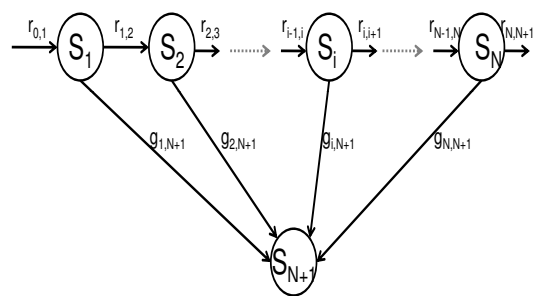


Fig. 3. Markov model with absorbing state

We model the packet replication process as a continuous Markov chain with absorbing state as shown in Figure 3 [19]. We assume all nodes keep replicating copies according to the forwarding procedure as aforementioned in a distributive manner. Thus, we can calculate the necessary time stamps in Figure 2 using this Markov model. In this model, the state  $S_i$  ( $1 \leq i \leq N$ ) represents that there are  $i$  copies in the network, and the state  $S_{N+1}$  represents the absorbing state, which means this packet is delivered to one of the sinks (buoys).  $N$  is the predetermined maximum number of copies allowed in the network, which means at most  $N$  nodes will eventually have a copy. In this setting,  $N$  is the total number of nodes. The state  $S_i$  can only transfer to the state  $S_{i+1}$  with rate  $r_{i,i+1}$  or directly transfer to the state  $S_{N+1}$  with rate  $g_{i,N+1}$ . Ideally, the state  $S_i$  corresponds to the time range  $[A_i, A_{i+1}]$ .

When the network is in the state  $S_i$ , there should be  $i$  copies in  $i$  different nodes. Every node is aware that only one of them is responsible to make a copy to any one node of the left  $N - i$  nodes which do not have this copy. Since any pair of nodes has identical contact process following poisson distribution with intensity  $\lambda$ , thus the transfer rate from  $S_i$  to  $S_{i+1}$  can be obtained as

$$r_{i,i+1} = (N - i)\lambda, \quad 1 \leq i \leq N - 1 \quad (1)$$

Similarly, since there are  $i$  nodes with copies and  $M$  sinks (buoys) and each node-sink pair follows poisson process with intensity  $\mu$ , we can also get the transfer rate from  $S_i$  to  $S_{N+1}$  as

$$g_{i,N+1} = iM\mu, \quad 1 \leq i \leq N \quad (2)$$

The rate  $r_{0,1}$  represents there is a packet ready for delivery, thus we have  $r_{0,1} = 1$ . We also have  $r_{N,N+1} = 0$ .

We use  $F_i^1(t)$  ( $1 \leq i \leq N+1$ ) to denote the probability that the network is in the state  $S_i$  until time  $t$ , starting from the state  $S_1$  when the source is ready to replicate the second copy (the original packet is the first copy), which responds to  $t_1$  in Figure 2. Without loss of generality, we assume  $A_1 = 0$ , thus  $F_1^1(0) = 1$  and  $F_i^1(0) = 0$  ( $2 \leq i \leq N+1$ ). According to Kolmogorov's equation, we can derive the following differential equations about the continuous Markov chain:

$$\begin{cases} \frac{dF_1^1(t)}{dt} &= -(r_{1,2} + g_{1,N+1})F_1^1(t) \\ \dots & \\ \frac{dF_i^1(t)}{dt} &= r_{i-1,i}F_{i-1}^1(t) - (r_{i,i+1} + g_{i,N+1})F_i^1(t) \\ \dots & \\ \frac{dF_{N+1}^1(t)}{dt} &= \sum_{j=1}^N g_{j,N+1}F_j^1(t) \end{cases} \quad (3)$$

Applying Laplace transform on equation 3, we can get

$$\begin{cases} \overline{sF_1^1(s)} - F_1^1(0) &= -(r_{1,2} + g_{1,N+1})\overline{F_1^1(s)} \\ \dots & \\ \overline{sF_i^1(s)} - F_i^1(0) &= r_{i-1,i}\overline{F_{i-1}^1(s)} - (r_{i,i+1} + g_{i,N+1})\overline{F_i^1(s)} \\ \dots & \\ \overline{sF_{N+1}^1(s)} - F_{N+1}^1(0) &= \sum_{j=1}^N g_{j,N+1}\overline{F_j^1(s)} \end{cases} \quad (4)$$

where  $\overline{F_i^1(s)} = \mathcal{Laplace}(F_i^1(t))$ .

Since we have  $F_1^1(0) = 1$  and  $F_i^1(0) = 0$  ( $2 \leq i \leq N+1$ ), we can solve equation 4 to get

$$\begin{cases} \overline{F_1^1(s)} &= \frac{1}{s+r_{1,2}+g_{1,N+1}} \\ \dots & \\ \overline{F_i^1(s)} &= \prod_{j=1}^i \frac{r_{j-1,j}}{s+r_{j,j+1}+g_{j,N+1}} \\ \dots & \\ \overline{F_{N+1}^1(s)} &= \sum_{k=1}^N g_{k,N+1} \frac{1}{s} \prod_{j=1}^k \frac{r_{j-1,j}}{s+r_{j,j+1}+g_{j,N+1}} \end{cases} \quad (5)$$

Through inverse Laplace transform, we get

$$F_{N+1}^1(t) = \sum_{k=1}^N g_{k,N+1} u(t) \otimes (\otimes_{j=1}^k r_{j-1,j} e^{-(r_{j,j+1}+g_{j,N+1})t}) \quad (6)$$

where  $\otimes$  indicates convolution. We should notice that  $F_{N+1}^1(t)$  is the CDF of delay distribution.

Since we want to calculate the necessary replication time stamp when there are already  $i$  copies in the network, we are more interested in the CDF of delay distribution from the initial state  $S_i$ . Following the same process, without loss of generality, we can get the CDF of delay distribution from the initial state  $S_i$ ,  $F_{N+1}^i(t)$ , as

$$F_{N+1}^i(t) = \sum_{k=i}^N g_{k,N+1} u(t) \otimes (\otimes_{j=i}^k r_{j-1,j} e^{-(r_{j,j+1}+g_{j,N+1})t}) \quad (7)$$

where  $r_{i-1,i} = 1$  when starting from the state  $S_i$ .

### C. Generation of Forwarding Time Sequence

After obtaining the CDF of delay distribution from any state  $S_i$ , each node can determine, in a distributed way, the necessary time sequence when the network needs to replicate

one more copy. If a node is supposed to replicate the  $(i+1)$ th copy, then it should calculate the time stamp  $A_i$  in Figure 2, and start the process of replication to guarantee that there are  $i+1$  copies until the age of the packet is  $A_{i+1}$ . Then this node should calculate the necessary time  $T_i$  to deliver the packet to sinks from the state that there are already  $i$  copies (notice that the necessary time  $T_i$  already include the time required to replicate the  $i+1$ th copy).

We define the necessary time  $T_i$  as

$$1 - F_{N+1}^i(T_i) \leq \epsilon \quad (8)$$

where  $\epsilon$  ( $0 \leq \epsilon \leq 1$ ) is a pre-determined tolerant threshold.

Equation 8 means that, if there are  $i$  copies in the network and from the time one node starts to replicate the  $i+1$ th copy, the probability that this packet can be delivered within the following  $T_i$  time is larger than  $1 - \epsilon$ .

For a certain packet with delivery requirement  $D$ , if there are  $i$  copies currently in the network, then all nodes can hold their copies until the age of the packet reaches  $T_i$  when the corresponding node must start replicating the  $(i+1)$ th copy, otherwise the probability that this packet cannot be delivered within the delay requirement  $D$  will exceed the tolerance. We can define  $T_i$  as

$$A_i = D - T_i \quad (9)$$

### D. System Delay Distribution

To obtain the forwarding sequence, we assume that the incoming rate  $r_{i-1}$  into state  $S_i$  is 1 when we calculate the necessary time  $T_i$ . However, during the whole evolving process, the incoming rate into any state is not 1. The forwarding process includes two phases: 1) source holding and 2) copy replication. During the first phase, which lasts for the time  $A_1$ , the source will hold the packet to wait for the sinks, thus the delay distribution is  $f(d) = M\mu e^{-M\mu d}$  ( $0 \leq d \leq A_1$ ). During the second phase, the network replicates copies to other nodes until packet delivery, and the incoming rate  $r_{0,1}$  into state  $S_0$  is the probability that this packet is not delivered during the first phase, which should be

$$r_{0,1} = 1 - \int_{t=0}^{A_1} M\mu e^{-M\mu t} dt = e^{-M\mu A_1} \quad (10)$$

From equation 5 and inverse laplace transform, we get the probability that the network will be in the state  $S_i$  is  $F_j^1(t) = \otimes_{j=1}^k r_{j-1,j} e^{-(r_{j,j+1}+g_{j,N+1})t}$ . Thus, because the delay distribution from the first state  $S_1$  is  $f(d) = \frac{dF_{N+1}^1(t)}{dt}$ , from equation 3, we know that the delay distribution in the second phase is

$$f(d) = \sum_{k=1}^N g_{k,N+1} \otimes_{j=1}^k r_{j-1,j} e^{-(r_{j,j+1}+g_{j,N+1})(d-A_1)} \quad A_1 \leq d \leq D \quad (11)$$

Overall, the system delay distribution through RBAR is

$$f(d) = \begin{cases} M\mu e^{-M\mu d} & 0 \leq d \leq A_1 \\ \sum_{k=1}^N g_{k,N+1} \otimes_{j=1}^k r_{j-1,j} e^{-(r_{j,j+1}+g_{j,N+1})(d-A_1)} & A_1 \leq d \leq D \end{cases} \quad (12)$$

Thus, the CDF of delay distribution is

$$F(d) = \begin{cases} 1 - e^{-M\mu d} & 0 \leq d \leq A_1 \\ F_{N+1}^1(d - A_1) + 1 - e^{-M\mu A_1} & A_1 \leq d \leq D \end{cases} \quad (13)$$

The probability that the packet can be delivered within the delay requirement is that  $F(D) = 1 - r_{0,1}\epsilon = 1 - e^{-M\mu A_1}\epsilon$ .

Notice that this calculation is based on the assumption that the forwarding procedure in the second phase starts from the first state, and enters each state according to the transition rates in the Markov model. However, in the real copy replication phase, nodes holding copies will calculate the replication time and make their forwarding decision independently, which means some replications may happen very closely and not follow the continuous Markov model strictly. Therefore, the system delay distribution in equation 13 can be treated as a lower bound. While from another perspective, we can assume that each node will take its responsibility to replicate one copy exactly at the determined time sequence  $A_i$  ( $1 \leq i \leq N$ ). Thus, the delay distribution in the period  $(A_i, A_{i+1}]$  ( $1 \leq i \leq N - 1$ ) should follow the distribution start from the state  $S_i$  as shown in equation 7. So we can get the upper bound of the delay distribution as following:

$$F(d) = \begin{cases} 1 - e^{-M\mu d} & 0 \leq d \leq A_1 \\ \dots \\ (1 - F(A_i))F_{N+1}^i(d - A_i) + F(A_i) & A_i < d \leq A_{i+1}, 1 \leq i \leq N - 1 \\ \dots \\ (1 - F(A_N))F_{N+1}^N(d - A_N) + F(A_N) & A_i < d \leq D \end{cases} \quad (14)$$

Correspondingly, the PDF of delay distribution is where the rate  $r(i, i + 1)$  is updated as  $1 - F(A_i)$ :

$$f(d) = \begin{cases} M\mu e^{-M\mu d} & 0 \leq d \leq A_1 \\ \dots \\ \sum_{k=i}^N g_{k, N+1} \otimes_{j=i}^k (1 - F(A_i))e^{-(r_{j, j+1} + g_{j, N+1})(d - A_i)} & A_i < d \leq A_{i+1}, 1 \leq i \leq N - 1 \\ \dots \\ g_{N, N+1}(1 - F(A_N))e^{-g_{N, N+1}(d - A_N)} & A_N < d \leq D \end{cases} \quad (15)$$

### E. Average Energy Consumption

Being aware of the forwarding sequence, we can estimate the average energy required to deliver a packet. In the ideal case, there should be  $i$  copies exist during the time period  $(A_i, A_{i+1}]$  ( $1 \leq i \leq N - 1$ ), if one packet is received by the sink during this period, then it involves totally  $i$  transmissions (1 transmission if received in  $(0, A_1]$  and  $N$  transmissions if received in  $(A_N, D]$ ). However, there will be one node to start replicate the  $i$ th copy from time  $A_{i_1}$  ( $2 \leq i \leq N$ ) and this replication time follows the cumulative exponential distribution defined by the rate  $r(i - 1, i) = (N - i + 1)\lambda_{nn}$ . Thus, we can modify the time period for the  $i$ th copy by considering the probability it exists when the packet is delivered. We assume there is always a node to replicate the  $i$ th copy until time  $A_{i-1}$ , thus the probability that the  $i$ th copy exist at time  $t$  is

$$p(i, t) = \begin{cases} 1 & i = 1, 0 < t \leq D \\ 1 - e^{-r(i-1, i)(t - A_{i-1})} & 2 \leq i \leq N, A_{i-1} < t \leq D \end{cases} \quad (16)$$

Thus, the total number of transmissions required for delivering one packet can be obtained as

$$W = \int_{t=0}^D p(1, t)F(t) dt + \sum_{i=2}^N \int_{t=A_{i-1}}^D p(i, t)F(t) dt \quad (17)$$

where  $F(t)$  is the CDF of delay distribution as shown in equation 14.

## V. PERFORMANCE EVALUATION

In this section, We compare RBAR with

- Epidemic: the basic flooding scheme;
- Direct transmission: in which the source waits to directly transmit to the sink without replicating any copies;
- Spray and wait: in which totally maximum  $N$  copies for a packet can be replicated in the network,  $N$  is predetermined.

### A. Simulation Settings

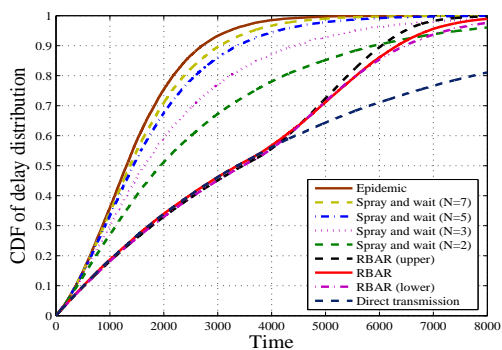
In this simulation, we deploy a two-layer homogeneous UWSN, 10 nodes in the lower layer to generate packets and 2 sinks (buoys) in the upper layer to collect packets. All nodes move according to the random Waypoint model and we assume that all nodes have the identical inter-contact time which follows exponential distribution. The average contact rate between any two sensor nodes is  $\lambda_{nn} = 0.0002$  and the average contact rate between a node and a sink is  $\lambda_{ns} = 0.0001$ . All nodes have unconstrained buffer space and battery. The bandwidth is also unlimited so a node can transmit the whole bundle of packets it has to another node in one contact. Each node will generate 400 packets, which arrive as a poisson process.

### B. Comprehensive Performance Comparisons

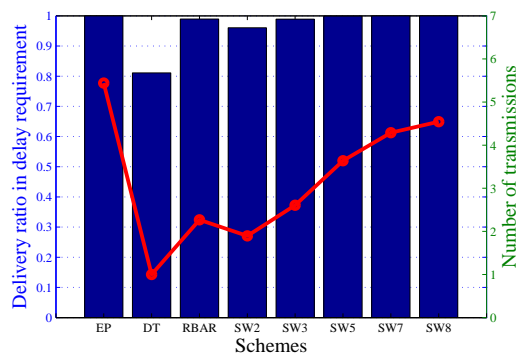
In this chapter, we compare the performance of RBAR with other schemes under different delay requirements.

1) *Under long delay requirement:* When the delay requirement is long, we hope our scheme RBAR can save energy as long as it satisfies the delay requirement. Figure 4 exhibits the performance of different schemes when the delay requirement is 8000s for all packets, in which Figure 4(a) plots the CDF of delay distributions of different schemes, while Figure 4(b) shows the delivery ratio for packets satisfying delay requirement and the corresponding average energy consumption for each scheme (energy consumption is represented using the number of transmissions to deliver a packet).

From Figure 4(a), we can see that Epidemic is the most aggressive scheme. Epidemic can deliver more than 90% of packets with delay less than 2700s and all packets with delay less than 5000s. On the contrary, Direct transmission is the most conservative one, which only delivers 40% of packets with delay less than 2700s and just deliver 80% of all packets within delay requirement finally. As a compromise, Spray and wait predetermines the total number of copies,  $N$ , can be replicated, and their performance falls into the range between Epidemic and Direct transmission. The larger  $N$  is, the closer it approaches to Epidemic. However, all the delay distributions of the schemes discussed above follow a certain trend, while RBAR is dynamically adjusted along the time. As we observe,



(a) CDF of delay distribution.



(b) Delivery ratio in delay requirement and energy consumption.

Fig. 4. Comparison of different schemes under long delay requirement.

RBAR is the same as Direct transmission before the delay around 4000s, this is because RBAR is in the first phase of source forwarding to save energy during this period. After that, the source has to replicate copies to guarantee the packet can be delivered within the delay requirement, resulting the rapidly increased CDF and satisfied final delivery ratio. This means that RBAR is well controlled during the replication procedure and adaptively adjusted according to the delay requirement. Moreover, we also plot the upper and lower bounds for RBAR according to equations 13 and 14 respectively, and demonstrate that the simulation result confirms our analysis.

As an evidence from another perspective, Figure 4(b) plots the delivery ratio in delay requirement and energy required in different schemes. Clearly we can see that Epidemic achieves the highest delivery ratio and consumes the most energy, while Direct transmission is on the opposite side with lowest delivery ratio and lowest energy consumption. Similarly, Spray and wait still falls between Epidemic and Direct transmission, in which the larger number of  $N$  means more replications and more energy consumption with higher delivery ratio. RBAR also shows good performance with satisfied delays and reasonably low energy consumption compared with others.

2) *Under short delay requirement*: To demonstrate that RBAR can be adaptively adjusted, we now compare the performance under short delay requirement of 4000s in Figure 5. Correspondingly, when the delay requirement is short, RBAR should perform aggressively at the beginning to satisfy the delay requirement. In Figure 5(a), Epidemic, Direct transmission and Spray and wait all exhibit very similar trends as their counterparts in Figure 4. However, RBAR is totally different, which replicates packets aggressively approaching Epidemic at the very beginning. The reason behind this phenomenon is that, under very short delay requirement, RBAR can directly enter the second phase of copy replication to speed up the replication procedure.

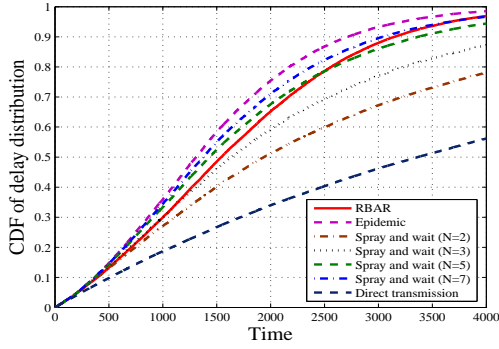
Comparing Figures 4(b) and 5(b), we can clearly observe the benefits of RBAR over other schemes. Epidemic is always too aggressive and Direct transmission is always too conservative. For Spray and wait with  $N = 2$ , although it achieves low energy consumption under long delay requirement, it leads to very low delivery ratio under short delay requirement. While for Spray and wait with  $N = 7$ , it achieves satisfied delivery ratio

under short delay requirement, but also causes huge energy consumption under long delay requirement. Comprehensively, RBAR dynamically adjusts the redundancy and always uses reasonably low energy to satisfy different delay requirements.

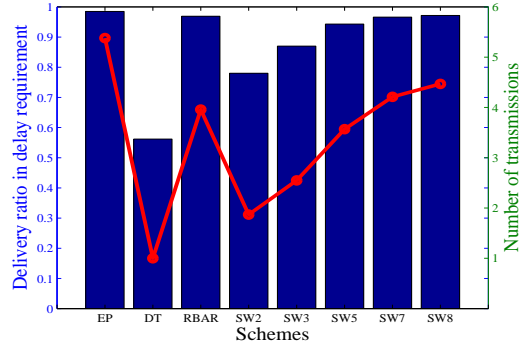
3) *RBAR is adaptive*: As we have discussed, traditional DTN routing schemes do not differentiate packets with various delay requirements, while RBAR adaptively changes the replication strategy to satisfy the requirements. Figures 6(a) and (b) closely examine the difference by plotting the delivery ratio and average number of transmissions for different schemes when the delay requirement increases. For Epidemic, Direct transmission and Spray and wait, they always follow one single replication principle and do not consider the delay requirement, thus the average number of transmissions is almost constant but the percentage of packets delivered with requirements changes significantly. On the contrary, RBAR can always achieve constant delivery ratio with reasonably low energy in accordance to the specific delay requirements.

4) *Under hybrid delay requirement*: As we observe from Figure 6, under a certain delay requirement, some schemes can get similar delivery ratio as RBAR, while some other schemes may achieve similar energy consumption as RBAR. However, none of them can fulfill these performance metric simultaneously. In this simulation, we examine the performance under hybrid delay requirements. We assume a packet is assigned a emergency level  $p$  when it is generated according to the content, which ranges from 0 to 100 with uniform distribution and the delay requirement is determined as  $D(p) = ap + b$ , where  $a = 50$  and  $b = 4000$ . We compare the performance of different schemes in Figure 7. In Figure 7(a), we observe that RBAR is different from others with various trends of CDF of delay distribution at different time, because RBAR adjusts the redundancy and put different efforts at different time. So RBAR achieves the best tradeoff between delivery ratio and energy consumption as shown in Figure 7(b).

5) *When battery is constrained*: Lastly we explore the performance when the precious battery resource is constrained. Since underwater sensor nodes are powered by battery, which is hard, if not impossible, to be replaced. A wise scheme in UWSN should always be energy friendly. Figure 8 compares different schemes by setting the battery for each node to support 1200 transmissions. Obviously, with constrained

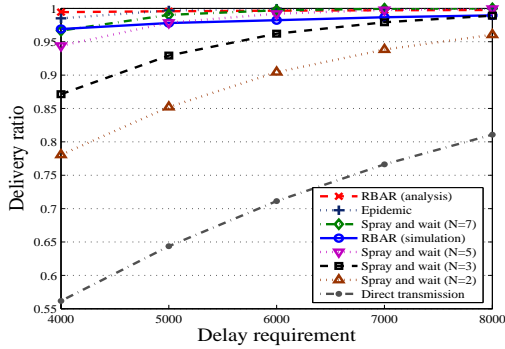


(a) CDF of delay distribution.

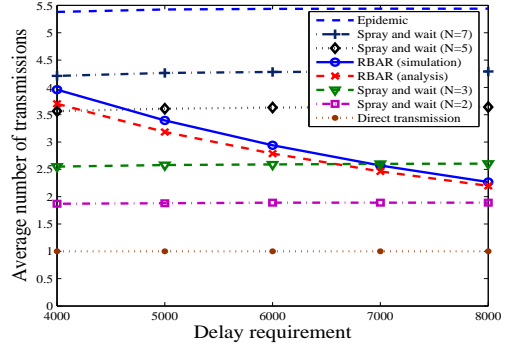


(b) Delivery ratio and energy consumption.

Fig. 5. Comparison of different schemes under short delay requirement.

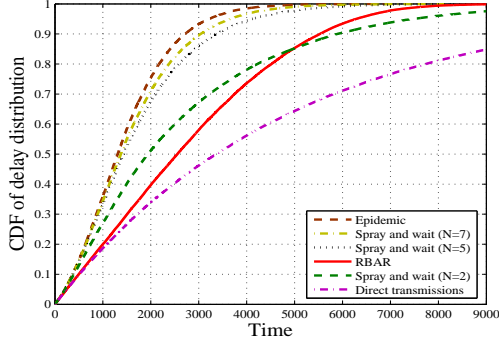


(a) Delivery ratio under different delay requirements.

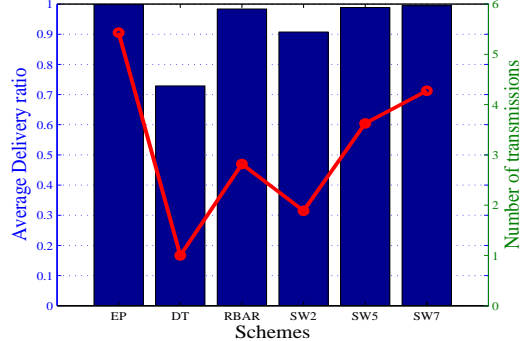


(b) Average number of transmissions under different delay requirements.

Fig. 6. Performance of schemes under different delay requirements.



(a) CDF of delay distribution.



(b) Delivery ratio and energy consumption.

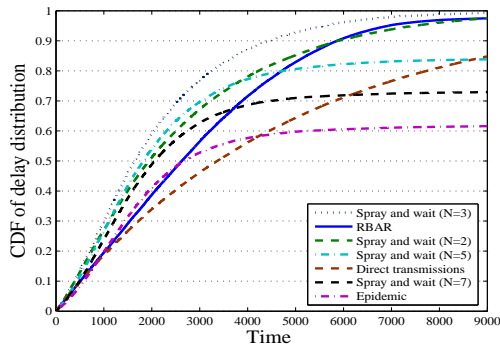
Fig. 7. Comparison of different schemes under hybrid delay requirement.

battery resource, Epidemic can no longer achieve the highest delivery ratio, because it is too aggressive to replicate copies and wastes resource. While RBAR reaches the highest delivery ratio and reasonably low energy consumption, which relies on the adaptive resource allocation based on the different delay requirements. In this special case, Spray and wait with  $N = 3$  performs similarly. However, for other cases, it may not be a good candidate but RBAR can always fit into different situations.

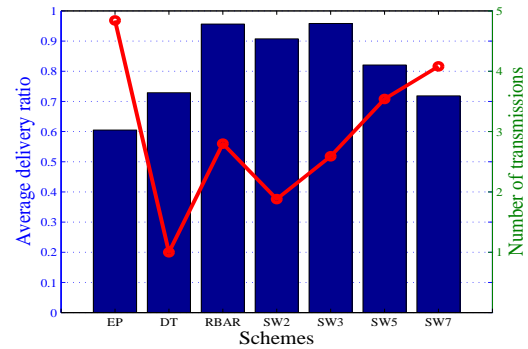
## VI. CONCLUSION

In this paper, we present redundancy based adaptive routing (RBAR) for underwater DTNs. RBAR is dedicated for networks where all nodes and inter-contact time are independent and identically distributed. In these networks, utility based routing schemes are challenging since it is hard to determine which relay is better than others, so RBAR is a good alternative by utilizing redundancy to increase the probability of delivery. Meanwhile, it provides corresponding qualities for various packets with different delay requirements.

RBAR explicitly controls the replication procedure and the



(a) CDF of delay distribution.



(b) Delivery ratio and energy consumption.

Fig. 8. Comparison of different schemes under hybrid delay requirement and battery constraint.

number of copies in the network all the time, so it guarantees the in-time delivery and better resource reallocation than others. Through extensive simulations, we demonstrate that RBAR can provide delivery diversity to applications with different delay requirements and achieve a good trade-off among delivery ratio, delay and energy consumption.

#### REFERENCES

- [1] 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, no. 3, pp. 12–18, 2006.
- [2] S. Jain, K. Fall, and R. Patra, "Routing in a Delay Tolerant Network," in *Proc. of SIGCOMM, Portland, Oregon, USA*, September 2004.
- [3] A. Vahdat and D. Becker, "Epidemic Routing for Partially Connected Ad Hoc Networks," tech. rep. Duke University, April 2000.
- [4] T. Spyropoulos, K. Psounis, and C. S. Raghavendra, "Spray and Wait: an Efficient Routing Scheme for Intermittently Connected Mobile Networks," in *Proc. of SIGCOMM, Philadelphia, Pennsylvania, USA*, 2005.
- [5] A. Lindgren, A. Doria, and O. Schelén, "Probabilistic routing in intermittently connected networks," *SIGMOBILE Mob. Comput. Commun. Rev.*, vol. 7, no. 3, 2003.
- [6] K. A. Harras, K. C. Almeroth, and E. M. Belding-Royer, "Delay Tolerant Mobile Networks (DTMNs): Controlled Flooding in Sparse Mobile Networks," *Lecture Notes in Computer Science*, May 2005.
- [7] J. Xue, X. Fan, Y. Cao, J. Fang, and J. Li, "Spray and Wait Routing Based on Average Delivery Probability in Delay Tolerant Network," *Networks Security, Wireless Communications and Trusted Computing, International Conference on*, vol. 2, 2009.
- [8] W. Gao, Q. Li, B. Zhao, and G. Cao, "Multicasting in Delay Tolerant Networks: a Social Network Perspective," in *Proceedings of the tenth ACM international symposium on Mobile ad hoc networking and computing*, pp. 299–308, ACM, 2009.
- [9] C. Lee, D. Chang, Y. Shim, N. Choi, T. Kwon, and Y. Choi, "Regional Token Based Routing for DTNs," in *Proc. of ICOIN, Chiang Mai, Thailand*, JAN 2009.
- [10] Q. Yuan, I. Cardei, and J. Wu, "Predict and Relay: an Efficient Routing in Disruption-tolerant Networks," in *Proc. of ACM MOBIHOC, New Orleans, LA, USA*, 2009.
- [11] J. Burgess, B. Gallagher, D. Jensen, and B. N. Levine, "MaxProp: Routing for Vehicle-Based Disruption-Tolerant Networks," in *Proceedings of 25th IEEE International Conference on Computer Communications*, 2006.
- [12] A. Balasubramanian, B. N. Levine, and A. Venkataramani, "DTN Routing as a Resource Allocation Problem," tech. rep., University of Massachusetts Amherst, Amherst, MA.
- [13] E. P. Jones, L. Li, and P. A. Ward, "Practical Routing in Delay-tolerant Networks," in *Proc. of WDTN, Philadelphia, PA, USA*, September 2005.
- [14] J. Wu, M. Lu, and F. Li, "Utility-Based Opportunistic Routing in Multi-Hop Wireless Networks," in *Proc. of IEEE ICDCS, Washington, DC, USA*, 2008.
- [15] I. Cardei, C. Liu, J. Wu, and Q. Yuan, "DTN Routing with Probabilistic Trajectory Prediction," in *Proc. of WASA, Dallas, Texas, USA*, 2008.
- [16] C. Liu and J. Wu, "An Optimal Probabilistic Forwarding Protocol Delay Tolerant Networks," in *Proc. of ACM MOBIHOC, New Orleans, LA, USA*, 2009.
- [17] R. Groeneveld, P. Nain, and G. Koole, "Message Delay in MANET," *SIGMETRICS Perform. Eval. Rev.*, vol. 33, no. 1, pp. 412–413, 2005.
- [18] T. Spyropoulos, K. Psounis, and C. S. Raghavendra, "Spray and Wait: an Efficient Routing Scheme for Intermittently Connected Mobile Networks," in *Proceedings of the 2005 ACM SIGCOMM workshop on Delay-tolerant networking*, pp. 252–259, ACM, 2005.
- [19] S. M. Ross, *Introduction to Probability Models*. Elsevier Science & Technology Books, 9th ed., November 2006.