Time-Sensitive Utility-Based Routing in Duty-Cycle Wireless Sensor Networks with Unreliable Links

Mingjun Xiao[†‡§], Jie Wu[‡], and Liusheng Huang[†§]
[†]School of Computer Science and Technology, University of Science and Technology of China, Hefei, P. R. China
[‡]Department of Computer Information and Sciences, Temple University, Philadelphia, PA 19122, USA
[§]Suzhou Institute for Advanced Study, University of Science and Technology of China, Hefei, P. R. China
Email: jiewu@temple.edu, {xiaomj, lshuang}@ustc.edu.cn

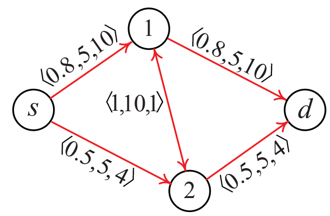
***Abstract*—Utility-based routing is a special routing approach, which takes the reliability and transmission costs into account at the same time. However, the existing utility-based routing algorithms have not yet considered the delivery delay. Thus, they cannot work well in duty-cycle wireless sensor networks (WSNs) since delay is an important factor in such WSNs. In this paper, we propose a novel utility model – *time-sensitive utility* model. Unlike previous work, the utility of a message delivery in our model is not only affected by the reliability and transmission costs but also by the delivery delay. Under the time-sensitive utility model, we derive an iterative formula to compute the time-varying utility of each message delivery. Based on the formula, we propose an optimal time-sensitive utility-based routing algorithm, which is also extended to the case where retransmission is allowed. The theoretical analysis and simulation results show that our proposed algorithms can maximize the average utility of message deliveries, which makes a good tradeoff among reliability, delay, and cost.**

***Index Terms*—Distributed algorithms, duty-cycle wireless sensor networks, reliability, routing, time-sensitive utility.**

I. Introduction

Utility-based routing in wireless networks is a special routing approach based on a composite utility metric [1], [2]. The utility is in terms of the benefit (i.e., a reward for the routing source delivering a message to the destination) minus the expected cost incurred by message delivery. Unlike wired connections, wireless connections are unreliable due to interference and coverage issues. With utility-based routing, the more valuable message will be delivered through a more reliable route at the expense of a higher energy cost in transmission [1], which remains a common phenomenon in wireless communication. This phenomenon reflects a trade-off between a highly reliable route (which is usually more costly) and a less reliable route (which is usually less costly) based on the value of the message. A simple analogy that relates to utility-based routing is the postal service: a high-value package (e.g., one that contains a passport for a visa application) usually uses registered mail for reliability at a higher premium cost. An ordinary package is usually mailed through a regular service.

In this paper, we focus on utility-based routing in duty-cycle wireless sensor networks (WSNs), in which sensors periodically schedule themselves to be active for work and then stay dormant at other times to reduce the energy consumption [3]–[7]. Compared with traditional WSNs, message delivery in duty-cycle WSNs has a non-negligible delay since it has to



benefit	path
50-t	$s\to1\to d$
40-t	$s\to2\to d$
30-0.1t	$s\to2\to1\to d$

Fig. 1. An example of time-sensitive utility-based routing on a weighted graph. The edge weight of the graph is $\langle reliability, delay, cost\rangle$. There are three messages with a linearly decreased benefit over time t. Utility-based routing tries to achieve the maximum utility, i.e., benefit minus cost. As a result, it would let the three messages be delivered along different paths. Their utility values are calculated in Section III-A and are listed in Fig. 5.

wait for a certain amount of time until the message receiver becomes active [6]. The delivery delay is thus an important factor for the routing design. However, it has not been adopted into the current utility-based routing metric.

In order to take the delivery delay into account, we introduce time into the utility model and propose a *T*ime-sensitive *U*tility-based *R*outing (TUR) algorithm for duty-cycle WSNs. The benefit of a message in this algorithm linearly decreases with the delivery time. The utility is still defined as the benefit minus the expected cost. Since the benefit is time-related, the delivery delay is indirectly added into the utility model. As a result, the TUR algorithm makes a trade-off among reliability, delay, and cost. It allows reliability-concerned messages, delay-concerned messages, and cost-concerned messages to be delivered along different paths as shown in the example of Fig. 1. More specifically, our major contributions include:

1) We extend the utility model into duty-cycle WSNs and propose a time-sensitive utility model. Compared with the existing utility model, the time-sensitive utility simultaneously takes reliability, delay, and cost into account. As a result, utility-based routing in this model can make a trade-off among the three factors.
2) We propose an optimal time-sensitive utility-based routing algorithm — TUR. In this algorithm, we first present a special iterative formula to compute the expected utility of a given message delivery. From the iterative formula, we design a backward derivation algorithm to determine the optimal delivery path. The TUR algorithm is a single-copy algorithm without retransmission at each hop. To the best of our knowledge, it is the first utility-based routing algorithm that considers the delivery delay.

3) We also extend the TUR algorithm to cover the cases where retransmission is allowed. We consider two cases: the retransmission occurs within the same duty-cycle or at different duty-cycles. For both cases, we present the optimal solutions.
4) We have conducted extensive simulations to evaluate the TUR algorithm. The results prove that the TUR algorithm can achieve the better expected utility compared to other algorithms. Meanwhile, the results also show that the proposed scheme can make a good balance among reliability, delay, and cost.

The remainder of the paper is organized as follows. We introduce the duty-cycle WSN, the time-sensitive utility model, and the problem of utility-based routing in Section II. The TUR algorithm is proposed in Section III and is extended in Section IV. In Section V, we evaluate the performance of our algorithms through extensive simulations. After reviewing related work in Section VI, we conclude the paper in Section VII. All proofs are presented in the Appendix.

II. Network Model & Problem

A. Network Model

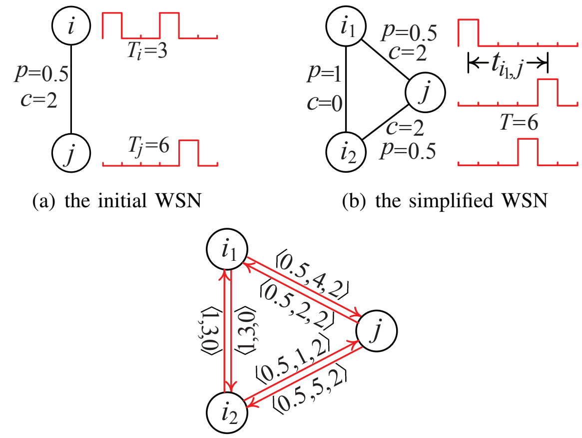
We focus on the static duty-cycle WSN. Each sensor only has two possible working states: *the active state*, in which the sensor can perform all the functions of sensing, listening, transmitting, and receiving; and *the dormant state*, in which the sensor turns off all the functional modules except for a wake-up timer. Specifically, when a dormant sensor wakes up, it either switches to the active state, or transmits packets and then switches back to the dormant state. In other words, a sensor can transmit a packet at any time but can receive a packet only when it is active. Before the concrete network model, we first present several basic assumptions:

1) Time is divided into equal-length time slots, and the whole network is loosely synchronized. The synchronization can be achieved through existing approaches, e.g., FTSP [8]. In general, the time synchronization error can be ignored compared with a time slot [6].

2) Each sensor schedules its working states cyclically. For simplicity, we assume that all sensors share a common duty-cycle and each sensor stays active at only one fixed time slot during each duty-cycle, which is named by the *active time slot* of the sensor. This assumption is reasonable. If sensors have different duty-cycles, the common duty-cycle can be set as their least common multiple. If a sensor has multiple active time slots within a duty-cycle, we can replace this node by several virtual nodes, each of which only has one active time slot in a duty-cycle.

3) The wireless communication links are unreliable, and the CSMA/CA mechanism is adopted to cope with the existence of collision. Previous research shows that the link quality changes very slowly over time [9]. Therefore, the average successful transmission probability derived from history records is adopted to evaluate the link reliability.

Based on the above assumptions, we consider a duty-cycle WSN that is composed of a set of sensor nodes, denoted by



(a) the initial WSN
(b) the simplified WSN
(c) the weighted graph (edge weight: $\langle p, t, c\rangle$)
Fig. 2. Example: duty-cycle WSN modeling.

V. The common duty-cycle is T. For each pair of neighboring sensors, i and j ($i, j \in V$), there is a successful *transmission probability* $p_{i,j}$. Their active time slots are a_i and a_j ($a_i, a_j \in [1, T]$), respectively. Note that node i gets a message only at the time slot a_i. If it wants to send the message to node j, it must sleep until node j becomes active at the time slot a_j. The transmission delay can be ignored since it is much less than the delay incurred by the sleep. Thus, the message *delivery delay* from node i to node j is $t_{i,j} = (a_j - a_i) \mod T$. Besides, the *transmission cost* from node i to node j is denoted by $c_{i,j}$. Then, we can model the duty-cycle WSN as a direct weighted graph $G = \langle V, W\rangle$, where $W = \{\langle p_{i,j}, t_{i,j}, c_{i,j}\rangle | i, j \in V\}$.

Fig. 2 shows an example of duty-cycle WSN modeling. Fig. 2(a) is an initial duty-cycle WSN composed of two sensors i and j, whose duty-cycles are 3 and 6 time slots, and whose active time slots are 1 and 5, respectively. In Fig. 2(b), we utilize two virtual sensors, i_1 and i_2, to replace sensor i. Then, the initial network is simplified to be a duty-cycle network, in which there is only one common duty-cycle, and each node only has one active time slot. After computing the delivery delays of neighboring nodes according to their active time slots, we construct the corresponding direct weighted graph, as shown Fig. 2(c). In fact, any duty-cycle WSN can be converted to a direct weighted graph in this way.

B. Problem

In this paper, we only study *single-copy non-ACK* routing and propose a time-sensitive utility model. Unlike the previous utility model, the time-sensitive utility model assigns a time-sensitive benefit and a utility to a message delivery from an arbitrary source to a destination. This utility metric takes delivery delay, delivery cost, and reliability into account. Considering a message delivery from a source s to a destination d, we present the basic concepts of benefit and utility as follows.

Definition 1: The *benefit* of a message, denoted as $b(t)$, refers to a linearly decreasing reward over time t if it is successfully delivered to its destination; otherwise, zero reward is returned. Let the initial benefit be β, and let the decreased benefit in each time slot be named by *benefit decay coefficient*

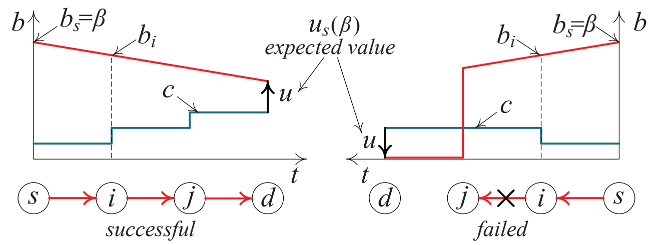


Fig. 3. An example of time-sensitive utility model.

and denoted by δ, then the benefit satisfies:

$$b(t) = \begin{cases} \beta - t \cdot \delta, & \text{successful delivery;} \\ 0 \quad\quad , & \text{failed delivery.} \end{cases} \tag{1}$$

Here, time t is the living time of the message. A new generated message ($t = 0$) has its maximum benefit value. Along with the message delivery, the benefit would linearly decrease due to the elapsed time. If the message delivery fails, the benefit would become zero. Thus, the concept of benefit takes into account both the delivery delay and the successful delivery probability.

Definition 2: The *utility* of a message delivery, denoted by u, is the benefit minus the total transmission cost of the message delivery, which means the gain of the message delivery. Let the total transmission cost be c, then the utility satisfies:

$$u = b(t) - c. \tag{2}$$

If the message is successfully delivered to the destination with the delay $t_{s,d}$, the utility would be $b(t_{s,d}) - c$; otherwise if it fails, the utility would be $0 - c$. The utility value is affected by the delivery delay, the path reliability, and the transmission cost. For example, for the delivery from s to 1 of the first message in Fig. 1, the benefit is $50 - t_{s,1} = 45$ and the utility is $45 - 10$ for the successful delivery. The benefit is 0 and the utility is $0 - 10$ for the failed delivery. The expected value of utilities for the two delivery cases is $0.8(45–10)+0.2(0–10) = 26$.

The above concepts b, u, and c are related to the message delivery from s to d. In addition, for simplicity of description, we also define two notions for each node: the remaining benefit of a node and the expected utility of a node. Consider an arbitrary node i in the delivery path from s to d, the remaining benefit and expected utility of node i are defined as follows.

Definition 3: The *remaining benefit* of node i, denoted by b_i, refers to the remaining benefit value when the message arrives at node i. That is:

$$b_i = \beta - \delta \cdot t_{s,i}. \tag{3}$$

Definition 4: The *expected utility* of node i, denoted by $u_i(b)$, is the expected utility for a message delivery from node i to the destination, in which the remaining benefit of the message is b when it arrives at (or is generated by) node i.

Note that b_i and $u_i(b)$ are the values from the point of view of node i, i.e., the case when node i is the current message forwarder. Moreover, $u_i(b)$ is an expected value. This is because the message delivery from node i to the destination is uncertain. It might succeed or fail at different hops. There are multiple possible results. For each result, there is a probability and a final utility value. $u_i(b)$ is the expected value of these final utility values. Note that $u_i(b)$ is a function of b. This means that $u_i(b)$ can be derived only when b is given in advance. Fig. 3 shows an example for the concepts, in which the benefit linearly decreases along time or becomes zero due to a failed delivery. b_i is the remaining benefit of node i. c is the transmission cost. There is a utility u for each message delivery no matter if it succeeds or fails. The expected utility of source node $u_s(\beta)$ is the expected value of u.

With the basic definitions of benefit and utility, we can present our problem of utility-based routing as follows: given a duty-cycle network $G = \langle V, W \rangle$, as described in Section II-A, a source node s, a destination node d, an initial benefit β, and a benefit decay coefficient δ, then our objective is to maximize the expected value of utility u for the message delivery from s to d. Since this expected value is exactly equal to the expected utility $u_s(\beta)$, our objective becomes to maximize $u_s(\beta)$.

III. Solution: The TUR Algorithm

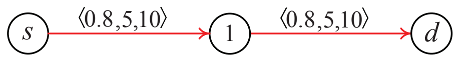
In this section, we consider the utility-based routing problem for a non-retransmission message delivery from an arbitrary source node s to a destination node d with an initial benefit β and a benefit decay coefficient δ. We propose an optimal *T*ime-sensitive *U*tility-based *R*outing (TUR) algorithm, where the maximum expected utility $u_s(\beta)$ can be achieved. The key part of the TUR algorithm is to find an optimal delivery path in the initial phase. We first present an iterative formula, by which each node can compute its own optimal expected utility value when it knows the optimal expected utility values of the neighboring nodes. Then, we design a backward derivation algorithm to calculate the optimal expected utility value of each node. Accordingly, the optimal delivery path is also determined. The routing phase of the TUR algorithm just let messages be delivered along their optimal paths. Since the routing phase is straightforward, we only focus on the process of computing the expected utility values of nodes and finding the optimal delivery path in the following parts.

A. The Basic Formula

We first consider an arbitrary delivery path from node s to node d and derive a formula to compute the expected utility value. Without the loss of generality, we let the path be "$s = 0 \to 1 \to \cdots \to n-1 \to d = n$". Then, the expected utility of the message delivery from s to d is $u_s(\beta) = u_0(\beta)$. Assume that all edge weights in the path, including the successful transmission probability, the delivery delay, and the transmission cost, are known. By computing the probability and utility values for each possible delivery case, we can the get the formula. More specifically, we have the following theorem.

Theorem 1: The expected utility value for the message delivery with an initial benefit β and a benefit decay coefficient δ along a given path "$s = 0 \to 1 \to \cdots \to n-1 \to d = n$" satisfies:

$$u_s(\beta) = \prod_{i=0}^{n-1} p_{i,i+1}\left(\beta - \delta \sum_{i=0}^{n-1} t_{i,i+1}\right) - \sum_{i=0}^{n-1} c_{i,i+1} \prod_{j=0}^{i-1} p_{j,j+1}. \tag{4}$$



benefit	β=50, δ=1
directly computation	u_s=0.8×0.8×(50−1×(5+5))−(10+10×0.8)=7.6
iteratively computation	b_s=50, b_1=45, b_d=40
	$u_d(b_d)$= b_d=40
	$u_1(b_1)$=$p_{1,d}$×$u_d(b_d)$−$c_{1,d}$=0.8×40−10=22
	$u_s(b_s)$=$p_{s,1}$×$u_1(b_1)$−$c_{s,1}$=0.8×22−10=7.6

Fig. 4. An example of the expected utility computation. The edge weight of the graph is $\langle reliability, delay, cost\rangle$. The direct computation and the iterative computation achieve the same result.

Now we derive an iterative formula which can be used to locally compute the expected utility value. Consider two arbitrary adjacent nodes i and $j = i+1$ $(0 \leq i \leq n-1)$ in the delivery path "$s = 0 \rightarrow 1 \rightarrow \cdots \rightarrow n-1 \rightarrow d = n$". Note that their expected utilities $u_i(b)$ and $u_j(b)$ actually are two functions about the remaining benefit b. For most of the function values, e.g., $u_i(\beta)$ and $u_j(\beta)$, there is not a local iterative relationship between them. Even if the value of $u_j(\beta)$ and the link information between i and j are known, there is no formula that we can use to derive the value $u_i(\beta)$. Fortunately, we find that for a pairwise special remaining benefits b_i and b_j, there is a local relationship between $u_i(b_i)$ and $u_j(b_i)$, as shown in the following theorem.

Theorem 2: The expected utilities of two neighboring nodes i and j satisfy:

$$u_i(b_i) = p_{i,j} u_j(b_j) - c_{i,j}. \tag{5}$$

Eq. 5 is an iterative formula, by which each node i can derive its own expected utility from the expected utility value of its next-hop neighbor node. Thus, once we know the value of $u_d(b_d)$, we can use Eq. 5 to iteratively derive the value of $u_s(\beta) = u_s(b_s)$, which would achieve the same result as the direct computation according to Eq. 4. Fig. 4 shows a simple example to compute the expected utility of the delivery path "$s \rightarrow 1 \rightarrow d$" in Fig. 1 through the two methods. These results demonstrate that the direct computation and the iterative computation achieve the same result. We also compute the expected utility values of all delivery paths in Fig. 1 and list them in Fig. 5. These results prove that the messages with various benefits would have different optimal delivery paths.

Moreover, we have $u_i(b_i) < u_j(b_j)$ according to Eq. 5. This means that the expected utility values of nodes would increase along with a message delivery path. The destination node has the maximum expected utility value in the delivery path.

B. The Basic Idea

Once we have the iterative formula about the expected utility, we can derive the expected utility value of node s by applying a backward derivation algorithm. The expected utility of the destination node is first calculated. This expected utility is used as a "*seed*" to iteratively compute the expected utility value of its neighbor through Eq. 5. Then, the expected utility of next node is calculated in the same way, and so on,

path \ benefit	50-t	40-t	30-0.1t
$s\rightarrow1\rightarrow d$	7.6	1.2	0.56
$s\rightarrow2\rightarrow d$	4	1.5	1.25
$s\rightarrow2\rightarrow1\rightarrow d$	2.5	-1.5	1.7
$s\rightarrow1\rightarrow2\rightarrow d$	1.6	-2.4	0.8

Fig. 5. The expected utility values of each delivery path in Fig. 1.

until the expected utility values of all nodes are determined. Accordingly, the related optimal delivery path would be found during this iterative computation process.

However, there is a problem with this method. That is, the seed of the iterative computation process, i.e., the expected utility of destination $u_d(b_d)$, cannot be directly determined. According to our definition, the expected utility $u_d(b_d)$ is not a simple value but a function of the remaining benefit b_d. It can be calculated only when b_d is known. The remaining benefit b_d can be computed only when the delivery delay is determined. However, the delivery delay cannot be calculated since we do not know the message delivery path. A similar problem also exists in the middle of the computation process. For example, when we want to compute an expected utility $u_i(b_i)$ through Eq. 5, we need to know the remaining benefits b_i. However, b_i can be determined only when the optimal delivery path from source to node i is known.

To overcome the above problem, we extend the iterative computation of a single-point expected utility to the computation of the whole expected utility function for each node. This is feasible because the expected utility function is discrete and the range of the function parameter (i.e., the possible values of the remaining benefit) is limited. Note that the maximum remaining benefit value in the whole network is the initial benefit β. The minimum and the maximum delivery delays between pairwise neighboring nodes are one time slot and T–1 time slots, respectively. The difference of remaining benefit values of pairwise neighboring nodes thus only might be $\{\delta, 2\delta, \cdots, (T–1)\delta\}$. Moreover, each message delivery path has at most $(|V|–1)$ hops. Therefore, the remaining benefit values of each node only might be $\{\beta–(|V|–1)(T–1)\delta, \cdots, \beta–\delta, \beta\}$, denoted by Φ. Here we point out that the size of the remaining benefit set Φ is not a large value since the duty-cycle T in a duty-cycle WSN is generally much less than the number of nodes $|V|$ in order to provide a valid service. Based on this idea, our solution is presented as follows.

At the beginning, the destination node calculates the expected utility value $u_d(b)$ for each possible remaining benefit $b \in \Phi$. Then, it starts the Φ paralleled backward derivation computation processes by taking these expected utility values as the seeds. Each node uses Eq. 5 to determine its maximum expected utility values and pushes the backward computation process until the source node gets the expected utility values. Accordingly, the optimal delivery path also would be recorded. Note that the Φ paralleled backward derivation computation processes are not independent of each other. The backward derivation computation with a large-b seed will require the computation results with a low-b seed.

Algorithm 1 The Centralized TUR Algorithm

Require: $G=\langle V, W=\{\langle p_{i,j}, t_{i,j}, c_{i,j}\rangle | i,j\in V\}\rangle$, d $(\in V)$, Φ, δ.
Ensure: $u_i(b)$, $path_i(b)$ $(i\in V, b\in\Phi)$.

1: **for** each $b\in\Phi$ (in the ascending order) **do**
2: $\quad u_d(b)=b$, $u_{i(\neq d)}(b)=0$, $Q=\emptyset$;
3: $\quad$ **while** $V-Q\neq\emptyset$ **do**
4: $\quad\quad$ Find the node i with the largest $u_i(b)$ from $V-Q$;
5: $\quad\quad$ **if** $u_i(b)=0$ **then**
6: $\quad\quad\quad$ Break;
7: $\quad\quad Q=Q\cup\{i\}$;
8: $\quad\quad$ **for** each neighbor j of node i **do**
9: $\quad\quad\quad$ Compute the new utility $u'_j(b)$ using Eq. 5;
10: $\quad\quad\quad$ **if** $u_j(b) < u'_j(b)$ **then**
11: $\quad\quad\quad\quad u_j(b)=u'_j(b)$, $path_j(b)=i$;

Algorithm 2 The Distributed TUR Algorithm

Require: $G=\langle V, W=\{\langle p_{i,j}, t_{i,j}, c_{i,j}\rangle | i,j\in V\}\rangle$, d $(\in V)$, Φ, δ.
Ensure: $u_i(b)$, $path_i(b)$ $(i\in V, b\in\Phi)$.

1: **for** each node i **do**
2: $\quad$ Initialize: $u_{i(=d)}(b) = b$, $u_{i(\neq d)}(b) = 0$ $(\forall b\in\Phi)$;
3: $\quad$ **for** each time slot in T **do**
4: $\quad\quad$ **if** node i is active **then**
5: $\quad\quad\quad$ Receive new expected utilities from neighbors;
6: $\quad\quad\quad$ Compute new $u'_i(b)$ $(\forall b\in\Phi)$ using Eq. 5;
7: $\quad\quad\quad$ **if** $u_i(b) < u'_i(b)$ $(\forall b\in\Phi)$ **then**
8: $\quad\quad\quad\quad u_i(b)=u'_i(b)$;
9: $\quad\quad\quad\quad$ Determine $path_i(b)$ according to $u'_i(b)$;
10: $\quad\quad$ **if** neighbor j is active **then**
11: $\quad\quad\quad$ Send new expected utility to node j;

C. The Detailed Algorithms

With regard to our solution, we first present a centralized algorithm (Algorithm 1), and then we also give a distributed version of this algorithm (Algorithm 2).

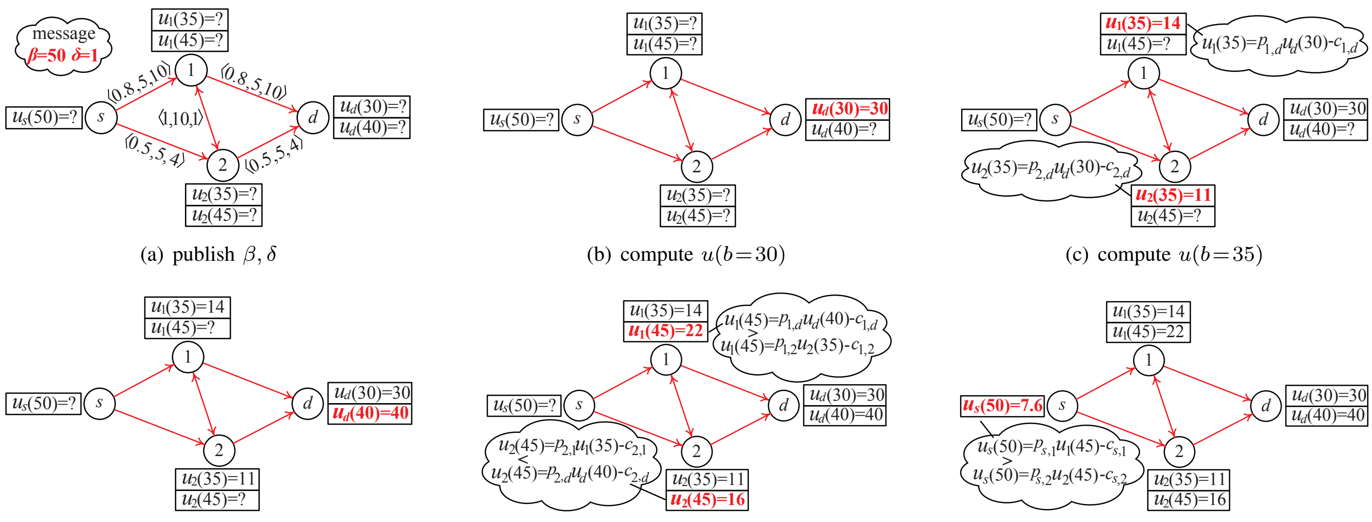
The centralized algorithm (Algorithm 1) assumes that the source node has collected the reliability, delivery delay, and transmission cost of the whole network and has constructed a weighted directed graph. Based on this graph, the centralized algorithm first computes the expected utility values of all the nodes for the minimum remaining benefit in Φ. Then, it increases the remaining benefit and computes the expected utility values for the new remaining benefits in Φ step-by-step, as shown in Step 1. For each remaining benefit b, the corresponding expected utility values can be calculated since they only depend on the expected utility values which have been computed before. Steps 2-11 give the basic process of the backward derivation computation. Step 2 makes an initialization. Steps 4-7 extend the set of nodes whose optimal expected utility values have been determined. Steps 9-11 determine the optimal expected utility value. In Step 5, if $u_i(b)=0$, we stop the current computation since the message delivery cannot achieve a positive utility. Besides, each node i records its optimal next hop in $path_i(b)$ for each remaining benefit b.

The correctness of this algorithm is straightforward. The backward iterative computation scheme and Eq. 5 can ensure the optimality of our algorithm. Moreover, the computational overhead is only $O(|\Phi||V|^2)=O(T|V|^3)$.

Algorithm 2 is a distributed solution. Each node in this algorithm initializes in Step 1, and then continuously updates its expected utility values when it becomes active (Steps 4-9) until the algorithm converges. More specifically, the node first receives the new expected utility values from its neighboring nodes in Step 5. Then, it computes its own new expected utility values according to Eq. 5 in Steps 6-9, and it meanwhile determines the optimal delivery paths. When its neighboring nodes become active, it also would notify them of its new optimal expected utility values (Steps 10-11).

Compared to the centralized algorithm, the distributed algorithm adopts a similar process to compute the expected utility values of each node, but removes the scheduling order of these expected utility values being calculated. Note that the expected utility values of nodes would strictly increase along with a message delivery path. Moreover, each expected utility can be computed only when the expected utility related to a smaller remaining benefit is computed before. These would ensure that the whole computation would not lead to a loop. All of the expected utility values would be automatically calculated in sequence due to their dependent relationships. This ensures the correctness and convergence of the algorithm. Moreover, in each round of computation, i.e., a duty-cycle T, at least one optimal expected utility value can be determined. For each remaining benefit b, the maximum expected utility value is first determined. Then, the second maximum expected utility value is determined in the next round of computation, and so on. The whole algorithm will converge by at most $O(|\Phi||V|)=O(T|V|^2)$ rounds of computation. In each round of computation, each node would receive at most $\Phi(|V|-1)$ expected utility values from its neighboring nodes. Thus, the computational overhead is $O(|\Phi||V|)=O(T|V|^2)$.

In both algorithms, we have actually calculated all possible expected utility values. In fact, many of them are useless. Thus, we can remove these useless computations to reduce the overhead. This can be realized by a flooding operation. When the source node s publishes an initial benefit value β to the network, each node i derives and records its remaining benefit b_i that might be used. After this process, we only need to compute the expected utility related to these recorded remaining benefit values when we use the TUR algorithm. As a result, many useless expected utility values would not be calculated. Fig. 6 shows the process of computing the expected utility of nodes in Fig. 1 through Algorithm 1. In Fig. 6(a), the source publishes the initial benefit value β, and each node records the remaining benefits that might be used in the following steps. In Figs. 6(b)-6(f), we compute the expected utility values of all nodes by increasing the remaining benefits step-by-step. The expected utility for the remaining benefit $b=30$ is first calculated, which is used to compute the expected utility for $b=35$. If there are multiple expected utility



(a) publish β, δ (b) compute $u(b\!=\!30)$ (c) compute $u(b\!=\!35)$

(d) compute $u(b\!=\!40)$ (e) compute $u(b\!=\!45)$ (f) compute $u(b\!=\!50)$

Fig. 6. Example: computing the expected utility of nodes in Fig. 1 for the message delivery with benefit $50-t$.

values incurred by multiple paths, the largest one is selected, as shown in Figs. 6(e) and 6(f). The example, which only contains five rounds of computation, shows that our algorithm is efficient.

IV. Extensions

In this section, we extend the TUR algorithm from the case of non-retransmission to the case with retransmission. We consider two retransmission cases. One involves the retransmission occurring within the same active time slot when a time slot is set to be large enough. Another is that the retransmission occurs at different duty-cycles when a time slot is set to be a small time interval. For both cases, we present the optimal solutions.

If the retransmission occurs within the same active time slot, it would improve the successful delivery probability and also increase the transmission cost, but it would not result in an increased delivery delay. Consider an arbitrary node i and its next-hop node j. After k-time retransmissions, the corresponding successful delivery probability becomes $1-(1-p_{i,j})^k$, and the transmission cost becomes $kc_{i,j}$. Thus, the iterative formula about the expected utility for the k-time retransmissions satisfies:

$$u_i(b_i)|_k = [1-(1-p_{i,j})^k]u_j(b_j) - kc_{i,j}. \quad (6)$$

According to Eq. 6, we can find an optimal retransmission times $\hat{k}$ to maximize the expected utility value $u_i(b_i)$. More specifically, we have the following theorem.

Theorem 3: The optimal retransmission times $\hat{k}$ for the message delivery from node i and its next-hop node j satisfies:

$$\hat{k} = \lfloor \frac{\ln c_{i,j} - \ln p_{i,j}u_j(b_j)}{\ln(1-p_{i,j})} \rfloor \text{ or } \lceil \frac{\ln c_{i,j} - \ln p_{i,j}u_j(b_j)}{\ln(1-p_{i,j})} \rceil. \quad (7)$$

If the retransmission occurs at different duty-cycles, it would not only increase the successful delivery probability and the transmission cost, but also would result in an increased delivery delay. An h-time retransmission would lead to a delivery delay hT. Accordingly, the remaining benefit of node j would be decreased by δhT. Thus, the iterative formula about the expected utility for the h-time retransmission becomes:

$$u_i(b_i)|_h = [1-(1-p_{i,j})^h]u_j(b_j - \delta hT) - hc_{i,j}. \quad (8)$$

Obviously, the optimal retransmission number $\hat{h}$ for this case must be less than $\hat{k}$ due to $u_j(b_j - \delta hT) < u_j(b_j)$. Thus, we have $\hat{h} \in [1, \hat{k}]$. Then, testing all possible $h \in [1, \hat{k}]$ to maximize $u_i(b_i)|_h$ by using Eq. 8, we can obtain the optimal retransmission number $\hat{h}$.

Note that under any circumstance, the optimal number of retransmissions can be determined locally once the expected utility value of the next hop node is given. Therefore, it can be directly embedded into TUR including both centralized and distributed algorithms. As a result, the optimal numbers of retransmission of all nodes can be determined.

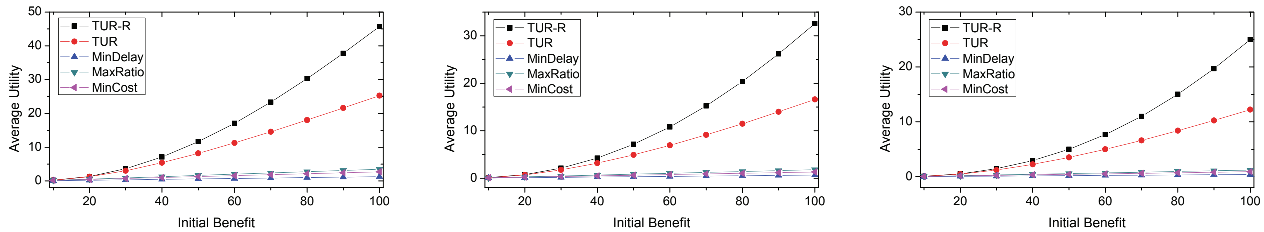
V. Performance Evaluation

In this section, we conduct extensive simulations to evaluate the performances of our proposed algorithms, including TUR and its extended version with the concern of retransmission, which is denoted by TUR-R. Besides TUR and TUR-R, we also implement three other algorithms to compare with. The compared algorithms, the evaluation methods, settings, and results are presented as follows.

A. Algorithms in Comparison

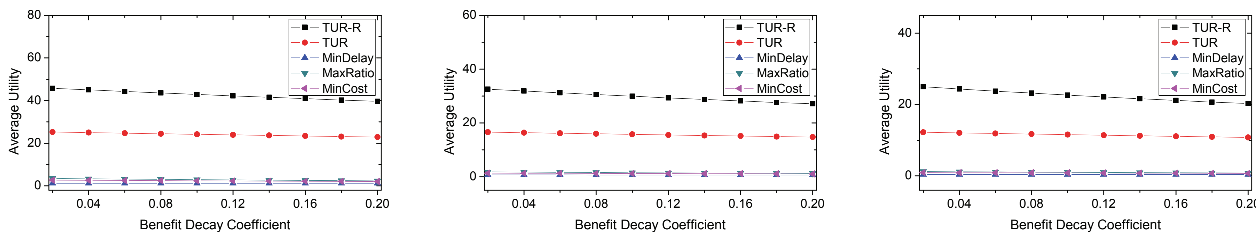
Since our proposed algorithms are the first utility-based routing algorithms designed for duty-cycle WSNs, to the best of our knowledge, there are no existing algorithms that we can compare with. Thus, according to the metrics what we are concerned with, we carefully design and implement three other algorithms: *MinDelay*, *MaxRatio*, and *MinCost*.

MinDelay is a shortest-path-based algorithm, in which each node exploits the Dijkstra algorithm to determine the shortest path w.r.t. delay, and then it lets messages be delivered along



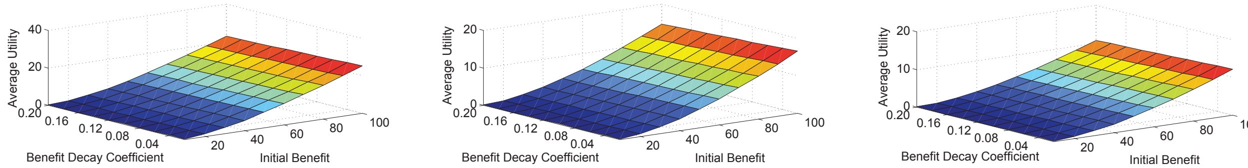
(a) Number of nodes: $|V| = 200$ (b) Number of nodes: $|V| = 400$ (c) Number of nodes: $|V| = 600$

Fig. 7. Performance comparisons of utility vs. initial benefit.



(a) Number of nodes: $|V| = 200$ (b) Number of nodes: $|V| = 400$ (c) Number of nodes: $|V| = 600$

Fig. 8. Performance comparisons of utility vs. benefit decay coefficient.



(a) Number of nodes: $|V| = 200$ (b) Number of nodes: $|V| = 400$ (c) Number of nodes: $|V| = 600$

Fig. 9. The relationship of utility vs. initial benefit and benefit decay coefficient.

TABLE I
EVALUATION SETTINGS.

Parameter name	Default value	Range
Deployment area S	$100m \times 100m$	-
Number of nodes $\|V\|$	-	200-600
Transmission radius	$2.5\sqrt{S/\|V\|}m$	-
Transmission probability	-	0.3-0.9
Transmission cost	-	1-10
Scheduling cycle	20	-
Initial benefit	100	10-100
Benefit decay coefficient	0.02	0.02-0.2
Number of messages	10,000	-

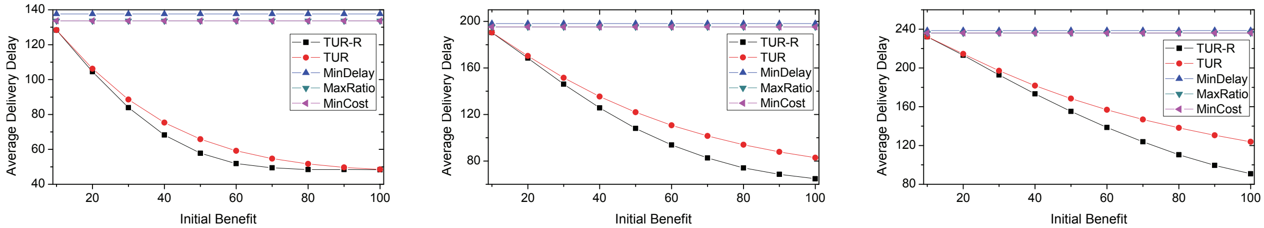
their shortest paths. MaxRatio lets messages be delivered along the paths which have the largest successful delivery probabilities. MinCost delivers messages along the paths with the smallest expected delivery cost. Both the paths with the largest delivery ratios and the paths with the minimum delivery costs are also determined by the Dijkstra algorithm.

B. Simulation Settings and Metrics

In the simulations, we deploy $|V|$ sensor nodes in a $100m \times 100m$ square area. More specifically, we divide the whole square area into $|V|$ equivalent small square lattices, and then let each node be deployed at a random position in a lattice. The transmission model of sensor nodes is the traditional disk model. That is, each pair of sensor nodes can communicate with each other only when their distance is less than a given transmission radius. We let all sensor nodes share a common transmission radius and set the radius to be 2.5 ($>\sqrt{1^2+2^2}$) times of the side length of the small square lattice. As a result, the sensor nodes in the neighboring lattices must be within the transmission radius and thus can communicate with each other. In this way, the $|V|$ sensor nodes are randomly and uniformly deployed in the whole square area while ensuring that the whole network is fully connected.

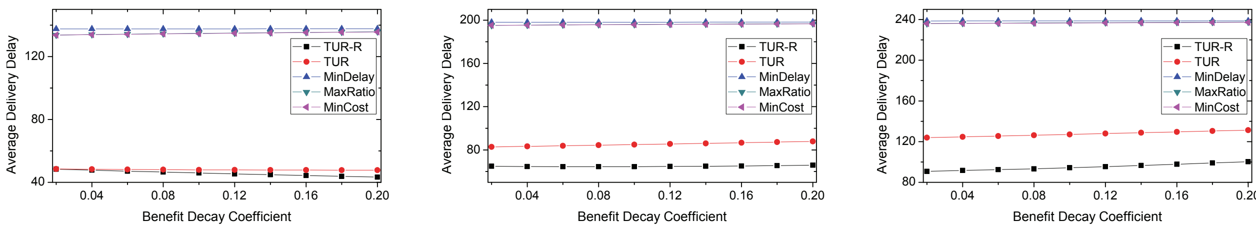
Next, we let all of the sensor nodes share a common duty-cycle and set the cycle to be 20 time slots. Each node becomes active only at one time slot in each cycle. The active time slot is randomly selected while ensuring that it is different from the neighboring nodes'. Each pair of neighboring nodes has associated with a successful transmission probability and cost, which are randomly selected from $[0.3, 0.9]$ and $[1, 10]$, respectively. In addition, the initial benefits and the benefit decay coefficients are selected from $[10, 100]$ and $[0.02, 0.2]$, respectively. All of the evaluation variables are shown in Table I.

The major metric in our simulations is the *average utility*, which is the average value of utilities of all message deliveries. In order to demonstrate that our utility-based algorithms make a good tradeoff among reliability, delay, and cost, we also compare the *average delivery delay*, *delivery ratio*, and *aver-*



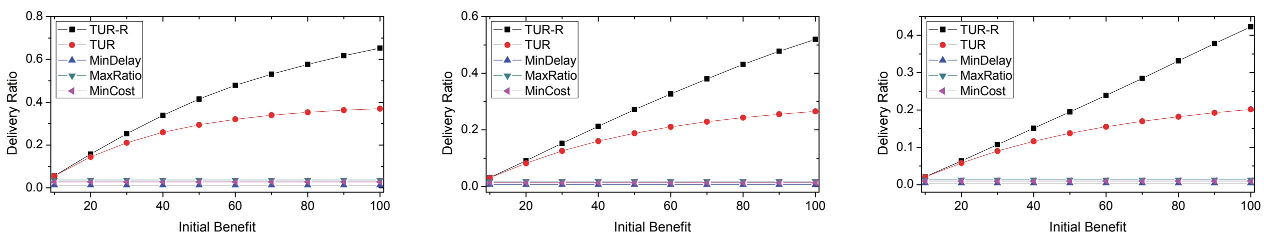
(a) Number of nodes: $|V| = 200$ (b) Number of nodes: $|V| = 400$ (c) Number of nodes: $|V| = 600$

Fig. 10. Performance comparisons of delivery delay vs. initial benefit.



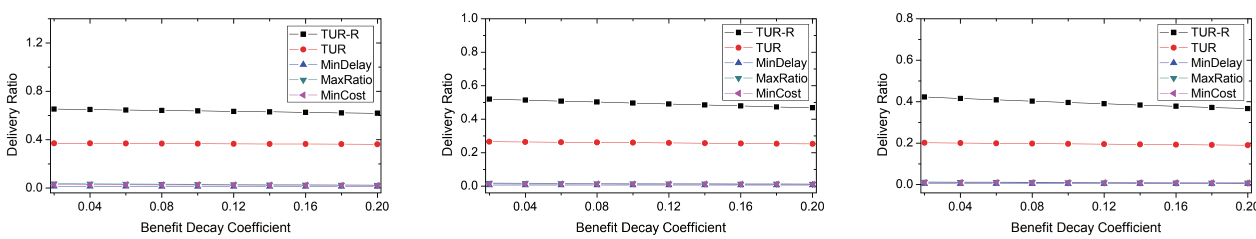
(a) Number of nodes: $|V| = 200$ (b) Number of nodes: $|V| = 400$ (c) Number of nodes: $|V| = 600$

Fig. 11. Performance comparisons of delivery delay vs. benefit decay coefficient.



(a) Number of nodes: $|V| = 200$ (b) Number of nodes: $|V| = 400$ (c) Number of nodes: $|V| = 600$

Fig. 12. Performance comparisons of delivery ratio vs. initial benefit.



(a) Number of nodes: $|V| = 200$ (b) Number of nodes: $|V| = 400$ (c) Number of nodes: $|V| = 600$

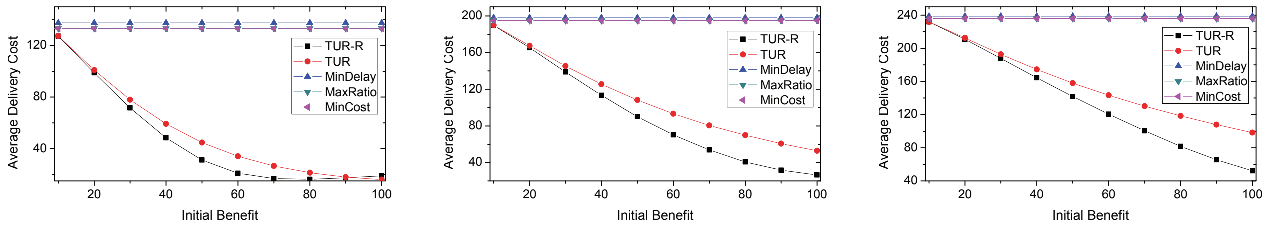
Fig. 13. Performance comparisons of delivery ratio vs. benefit decay coefficient.

age delivery cost of the five algorithms besides of the average utility. The average delivery delay and average delivery cost are the average value of delivery delay and the cost of all message deliveries. The delivery ratio is the ratio of successful deliveries and all message deliveries.

C. Evaluation Results

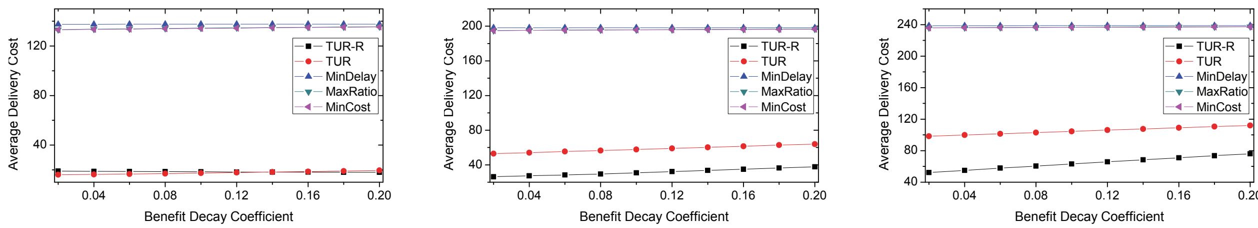
We conduct nine groups of simulations in total. In each simulation, we produce $10,000$ messages by randomly selecting the sources and destinations. For each message delivery, we record its utility, total transmission cost, and the delivery delay if the message delivery succeeds. The concrete simulations and results are presented as follows.

We first evaluate the performance on utility through three groups of simulations. The numbers of nodes are set to be $|V| = 200, 400, 600$. In the first group of simulations, we fix the benefit decay coefficient $\delta = 0.02$ and change the initial benefit value from 10 to 100, i.e., $\beta = 10, 20, \cdots, 100$, to compare the average utility of the five algorithms. The results are shown in Fig. 7. Compared with MinDelay, MaxRatio, and MinCost, TUR increases the utility by 1459.6%, 464.3%, and 637.3% on average, respectively. Compared with TUR, the TUR-R algorithm increases the utility by up to 104.3% (47.9% on average). In the second group of simulations, we fix the initial benefit $\beta = 100$ and change the benefit decay coefficient from 0.02 to 0.2. The comparison results on the average utility are shown in Fig. 8. Compared with MinDelay, MaxRatio, and MinCost, TUR increases the utility by 2305.2%, 923.9%, and 1149.9% on average, respectively. Compared with TUR,



(a) Number of nodes: $|V| = 200$ (b) Number of nodes: $|V| = 400$ (c) Number of nodes: $|V| = 600$

Fig. 14. Performance comparisons of delivery cost vs. initial benefit.



(a) Number of nodes: $|V| = 200$ (b) Number of nodes: $|V| = 400$ (c) Number of nodes: $|V| = 600$

Fig. 15. Performance comparisons of delivery cost vs. benefit decay coefficient.

the TUR-R algorithm increases the utility by up to 104.3% (87.3% on average). In the third group of simulations, we change both the initial benefit and the benefit decay coefficient at the same time to record the change of average utility of the TUR algorithm, as shown in Fig. 9. These results demonstrate the optimal utility performance of our proposed algorithms. Moreover, the larger the initial benefit and the smaller the benefit decay coefficient are, the larger the average utility would be. The results also show that retransmission can achieve an significant increase in performance.

Next, we evaluate the performances on the delivery ratio, delay, and cost through six groups of simulations. We first set the numbers of nodes to be $|V| = 200, 400, 600$, and then change the initial benefit and the benefit decay coefficient to record the average delivery delay, delivery ratio, and average delivery cost of the five algorithms, respectively. Since the delivery ratios of the five algorithms are different, it is unfair to only compare the average delivery delay and average delivery cost of the successful deliveries. In order to make the comparison fair, we also record the failed delivery with the maximum delay and cost. The results are shown in Figs. 10-15. Compared with MinDelay, MaxRatio, and MinCost, TUR decreases the delivery delay by 47.0%, 46.2%, and 46.2% on average, increases the delivery ratio by 3096.4%, 1144.1%, 1184.7% on average, and reduces the delivery cost by 59.1%, 58.4%, and 58.4% on average, respectively. Here TUR even has a much better performance with delay and cost than MinDelay and MinCost due to its good delivery ratio. The results show that the TUR algorithm has achieved good performances with reliability, delay, and cost at the same time. It makes a good tradeoff among the three factors.

VI. Related Work

The routing problem in WSNs has been studied for many years, and a lot of algorithms have been proposed for traditional non-duty-cycle WSNs [10]. Compared to the traditional WSNs, the delivery delay is an important factor in duty-cycle WSNs routing design. Without a concern for delay, these algorithms cannot work well in duty-cycle WSNs. Thus, some delay-concerned routing algorithms, including a unicast algorithm DSF [3], [7] and two flooding-based algorithms [4], [6], were proposed recently. However, compared with our utility-based algorithms, none of them, no matter the traditional algorithm or the delay-concerned routing algorithms, adopt the utility metric which takes reliability, delay, and cost into account at the same time.

The concept of utility-based routing was first proposed by M. Lu and J. Wu to balance the reliability and transmission cost of the message delivery in ad hoc networks [11]. Then, the utility-based routing algorithm is extended into the opportunistic transmission model in [1], [2]. The benefit in these utility models is unchanged during the message delivery. The utility models do not take the delivery delay into account. Without considering the delivery delay, they cannot work well in duty-cycle WSNs.

Unlike metrics used in these earlier papers, the time-sensitive utility is a composite metric which takes the delivery ratio, delay, and cost into account at the same time. To the best of our knowledge, the time-sensitive utility model and our proposed algorithms are the first utility model and utility-based routing algorithms designed for duty-cycle WSNs.

VII. Conclusion

In this paper, we propose a time-sensitive utility model for duty-cycle WSNs. Unlike the previous utility model, the time-sensitive utility model takes the delivery delay into account, which is an important metric in duty-cycle WSNs. Under this model, we present an iterative formula to compute the utility value of each message delivery. Based on the iterative formula, we design an optimal time-sensitive utility-based algorithm to deliver messages, and we extend the algorithm to the case where retransmission is allowed. Both of the algorithms can maximize the average utility values of the message deliveries, which can achieve a good tradeoff among reliability, delay, and cost. Simulations also prove the significant performance of our proposed algorithms.

ACKNOWLEDGMENT

This work is supported by the National Grand Fundamental Research 973 Program of China (Grant No.2011CB302905), the National Science and Technology Major Project (Grant No. 2011ZX03005-002), the National Natural Science Foundation of China (Grant No. 60803009), the Natural Science Foundation of Jiangsu Province in China(Grant No. BK2009150), and Chinese Universities Scientific Fund.

REFERENCES

[1] M. Lu, F. Li, and J. Wu, "Efficient opportunistic routing in utility-based ad hoc networks," *IEEE Transactions on Reliability*, vol. 58, no. 3, pp. 485–495, 2009.

[2] J. Wu, M. Lu, and F. Li, "Utility-based opportunistic routing in multi-hop wireless networks," in *IEEE ICDCS*, 2008.

[3] Y. Gu and T. He, "Dynamic switching-based data forwarding for low-duty-cycle wireless sensor networks," *IEEE Transactions on Mobile Computing*, vol. 10, no. 12, pp. 1741–1754, 2011.

[4] S. Guo, S. M. Kim, T. Zhu, Y. Gu, and T. He, "Correlated flooding in low-duty-cycle wireless sensor networks," in *IEEE ICNP*, 2011.

[5] Y. Gu, T. He, M. Lin, and J. Xu, "Spatiotemporal delay control for low-duty-cycle sensor networks," in *IEEE RTSS*, 2009.

[6] S. Guo, Y. Gu, B. Jiang, and T. He, "Opportunistic flooding in low-duty-cycle wireless sensor networks with unreliable links," in *ACM MobiCom*, 2009.

[7] Y. Gu and T. He, "Data forwarding in extremely low duty-cycle sensor networks with unreliable communication links," in *ACM SenSys*, 2007.

[8] M. Maróti, B. Kusy, G. Simon, and Á. Lédeczi, "The flooding time synchronization protocol," in *ACM SenSys*, 2004.

[9] S. Lin, J. Zhang, G. Zhou, L. Gu, T. He, and J. A. Stankovic, "Atpc: Adaptive transmission power control for wireless sensor networks," in *ACM SenSys*, 2006.

[10] J. N. Al-Karaki and A. E. Kamal, "Routing techniques in wireless sensor networks: A survey," *IEEE Wireless Communications*, vol. 11, no. 6, pp. 6–28, 2004.

[11] M. Lu and J. Wu, "Social welfare based routing in ad hoc networks," in *ICPP*, 2006.

Appendix

A. Proof of Theorem 1

We can derive Eq. 4 by computing and summing the utility values of all possible delivery cases.

If the message delivery succeeds, denoted by $s \Rightarrow d$, it means that each-hop message transmission in the path is successful. Then, the delivery delay is the sum of each-hop delay, i.e., $\sum_{i=0}^{n-1} t_{i,i+1}$. Moreover, the successful delivery probability $P|_{s\Rightarrow d}$, benefit $b|_{s\Rightarrow d}$, and total transmission cost $c|_{s\Rightarrow d}$ satisfy:

$$P|_{s\Rightarrow d}=\prod_{i=0}^{n-1}p_{i,i+1};\ b|_{s\Rightarrow d}=\beta-\delta\sum_{i=0}^{n-1}t_{i,i+1};\ c|_{s\Rightarrow d}=\sum_{i=0}^{n-1}c_{i,i+1}. \quad (9)$$

If the message delivery fails at the link "$k\rightarrow k+1$" ($0\leq k\leq n-1$), denoted by $k \not\Rightarrow k+1$, the corresponding benefit would become zero, and the total cost only contains the transmission costs for the delivery from s to h. That is:

$$P|_{k\not\Rightarrow k+1}=(1-p_{k,k+1})\prod_{i=0}^{k-1}p_{i,i+1};\ b|_{k\not\Rightarrow k+1}=0;\ c|_{k\not\Rightarrow k+1}=\sum_{i=0}^{k-1}c_{i,i+1}. \quad (10)$$

The expected utility u_s is the expected value of the utilities for the successful delivery and all possible failed deliveries. Thus, we have:

$$u_s(\beta)=P|_{s\Rightarrow d}(b|_{s\Rightarrow d}-c|_{s\Rightarrow d})+\sum_{k=0}^{n-1}P|_{k\not\Rightarrow k+1}(b|_{k\not\Rightarrow k+1}-c|_{k\not\Rightarrow k+1}). \quad (11)$$

Further, after replacing the right side of Eq. 11 by Eqs. 9-10 and by combining the related items, we can get Eq. 4.

B. Proof of Theorem 2

We derive the iterative formula about the expected utility values of two neighboring nodes i and j as follows. According to Eq. 4, we get the formulas for $u_i(b_i)$ and $u_j(b_j)$:

$$u_i(b_i)=\prod_{h=i}^{n-1}p_{h,h+1}\Big(b_i-\delta\sum_{h=i}^{n-1}t_{h,h+1}\Big)-\sum_{h=i}^{n-1}c_{h,h+1}\prod_{g=0}^{h-1}p_{g,g+1}; \quad (12)$$

$$u_j(b_j)=\prod_{h=j}^{n-1}p_{h,h+1}\Big(b_j-\delta\sum_{h=j}^{n-1}t_{h,h+1}\Big)-\sum_{h=j}^{n-1}c_{h,h+1}\prod_{g=0}^{h-1}p_{g,g+1}. \quad (13)$$

Comparing $u_i(b_i)$ and $u_j(b_j)$, we have:

$$u_i(b_i)=p_{i,j}u_j(b_j)-\prod_{h=i}^{n-1}p_{h,h+1}\Big(b_i-b_j-\delta\cdot t_{i,j}\Big)-c_{i,j}. \quad (14)$$

Since nodes i and j are adjacent in the delivery path, then according to Eq. 3, the remaining benefits of nodes i and j satisfy:

$$b_i=b_j+\delta\cdot t_{i,j}. \quad (15)$$

Therefore, by substituting Eq. 15 into Eq. 14, we can get:

$$u_i(b_i)=p_{i,j}u_j(b_j)-c_{i,j}.$$

C. Proof of Theorem 3

Based on Eq. 6, we compute the expected utility values $u_i(b_i)$ for the k-time retransmission and the $(k+1)$-time retransmission:

$$u_i(b_i)|_{k+1}=[1-(1-p_{i,j})^{k+1}]u_j(b_j)-(k+1)c_{i,j}; \quad (16)$$

$$u_i(b_i)|_{k}\ =[1-(1-p_{i,j})^{k}]u_j(b_j)-kc_{i,j}. \quad (17)$$

With Eq. 16 and Eq. 17, we have:

$$u_i(b_i)|_{k+1}-u_i(b_i)|_{k}=(1-p_{i,j})^k p_{i,j}u_j(b_j)-c_{i,j}. \quad (18)$$

Let k^* satisfy $(1-p_{i,j})^{k^*}p_{i,j}u_j(b_j)-c_{i,j}=0$, then we can get:

$$k^*=\frac{\ln c_{i,j}-\ln p_{i,j}u_j(b_j)}{\ln(1-p_{i,j})}. \quad (19)$$

According to Eq. 18, we have that $u_i(b_i)|_k<u_i(b_i)|_{k+1}$ if and only if $k<k^*$. That is, when the number of retransmissions k increases, the expected utility value $u_i(b_i)$ decreases after increasing. Moreover, the maximum expected utility value $u_i(b_i)$ can be achieved only when $k=k^*$. Since k is an integer, the optimal number of the retransmissions satisfies:

$$\hat{k}=\lfloor k^*\rfloor \text{ or } \hat{k}=\lceil k^*\rceil.$$