Deadline-sensitive Opportunistic Utility-based Routing in Cyclic Mobile Social Networks

Mingjun Xiao*†, Jie Wu†, He Huang‡, Liusheng Huang*, and Wei Yang*
*School of Computer Science and Technology / Suzhou Institute for Advanced Study, University of Science and Technology of China, Hefei, P. R. China
†Department of Computer Information and Sciences, Temple University, Philadelphia, PA 19122, USA
‡School of Computer Science and Technology, Soochow University, Suzhou, P. R. China

***Abstract*—A cyclic mobile social network (MSN) is a new type of delay tolerant network, in which mobile users periodically move around, and contact each other through their carried short-distance communication devices. In this paper, we introduce utility-based routing into cyclic MSNs, and propose a deadline-sensitive utility-based routing model. If a message is successfully delivered to its destination before a deadline, its source will receive a positive benefit as the reward. Otherwise, the source receives zero benefit. Also, each message delivery incurs a forwarding cost, no matter whether it succeeds or fails. The utility of a message delivery is defined as the benefit minus the forwarding cost. Under this model, we propose a deadline-sensitive opportunistic utility-based single-copy routing algorithm, DOUR. Each node first determines an optimal forwarding sequence, which is composed of a series of forwarding opportunities, in a distributed and greedy manner. Then, it forwards messages via these forwarding opportunities. Theoretical analysis and extensive simulations prove that DOUR can achieve the optimal utility for each message delivery. Moreover, we extend our algorithm to the case of multi-copy routing, and show that our proposed algorithms can inherently make a good tradeoff among the benefit, delay, and cost for each message delivery.**

***Index Terms*—deadline, delay tolerant network, mobile social network, opportunistic routing, utility**

I. Introduction

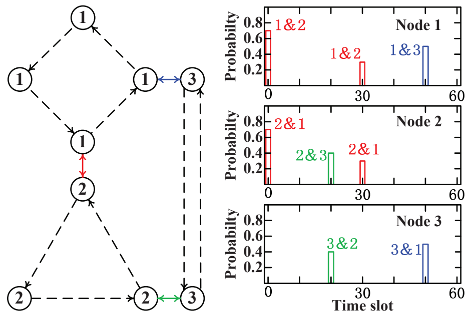
A mobile social network (MSN) is a special kind of delay tolerant network (DTN), in which mobile users walk around, and communicate with each other via Bluetooth or WiFi in their carried short-distance wireless communication devices [9], [11], [15]. As more and more users use smart phones, iPads, and mobile PCs to contact each other, MSNs have attracted considerable attention. By far, many routing algorithms have been proposed for MSNs [2], [5], [8]. However, most of these algorithms only consider simple metrics, such as delivery delay or delivery ratio, which cannot work well in real applications. It is thus necessary to design routing algorithms that take multiple factors into account at the same time.

To this end, a time-sensitive utility-based routing algorithm is proposed [16]. This is a routing scheme based on a special utility metric, which is a composition of benefit, delivery delay, and forwarding cost. In this utility model, each message has a linearly decreasing benefit along with the delivery time, and the source needs to charge a cost for each message forwarding. The utility of a message delivery is defined as the final benefit minus the total forwarding costs of this message. Moreover, an important message contains a large initial benefit, and a reliable delivery path charges a large forwarding cost. Then, this utility-based routing delivers each message by maximizing its expected utility. It can make a good balance between benefit, delivery delay, and forwarding cost [16].

In this paper, we focus on the cyclic MSN, and propose a deadline-sensitive utility model for this type of networks. A cyclic MSN means that nodes in the network periodically encounter each other so as to follow a cyclic mobility with a high probability. Many real MSNs have this cyclic re-appearance characteristic, due to the social behaviors of mobile users [7], [13]. On the other hand, in real MSN applications, users usually concern whether messages can be delivered to their destinations before some deadline. Thus, we propose a deadline-sensitive utility model. Unlike the time-sensitive utility, when a message in the deadline-sensitive utility model is successfully delivered to its destination before a given deadline, its source can receive a fixed benefit. Otherwise, the source receives zero benefit.

The key problem of our deadline-sensitive utility-based routing is how to determine a forwarding scheme that can maximize the expected utility for each message delivery. To this end, we propose an optimal opportunistic forwarding scheme to solve this problem. More specifically, we define and determine an *optimal forwarding sequence* for each message sender. The forwarding sequence consists of a series of *forwarding opportunities*, each of which includes a contact time, a node, and a contact probability, which indicates that the sender has a probability to meet and contact this node at some time. Then, if the message sender encounters a node and the corresponding forwarding opportunity belongs to the forwarding sequence, it will forward the message to this node; otherwise, it will ignore this node and wait for the next forwarding opportunity. When all nodes forward messages in this way, the message delivery can achieve the optimal utility. Based on this idea, we design an *optimal deadline-sensitive opportunistic utility-based routing* (DOUR) algorithm. The major contributions are summarized as follows.

1) We first introduce a deadline-sensitive utility model into MSN routing. Compared to previous utility models, our model takes the deadline of message delivery into account, and it can inherently balance the benefit, delivery delay, and forwarding cost of each message delivery.



(a) (b)

Fig. 1. Example: (a) shows a simple cyclic MSN with three nodes, in which the dashed lines are the trajectories of nodes (their lengths are not proportional to time), and each solid line indicates a possible contact between two nodes; (b) illustrates the precise time slots and probabilities of each contact ($T=60$).

2) We propose a distributed deadline-sensitive opportunistic utility-based single-copy routing (DOUR) algorithm. In DOUR, each node maintains an optimal forwarding sequence, and only delivers messages via the forwarding opportunities in the sequence. We also design a greedy algorithm to determine the optimal forwarding sequence for each node. Moreover, we have proven that such a routing scheme can achieve the optimal expected utility.
3) We extend the deadline-sensitive utility model to the case of multi-copy routing. Each message delivery can achieve its reward when any copy is successfully forwarded to the destination. For this case, we design a multi-copy deadline-sensitive opportunistic utility-based routing (m-DOUR) algorithm.
4) We have conducted extensive simulations on a real MSN trace to evaluate our proposed algorithms. The results show that DOUR can achieve the optimal performance. Moreover, both DOUR and m-DOUR can provide a good balance among benefit, delay, and cost.

The remainder of the paper is organized as follows. We introduce the utility model and problem in Section II. The basic idea and the detailed solution of DOUR are proposed in Sections III and IV, followed by the m-DOUR algorithm in Section V. In Section VI, we evaluate the performance of our algorithms through extensive simulations. After reviewing the related work in Section VII, we conclude the paper in Section VIII. *All proofs are presented in the Appendix.*

II. Model & Problem

In this section, we introduce the network model, and the deadline-sensitive utility model, followed by the problem.

A. Network Model

In this paper, we focus on the cyclic MSN, in which nodes periodically move around. Time is divided into equal-length time slots. Each node has some probabilities to meet other nodes at some particular time slots in each cycle. When two nodes encounter, they can form a contact and communicate with each other. Fig. 1 shows a simple example. Nodes 1, 2, and 3 move along a diamond path, a triangle path, and a linear path, respectively. The three nodes have a common motion cycle $T=60$ time slots, as shown in Fig. 1(a). Nodes 1 and

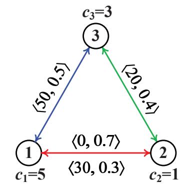


Fig. 2. The weighted graph of the MSN in Fig. 1 (destination: $d=3$; successful delivery reward $b=20$).

2 contact at time slot 0 and time slot 30 of each cycle (i.e., $[0, 60)$) in the probabilities 0.7 and 0.3, respectively. Nodes 1 and 3 contact at time slot 50 of each cycle in the probability 0.5. Nodes 2 and 3 contact at time slot 20 of each cycle in the probability 0.4, as shown in Fig. 1(b).

For simplicity, a cyclic MSN can be seen as a weighted graph, in which an edge between two nodes contains a set of probabilistic contacts. Each *probabilistic contact* is associated with a time slot and a contact probability, which indicate that two nodes have the probability to encounter and communicate with each other at the time slot. Moreover, in this paper, we assume that each node has a cost to forward messages. Thus, each node in the graph has a cost as the node weight. Fig. 2 shows the weighted graph of the cyclic MSN in Fig. 1. The weight of the edge between nodes 1 and 2 contains two probabilistic contacts $\langle 0, 0.7\rangle$ and $\langle 30, 0.3\rangle$, in which 0 and 30 are the contact time, and 0.7 and 0.3 are the corresponding contact probabilities, respectively. The node weight $c_1=5$ is the forwarding cost of node 1.

More specifically, we consider a cyclic MSN composed of n nodes. The cycle is T time slots. The corresponding weighted graph is $G=\langle V, E\rangle$, where $V=\{1, \cdots, n\}$ is the node set, and $E=\{\langle \tau_{i,j}, p_{i,j}\rangle | i, j \in V\}$ is the set of probabilistic contacts. Each pair of nodes i and j might have multiple probabilistic contacts in each cycle. We use $\langle \tau_{i,j}, p_{i,j}\rangle$ to denote a probabilistic contact between nodes i and j, in which $\tau_{i,j}$ and $p_{i,j}$ are the contact time and probability, respectively. Moreover, if there is at least one probabilistic contact between nodes i and j, we say that nodes i and j are *neighbors*, and we use N_i to denote the set of neighboring nodes of i. In addition, we use c_i to indicate the forwarding cost of node i.

B. Deadline-sensitive Utility Model

The deadline-sensitive utility model uses a composite metric, i.e., utility, to evaluate each message delivery. When a message is successfully delivered to its destination within a *deadline* (i.e., the *remaining time-to-live* (TTL) of this message), the source will receive a positive benefit as the reward. Otherwise, if the message fails to arrive at its destination before the deadline, the message will be discarded, and the failed delivery will result in zero benefit. More specifically, the concept of benefit is defined as follows.

Definition 1: The *benefit* of a message delivery refers to the real reward received by the source. Let t denote the deadline of the message delivery, and b the reward of the successful delivery. Then, the benefit, denoted by $b(t)$, satisfies:

$$b(t)=\begin{cases} b, & \text{successful delivery within time } t; \\ 0, & \text{failed delivery.} \end{cases} \tag{1}$$

TABLE I
DESCRIPTION OF MAJOR NOTATIONS.

Variable	Description
N_i	neighboring node set of node i.
b	benefit/reward for the successful delivery.
c_i	forwarding cost of node i.
t	time variable about the deadline.
τ	time constant (contact time).
$u_i(t)$, $u_i^*(t)$	expected utility for node i to send a message to its destination within the deadline t (Definition 3). The superscript means that it is optimal.
$\langle\tau, v, p\rangle$	a forwarding opportunity including time, node, and the contact probability (Definition 4).
$O_i(t)$	the set of all forwarding opportunities of node i before a deadline t (Definition 4).
$S_i(t)$, $S_i^*(t)$	forwarding sequence of node i (Definitions 5 and 7). The superscript means that it is optimal.
$u_i(t)\|_{S_i(t)}$	expected utility for node i to forward messages according to the forwarding sequence $S_i(t)$.

Each message delivery will incur a *forwarding cost*, no matter whether the delivery is successful or failed. The forwarding cost includes the carrying cost and the transmission cost of each step. Moreover, the benefit and the cost are assumed to have been unified as the same unit.

Definition 2: The *utility* of a message delivery is the benefit minus the total cost of this delivery, which means the overall gain of the message delivery. Let the total cost be c. Then, the utility, denoted by $u(t)$, satisfies:

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

The above concepts b and u are related to a message delivery from a source to a destination. For simplicity of the following discussion, we also define a virtual notion for each node, i.e., the expected utility of a node, as follows

Definition 3: The *expected utility* of a node i, denoted by $u_i(t)$, is the expected value of the utility for node i to send a message to its destination within the deadline t. The optimal expected utility of node i is denoted by $u_i^*(t)$.

Note that the above-defined utility is a function of deadline, which takes the benefit, delay, reliability, and cost of the message delivery into account. For example, node 1 in Fig. 2 has a message at time slot 0, and it wants to directly send this message to node 3 before the time slot 60, where the reward of successful delivery is $b=20$. The probability of the successful delivery is 0.5. The corresponding utility is $b-c_1=20-5=15$. The probability of the failed delivery is also 0.5. The corresponding utility is $0-c_1=0-5=-5$. Then, the expected utility of node 1 is $u_1(60)=0.5\times 15+0.5\times(-5)=5$.

In addition, we list the main auxiliary variables for ease of the following presentation in Table I.

C. Assumption and Problem

Consider an arbitrary cyclic MSN $G=\langle V,E\rangle$ with a cycle T time slots. Like previous works [2], [5], [8], we assume that each node has known its own probabilistic contacts to other nodes, which can be derived from historical meeting records. Let nodes s and d be the source and the destination, and let b and t be the reward of successful delivery and the delivery deadline, respectively. Then, our objective is to design a utility-based routing algorithm that can maximize the expected utility $u_s(t)$. For generality, we aim at maximizing $u_i(t)$ for each node i in the following sections. We only discuss the solution for a single fixed destination, which can easily be extended to the case of multiple destinations. Moreover, for simplicity, we also assume that the start time of each message delivery is time slot 0, unless otherwise stated. This will not affect the correctness of our solution, since all time slots in the paper are relative to the zero start time.

III. Overview of the Solution

In this section, we focus on the *single-copy* routing problem, and propose a distributed deadline-sensitive utility-based routing algorithm, i.e., DOUR. This algorithm adopts an optimal opportunistic routing strategy, which can let each message delivery achieve the maximum expected utility. The basic idea is presented as follows.

When each node $i \in V$ encounters a neighboring node $j \in N_i$, node i first receives the (current) optimal expected utility $u_j^*(t)$ from node j. Based on this information, node i derives its own optimal expected utility, during which an optimal forwarding sequence of node i will be determined. Then, node i sends this latest optimal expected utility to neighboring nodes for their computations. All nodes in the network iteratively conduct this operation. As a result, each node will determine an optimal forwarding sequence.

The detailed method to compute the optimal expected utility and determine the optimal forwarding sequence will be proposed in the next section. Moreover, we will also show the convergency of the iterative computation in the next section. Here, we only present the basic concept of the optimal forwarding sequence.

Definition 4: A *forwarding opportunity* of a node i, denoted by $\langle\tau, v, p\rangle$, is composed of a contact time τ, a message receiver v, and a contact probability p, which indicates that node i can send messages to node v at time τ with the contact probability p. Moreover, the *set of all forwarding opportunities* of node i for a given deadline t is denoted by $O_i(t)$.

The forwarding sequence is a sequence of forwarding opportunities of a node. That is,

Definition 5: A *forwarding sequence* of a node i, denoted by $S_i(t)$, is an ordered subset of $O_i(t)$, where all forwarding opportunities are in terms of ascending contact times. That is, $S_i(t)=\{\langle\tau_1,v_1,p_1\rangle, \langle\tau_2,v_2,p_2\rangle,\cdots, \langle\tau_m,v_m,p_m\rangle\}$, where $S_i(t)\subseteq O_i(t)$, and $0\leq\tau_1\leq\tau_2\leq\cdots\leq\tau_m\leq t$.

Here, the neighboring node of i might appear in $S_i(t)$ more than once, which is related to different forwarding opportunities. Moreover, when given an arbitrary forwarding sequence, a node will forward messages according to the following rule.

*Definition 6:*The *opportunistic forwarding rule*: for a given forwarding sequence $S_i(t)=\{\langle\tau_1,v_1,p_1\rangle, \langle\tau_2,v_2,p_2\rangle,\cdots, \langle\tau_m,v_m,p_m\rangle\}$, node i forwards messages, in turn, to nodes $v_1,v_2,\cdots,v_m$ at the time slots $\tau_1,\tau_2,\cdots, \tau_m$, until the

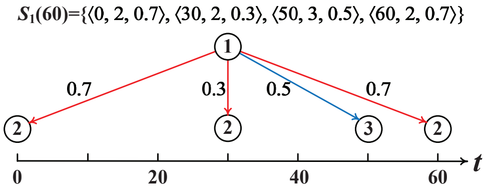


Fig. 3. An example of opportunistic forwarding

messages are successfully forwarded to some node, or all of these forwarding opportunities are exhausted.

Fig. 3 shows a forwarding sequence of node 1 in Fig. 2, i.e., $S_1(60)$=$\{\langle 0,2,0.7\rangle, \langle 30,2,0.3\rangle, \langle 50,3,0.5\rangle, \langle 60,2,0.7\rangle\}$. According to the opportunistic forwarding rule, node 1 will forward the message to node 2 at the time slots 0 and 30, to node 3 at the time slot 50, and to node 2 at the time slot 60; this is done in turn, until a forwarding succeeds, or all attempts fail.

Note that a node will achieve different expected utilities when it forwards messages via different forwarding sequences. Let the expected utility of node i related to the forwarding sequence $S_i(t)$ be denoted by $u_i(t)|_{S_i(t)}$. Then, we have:

Definition 7: The *optimal forwarding sequence*, denoted by $S_i^*(t)$, is the forwarding sequence, through which node i can achieve its optimal expected utility when it forwards messages:

$$S_i^*(t) = \underset{S_i(t)\subseteq O_i(t)}{argmax}\ u_i(t)|_{S_i(t)}. \tag{3}$$

Once the optimal forwarding sequences have been determined, all nodes forward messages following these sequences. The optimality of the forwarding sequences will ensure each node to achieve its optimal expected utility.

IV. Solution Details

In this section, we first derive the formula to compute the expected utility. Then, we determine the optimal forwarding sequence by computing the optimal expected utility for each node. Finally, we present the detailed DOUR algorithm, followed by the performance analysis.

A. Expected Utility Computation

We compute the expected utility of each node $i\in V$ through an iterative manner. That is, each node continuously collects the latest optimal expected utility of each neighboring node. Then, it derives its own optimal expected utility, based on the collected information. Here, we present the general formula to compute the expected utility for an arbitrary forwarding sequence as follows.

Theorem 1: Assume that node i has a forwarding sequence $S_i(t) = \{\langle\tau_1, v_1, p_1\rangle, \langle\tau_2, v_2, p_2\rangle, \cdots, \langle\tau_m, v_m, p_m\rangle\}$, where $v_1, v_2, \cdots, v_m \in N_i$, and their optimal expected utilities are $u_1^*(t), u_2^*(t), \cdots, u_m^*(t)$. The expected utility, which is related to this forwarding sequence, satisfies:

$$u_i(t)|_{S_i(t)} = \sum_{j=1}^{m}\prod_{h=1}^{j-1}(1-p_h)p_j u_j^*(t-\tau_j) - c_i. \tag{4}$$

Besides the iterative formula, we can directly compute the optimal expected utility of the destination, which is given by the following formula.

$$u_d^*(t) = b. \tag{5}$$

For example, the expected utility of node 1 of the cyclic MSN in Fig. 2 can be computed according to Eq.(4), where the successful message delivery reward is $b = 20$, and the forwarding sequence is $S_1(60)$ in Fig. 3. The successful probabilities of the four forwarding opportunities are 0.7, 0.09, 0.105, 0.0735, respectively. The failed forwarding probability is 0.0315. Note that the optimal expected utilities of nodes 3 and 2 are $u_3^*(t) = 20$, and $u_2^*(t) = (1-0.6^m)\times 20 - 1$, where $m = \lfloor\frac{t-20}{60}\rfloor + 1$, and $1-0.6^m$ is the successful probability of message delivery from node 2 to node 3 before the deadline. Thus, we have $u_1(60)|_{S_1(60)} = 0.7\times u_2^*(60) + 0.09\times u_2^*(30) + 0.105\times u_3^*(10) + 0.0735\times u_2^*(0) - 5 = 2.5565$.

B. Optimal Forwarding Sequence

According to Eq.(3), we derive the optimal forwarding sequence of an arbitrary node i for a given deadline t, i.e., $S_i^*(t)$, as follows.

First, we determine the set of all forwarding opportunities of node i for the deadline t, i.e., $O_i(t)$. In fact, the forwarding opportunities can be derived from the probabilistic contacts between nodes. From an arbitrary probabilistic contact $\langle\tau_{i,j}, p_{i,j}\rangle$, node i can derive the corresponding forwarding opportunities from itself to node v_j: $\langle\tau_{i,j}, v_j, p_{i,j}\rangle$, $\langle T+\tau_{i,j}, v_j, p_{i,j}\rangle$, $\cdots$, $\langle\lfloor\frac{t}{T}\rfloor T+\tau_{i,j}, v_j, p_{i,j}\rangle$, where T is the cycle. In this way, all forwarding opportunities of node i can be determined. For example, the set of all forwarding opportunities of node 1 in Fig. 2 is $O_1(60) = \{\langle 0,2,0.7\rangle, \langle 30,2,0.3\rangle, \langle 50,3,0.5\rangle, \langle 60,2,0.7\rangle\}$.

Second, we derive the optimal forwarding sequence $S_i^*(t)$ according to Eq.(3). That is, we search for the forwarding sequence from all subsets of $O_i(t)$ and compute the corresponding expected utility of node i according to Eq.(4), until we find a forwarding sequence to maximize this expected utility value. However, searching all possible subsets of $O_i(t)$ will result in an exponential computation cost. To this end, we present the following theorem, by which we can determine the optimal forwarding sequence in a greedy manner.

Theorem 2: Let $S_i(t>\tau)$ denote a subsequence of $S_i(t)$, where the contact time of each forwarding opportunity in $S_i(t>\tau)$ is larger than the time τ. Then, a forwarding opportunity $\langle\tau_j, v_j, p_j\rangle$ in $O_i(t)$ belongs to the optimal forwarding sequence $S_i^*(t)$ if and only if the expected utility of node v_j about the deadline $t-\tau_j$ minus the cost c_i is larger than the expected utility of node i through the forwarding sequence $S_i^*(t>\tau_j)$. That is,

$$\langle\tau_j, v_j, p_j\rangle \in S_i^*(t) \Leftrightarrow u_j^*(t-\tau_j) - c_i > u_i(t)|_{S_i^*(t>\tau_j)}, \tag{6}$$

where $\langle\tau_j, v_j, p_j\rangle \in O_i(t)$.

According to Theorem 2, we can efficiently derive the optimal forwarding sequence. First, we sort all forwarding opportunities of $O_i(t)$ in the ascending order of time. Then, we test these forwarding opportunities one-by-one, to determine whether they belong to the optimal forwarding sequence according to Eq.(6). The test is in the reverse order of these forwarding opportunities in $O_i(t)$, i.e., the descending order

Algorithm 1 Derive the optimal forwarding sequence $S_i^*(t)$

Require: $O_i(t)=\{\langle\tau_1,v_1,p_1\rangle,\cdots,\langle\tau_m,v_m,p_m\rangle\}$, $u_1^*(t),\cdots,u_m^*(t),c_i$
Ensure: $S_i^*(t)$, $u_i^*(t)$
1: Initialize: $S_i^*(t)=\emptyset$; $u_i(t)|_{S_i^*(t)}=-c_i$;
2: **for** $j=m$ to 1 **do**
3:     **if** $u_j^*(t-\tau_j)-c_i>u_i(t)|_{S_i^*(t)}$ **then**
4:         $S_i^*(t)=S_i^*(t)\cup\{\langle\tau_j,v_j,p_j\rangle\}$;
5:         Compute $u_i(t)|_{S_i^*(t)}$ according to Eq.(4);
6: **return** $S_i^*(t)$, $u_i^*(t)=u_i(t)|_{S_i^*(t)}$;

of time. Such a test order will ensure that, when we determine a forwarding opportunity $\langle\tau_j,v_j,p_j\rangle$, $S_i^*(t>\tau_j)$ has been determined before this. This test method can derive the optimal forwarding sequence by scanning $O_i(t)$ one time. Moreover, when the optimal forwarding sequence is derived out, the optimal expected utility $u_i^*(t)$ also is determined. That is,

$$u_i^*(t)=u_i(t)|_{S_i^*(t)}. \tag{7}$$

For instance, we can derive the optimal forwarding sequence $S_1^*(60)$ for node 1 in Fig. 2 through the above greedy method. At the beginning, we determine all forwarding opportunities, i.e., $O_1(60)=\{\langle 0,2,0.7\rangle, \langle 30,2,0.3\rangle, \langle 50,3,0.5\rangle, \langle 60,2,0.7\rangle\}$. Then, in the first step, we determine whether $\langle 60,2,0.7\rangle$ belongs to $S_1^*(60)$ or not. In fact, $u_2^*(0)-c_1=-1-5=-6<u_1(60)|_{S_1^*(t>60)}=u_1(60)|_{\emptyset}=-5$. According to Theorem 2, we have $\langle 60,2,0.7\rangle\notin S_1^*(60)$. In the second step, we determine whether $\langle 50,3,0.5\rangle$ belongs to $S_1^*(60)$ or not. Note that $u_3^*(10)-c_1=20-5=15>u_1(60)|_{S_1^*(t>50)}=u_1(50)|_{\emptyset}=-5$. Thus, $\langle 50,3,0.5\rangle\in S_1^*(60)$. In the same way, we have $\langle 30,2,0.3\rangle\notin S_1^*(60)$ and $\langle 0,2,0.7\rangle\notin S_1^*(60)$ in the third and fourth steps, respectively. Finally, we get $S_1^*(60)=\{\langle 50,3,0.5\rangle\}$. Accordingly, $u_1^*(60)=u_1(60)|_{S_1^*(60)}=5$.

C. The Detailed Algorithm

Based on the above solution, we first present the greedy algorithm to determine the optimal forwarding sequence and calculate the optimal expected utility, as shown in Algorithm 1. In Step 1, the optimal forwarding sequence and the corresponding expected utility is initialized. Steps 2-5 test each forwarding opportunity in $O_i(t)$ in a reverse order, to derive the optimal forwarding sequence by using Theorem 2. If a forwarding opportunity satisfies the condition in Step 3, this forwarding opportunity will be added into the optimal forwarding sequence in Step 4, and the corresponding expected utility is calculated in Step 5. The computational overhead of this algorithm is dominated by Step 5. In fact, the expected utility in Eq.(4) can be incrementally calculated. Thus, the computational overhead is $O(|O_i(t)|)$.

The DOUR algorithm iteratively executes Algorithm 1 to calculate the optimal expected utility and determine the optimal forwarding sequence for each node, as shown in Algorithm 2. The destination directly determines its optimal expected utility and optimal forwarding sequence in Step 1. The other node i determines its optimal expected utility and

Algorithm 2 DOUR

Require: b, c_i, t, T, $\{\langle\tau_{i,j},p_{i,j}\rangle|j\in N_i\}$ for each node $i\in V$
Initialization: For each node i **do**
1: $S_i^*(t)=\emptyset$; $u_d^*(t)=b$; $u_{i(\neq d)}^*(t)=-c_i$;
For each node i $(\in V-\{d\})$ **do**
2: **while** node i encounters a neighbor j **do**
3:     Receive $u_j^*(t)$ from j and update its local version;
4:     Determine the set of forwarding opportunities $O_i(t)$;
5:     $S_i^*(t)$, $u_i^*(t)\leftarrow$ **Algorithm** 1;
6:     **if** current forwarding opportunity $\in S_i^*(t)$ **then**
7:         Forward messages to j.

optimal forwarding sequence in Steps 2-5, when it encounters a neighboring node. If this encounter is a forwarding opportunity in the optimal forwarding sequence, the node will forward the message in Steps 6-7. The computational overhead is dominated by the execution of Algorithm 1 in Step 5, that is, $O(|O_i(t)|)$.

D. Performance Analysis

DOUR adopts an iterative computation in the whole network to derive the expected utility of each node. For the convergency of the iterative computation, we have the following theorem.

Theorem 3: The iterative computation in DOUR will converge within at most $|V|$ rounds of computation.

In Theorem 3, a round means that each pair of nodes will meet at least once. If the average contact probability per cycle between nodes is $\bar{p}$, a round of computation will take $\frac{T}{\bar{p}}$ time slots. If there is a small contact probability between a node and a neighboring node, it might take a long time to complete a round of computation. In this case, we can ignore this neighboring node's contribution to the expected utility computation. According to Eq.(4), this contribution is proportional to the contact probability of this node and is very small to be neglected. Thus, the results of DOUR will be still sufficiently good.

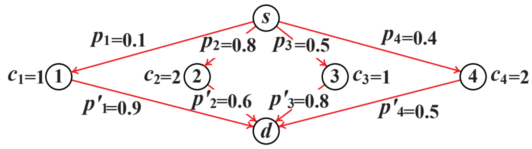
Since Theorems 1, 2 and 3 ensure the correctness and convergency of DOUR, we can straightforwardly get the optimality of DOUR without extra proofs, as follows.

Corollary 4: DOUR can achieve the optimal expected utility for each message delivery.

V. EXTENSION

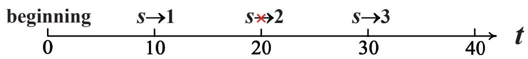
In this section, we extend our deadline-sensitive utility model into the case of multi-copy routing. Concretely, each message has multiple copies to be forwarded. If any one copy arrives at the destination before the deadline, the message delivery will achieve a positive benefit as the reward. If a copy fails to reach the destination at the deadline, it will be discarded. If all copies fail to reach the destination, the message delivery will result in zero benefit. On the other hand, the delivery of each copy has a forwarding cost. The utility is the benefit minus the forwarding cost of all copies.

For simplicity, we only focus on two-hop multi-copy routing. In fact, two-hop multi-copy routing is important in



$O_s(60)=\{\langle 10,1,0.1\rangle, \langle 20,2,0.8\rangle, \langle 30,3,0.5\rangle, \langle 40,4,0.4\rangle\}$

(a) All forwarding opportunities from s to d



$O'_s(60)=\{\langle 20,2,0.8\rangle, \langle 30,3,0.5\rangle, \langle 40,4,0.4\rangle, \langle 10,1,0.1\rangle\}$

t	0	10	20	30
contact	-	s→1	s↛2	s→3
copies	2	1	1	-
$S^*_s(60)$	⟨20, 2, 0.8⟩ ⟨30, 3, 0.5⟩	⟨20, 2, 0.8⟩	⟨30, 3, 0.5⟩	-

(b) The opportunistic forwarding process

Fig. 4. An example of 2-hop 2-copy routing (b=20).

practical applications. A recent study in [4] has revealed the performance of multi-hop routing in many real MSN traces, including Infocom trace, Dieselnet bus trace, etc. The results show that two-hop routing brings the most contributions to the routing performance among multi-hop multi-copy routing schemes, and it can even achieve a performance close to the optimal. The contribution will significantly diminish when the routing goes beyond two-hops. Thus, it is enough to only consider the two-hop multi-copy routing problem.

To this end, we consider the two-hop k-copy routing from the source s to the destination d for a given deadline t. By using the method in the last section, we can derive the set of all forwarding opportunities $O_s(t)$ from the contact information between s, d, and their common neighboring nodes. Without loss of the generality, let $O_s(t) = \{\langle\tau_1, v_1, p_1\rangle, \langle\tau_2, v_2, p_2\rangle, \cdots, \langle\tau_m, v_m, p_m\rangle\}$ ($v_1, v_2, \cdots, v_m \in N_s \cap N_d \cup \{d\}$, and $\tau_1 < \tau_2 < \cdots < \tau_m$). Under the two-hop routing model, each node v_j in $O_s(t)$ first receives messages from s at the time slot τ_j. Then, it directly forwards the messages to d with the deadline $t-\tau_j$. Thus, the successful forwarding probability from node v_j to d, denoted by p'_j, can be calculated. Then, our problem is how to maximize the utility of k-copy routing for the given $O_s(t)$ and $p'_1, \cdots, p'_m$.

However, solving the above problem optimally will lead to an exponential computation overhead. Thus, we propose a heuristic two-hop multi-copy deadline-sensitive opportunistic utility-based routing algorithm m-DOUR. The basic idea is that the source s always dynamically selects k best forwarding opportunities from $O_s(t)$, denoted by $S^*_s(t)$, and lets them transfer messages until all forwarding opportunities are exhausted. The detailed solution is presented as follows.

First, the source s sorts all forwarding opportunities in descending order of the contribution to maximizing the utility of the message delivery. Denote the new ordered set of all forwarding opportunities by $O'_s(t)$. Then, s sets $S^*_s(t)$ as the first k forwarding opportunities in $O'_s(t)$.

Here, $O'_s(t)$ is determined according to the following compare rule: if a node v_j has a large value on $p_j(p'_j b - c_j)$, it will be better than other nodes at maximizing the utility of the message delivery, and it will rank before others in $O'_s(t)$. In fact, for a given $S^*_s(t) = \{\langle\tau_1, v_1, p_1\rangle, \cdots, \langle\tau_k, v_k, p_k\rangle\}$, the utility of the message delivery satisfies:

$$u_s(t)|_{S^*_s(t)} = \left(1 - \prod_{j=1}^{k}(1 - p_j p'_j)\right) b - \sum_{j=1}^{k} p_j c_j - c_s. \tag{8}$$

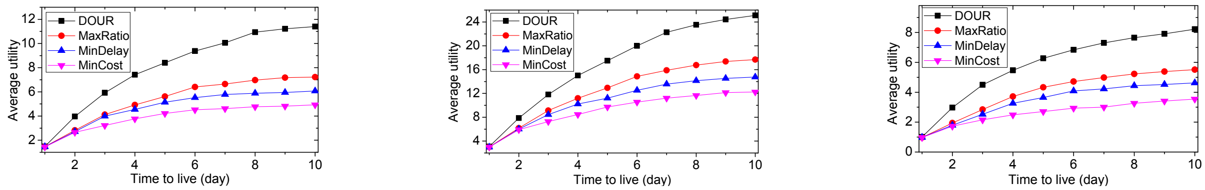
According to Eq.(8), for a node v_j in $S^*_s(t)$, the larger the value $p_j(p'_j b - c_j)$ is, the larger the utility $u_s(t)|_{S^*_s(t)}$ will be. Thus, the above rule can correctly derive $O'_s(t)$. Moreover, we term $p_j(p'_j b - c_j)$ as the *contribution factor* of node v_j (to the utility).

Second, the source s dynamically updates $S^*_s(t)$ and forwards messages opportunistically. Along with the time's eclipse, each forwarding opportunity $\langle\tau_j, v_j, p_j\rangle \in O_s(t)$ is considered. More specifically, if the forwarding opportunity $\langle\tau_j, v_j, p_j\rangle$ does not emerge at the time slot τ_j, then it will be deleted from $O'_s(t)$. Moreover, if $\langle\tau_j, v_j, p_j\rangle$ belongs to $S^*_s(t)$, it will also be deleted from $S^*_s(t)$. In this case, $S^*_s(t)$ will select the next best forwarding opportunity from $O'_s(t)$, to ensure that it always contains the best k (the number of current permitted message copies) forwarding opportunities. If the forwarding opportunity $\langle\tau_j, v_j, p_j\rangle$ does emerge at the time slot τ_j, it means that the contribution factor of node v_j to the utility becomes $1 \times (p'_j b - c_j)$ from $p_j(p'_j b - c_j)$. Then, by using the above comparison rule, s determines whether the current forwarding opportunity, under the condition that the contact between s and v_j has emerged, is better than those in $S^*_s(t)$. If this holds, s will forward a message copy to node v_j, will let k be subtracted by 1, and will delete the worst forwarding opportunity in $S^*_s(t)$ to still ensure that it contains the best new k forwarding opportunities. Otherwise, s will ignore the current forwarding opportunity $\langle\tau_j, v_j, p_j\rangle$.

Fig. 4 shows a simple example of two-hop 2-copy routing. The source s wants to forward messages to the destination d via the set of forwarding opportunities $O_s(60)$, as shown in Fig. 4(a). Fig. 4(b) shows the opportunistic routing process of m-DOUR. At the beginning, s derives the ordered forwarding opportunity set $O'_s(60)$, in which $\langle 20, 2, 0.8\rangle$ and $\langle 30, 3, 0.5\rangle$ are the best two forwarding opportunities. Then, s lets $S^*_s(60) = \{\langle 20, 2, 0.8\rangle, \langle 30, 3, 0.5\rangle\}$. At the time slot 10, the forwarding opportunity $\langle 10, 1, 0.1\rangle$ emerges. For this case, the contribution factor of node 1 becomes $1 \times (0.9 \times 20 - 1) = 17 > p_2(p'_2 b - c_2)$, which means that the current forwarding opportunity is even better than those in $S^*_s(60)$. Then, s forwards a copy to node 1, and updates $k = 1$, $S^*_s(60) = \{\langle 20, 2, 0.8\rangle\}$. At time slot 20, the forwarding opportunity $\langle 20, 2, 0.8\rangle$ fails to emerge. For this case, s deletes this opportunity from $O'_s(60)$, and lets $S^*_s(60) = \{\langle 30, 3, 0.5\rangle\}$. At time slot 30, the forwarding opportunity $\langle 30, 3, 0.5\rangle$ emerges. Then, s forwards the second copy to node 3 to end its message forwarding.

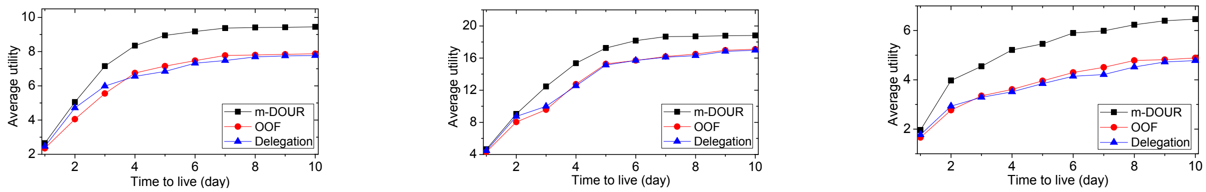
VI. Evaluation

We conduct extensive real trace-driven simulations to evaluate the performance of the proposed algorithms. The compared algorithms, the real trace that we used, the evaluation settings, and the results are presented, as follows.



(a) $b\!=\!20, c_{max}\!=\!1$ (b) $b\!=\!40, c_{max}\!=\!1$ (c) $b\!=\!20, c_{max}\!=\!2$

Fig. 5. The utility performance comparison of single-copy algorithms.



(a) $b\!=\!20, c_{max}\!=\!1$ (b) $b\!=\!40, c_{max}\!=\!1$ (c) $b\!=\!20, c_{max}\!=\!2$

Fig. 6. The utility performance comparison of multi-copy algorithms.

A. Algorithms in Comparison

Like the work in [16], we implement three other single-copy routing algorithms to evaluate the performances on the delivery ratio, delay, and cost: *MaxRatio*, *MinDelay*, and *MinCost*. They deliver each message along the paths with the largest successful delivery probability, the smallest delivery delay, and the smallest forwarding cost, respectively. Here, we cannot turn the utility-based algorithm in [16] into our utility model due to the totally different utility computation models.

Moreover, we also modify two related typical MSN routing algorithms under our utility model, to make them be comparable: *Delegation* [2] and *OOF* [8]. *Delegation* forwarding is a simple and efficient multi-copy routing algorithm. Each node in this algorithm forwards a message only when it encounters another node, which has a better metric than any one seen by this message by far. Under our utility model, we define the metric as the deadline-sensitive utility for each message delivery. *OOF* is a multi-copy opportunistic routing algorithm. By using the optimal stopping rule, this algorithm maximizes the delivery ratio, while ensuring that the number of forwardings per message does not exceed a certain threshold.

With regard to the multi-copy routing algorithms, we set the number of copies as 3, since a 3-copy routing can achieve most of the gain in real traces, and the marginal benefit of using additional copies is small, according to [4]. Moreover, in order to make the comparison fair, we also modify *Delegation* and *OOF* as two-hop routing algorithms.

B. Real Trace Used and Settings

A widely used real trace, that the nodes follow the cyclic mobility model, is the *UMassDieselNet* trace [10]. This trace contains the bus-to-bus contacts of 40 buses over 55 days. The bus system serves about ten routes. There are multiple shifts serving each of these routes. Shifts are further divided into morning, midday, afternoon, and evening sub-shifts. Drivers choose buses at random to run the sub-shifts. According to this trace, we construct the weighted contact graph, and set the simulation parameters as follows.

First, we construct the sub-shift level contact graph. We generate a mapping from the sub-shifts to the times by parsing one of the dispatch records DA_all.txt. Next, we construct a mapping from day and bus to the sub-shifts served by the bus on that day by parsing DB_sheet.txt. By using the two mappings, we turn 55 days of the bus-to-bus contacts into the contacts between sub-shifts. Moreover, if a bus is handed over from a sub-shift to another, we will create a virtual contact for them. Second, we assign the weights for the contact graph, including the contact probability of each edge, and the forwarding cost of each node. We set the time slot to be one minute. The cyclic T is set as one day. The contact probability between two sub-shifts is the ratio of the number of their contacted days and the total number of days 55. Moreover, we set a parameter, i.e., the *maximum forwarding cost* c_{max}. Then, the cost of each sub-shift is selected from the cost range $[0.1, c_{max}]$. The cost value is proportional to its degree in the contact graph. The sub-shift with the largest degree has the largest forwarding cost c_{max}.

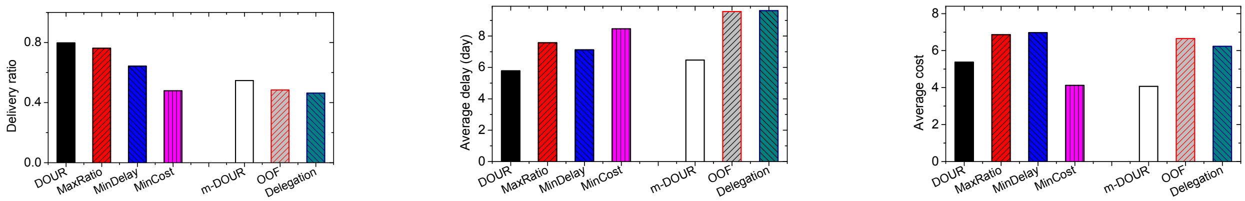
TABLE II
EVALUATION SETTINGS.

parameter name	value
the number of nodes in the trace n	92
the cycle T	1 day
the successful delivery benefit b	20, 40
the maximum forwarding cost c_{max}	1, 2
the number of messages	1,000

In each simulation, we generate 1,000 messages by randomly selecting the sources and the destinations. Each message is set with a TTL (i.e., deadline) and a successful delivery benefit/reward. The TTL of each message ranges from 1 to 10 days, and the successful delivery benefits are 20 or 40. These evaluation variables are shown in Table II.

C. Metrics and Evaluation Results

The major metric in our evaluation is the *average utility*, which is the average value of utilities of all message deliveries. In addition, we also compare the delivery ratio, the average cost, and the average delay of the above algorithms. The *delivery ratio* is the ratio of successful deliveries and all message deliveries. The *average cost* and the *average delay* are the average forwarding cost and delay of all message deliveries, respectively.



(a) Delivery ratio (b) Average delay (c) Average cost

Fig. 7. The performance comparison on the delivery ratio, the average delay, and the average cost.

We first evaluate the utility performance by setting the successful delivery benefit $b = 20, 40$ and the maximum forwarding cost $c_{max} = 1, 2$. Then, we change the TTL from 1 to 10 days. The results are shown in Figs. 5 and 6. These results show that DOUR has the largest average utility, which is much larger than the other algorithms. Moreover, m-DOUR also achieves a good utility performance, which is much larger than the other two multi-copy routing algorithms. The average utilities of all algorithms increase along with the increasing of TTL and successful delivery benefit, and decrease along with the increasing of forwarding cost.

We also evaluate the delivery ratio, average delay, and average cost through three simulations. In each simulation, we randomly set the successful delivery benefit $b = 20, 40$, and the maximum forwarding cost $c_{max} = 1, 2$. Also, we randomly select a $TTL \in [1, 10]$ for each message. We record the delivery ratio, average delay, and average cost of all message deliveries, respectively, as shown in Fig. 7. Moreover, for fairness of comparison, we let the delay of a failed message delivery be the TTL. The results show that, in addition to the optimal utility performance, DOUR achieves a good delivery ratio, average delay, and average cost, which is even better than *MaxRatio*, *MinDelay* and *MinCost*, due to its opportunistic forwarding scheme. Moreover, m-DOUR also achieves a good performance on the delivery ratio and cost.

VII. Related Work

By far, many routing algorithms have been proposed for MSNs, such as [1]–[3], [5], [6], [8], [14]. Compared to them, our utility-based routing is a routing scheme based on a special composite utility metric, which takes the deadline, benefit, and cost into consideration at the same time. In fact, the term "utility" has been widely used, but it is generally a weighted linear combination of two or more simple metrics, such as the utility metric in [12]. In contrast, our deadline-sensitive utility metric is not a simple combination of benefit, delivery delay, and cost, which is analogous to the postal service in the real world. The most related work is the time-sensitive utility-based routing in [16]. Nevertheless, this utility model is different from ours in that the benefit of each message delivery is proportional to the delivery delay. Our current model focuses on an all-or-nothing model based on a deadline, i.e., full benefit before deadline and zero benefit after deadline.

VIII. Conclusion

In this paper, we propose a deadline-sensitive utility-based routing model for cyclic MSNs, which takes into account the benefit, deadline, cost of each message delivery, as well as the reliability. Under this model, we propose a single-copy routing algorithm, DOUR, and a multi-copy routing algorithm, m-DOUR. Moreover, DOUR can achieve the maximum expected utility for each message delivery. Both of the proposed algorithms provide a good balance among the benefit, delay, and cost. Moreover, the two algorithms allow the emergent messages to be delivered along paths with a high probability for success, but at a large cost, much like the postal service in the real world. The simulations on the real MSN trace prove the significant performance of the proposed algorithms.

Acknowledgment

This work is supported by the National Natural Science Foundation of China (NSFC) (Grant No. 61379132, 60803009, 61003044, 61170058), the NSF of Jiangsu Province in China (Grant No. BK20131174, BK2009150).

References

[1] A. Balasubramanian, B. N. Levine, and A. Venkataramani. Dtn routing as a resource allocation problem. In *ACM SIGCOMM*, 2007.

[2] V. Erramilli, M. Crovella, A. Chaintreau, and C. Diot. Delegation forwarding. In *ACM MobiHoc*, 2008.

[3] W. Gao and G. Cao. User-centric data dissemination in disruption tolerant networks. In *IEEE INFOCOM*, 2011.

[4] D. Gunawardena, T. Karagiannis, A. Proutiere, E. Santos-Neto, and M. Vojnoviá. Scoop: Decentralized and opportunistic multicasting of information streams. In *ACM MobiCom*, 2011.

[5] P. Hui, J. Crowcroft, and E. Yoneki. Bubble rap: social-based forwarding in delay tolerant networks. In *ACM MobiHoc*, 2008.

[6] M. Khouzani, S. Eshghi, S. Sarkar, S. S. Venkatesh, and N. B. Shroff. Optimal energy-aware epidemic routing in dtns. In *ACM MobiHoc*, 2012.

[7] C. Liu and J. Wu. Routing in a cyclic mobispace. In *ACM MobiHoc*, 2008.

[8] C. Liu and J. Wu. On multicopy opportunistic forwarding protocols in nondeterministic delay tolerant networks. *IEEE Transactions on Parallel and Distributed Systems*, 23(6):1121–1128, 2012.

[9] Z. Lu, Y. Wen, and G. Cao. Information diffusion in mobile social networks: The speed perspective. In *IEEE INFOCOM*, 2014.

[10] J. Burgess et al. CRAWDAD data set umass/ diesel (v. 2008-09-14). Downloaded from http://crawdad. cs.dartmouth. edu/umass/diesel, Sept. 2008.

[11] T. Ning, Z. Yang, H. Wu, and Z. Han. Self-interest-driven incentives for ad dissemination in autonomous mobile social networks. In *IEEE INFOCOM*, 2013.

[12] E. Pagani and G. P. Rossi. Utility-based forwarding: a comparison in different mobility scenarios. In *MobiOpp*, 2012.

[13] T. Spyropoulos, K. Psounis, and C. S. Raghavendra. Performance analysis of mobility-assisted routing. In *ACM MobiHoc*, 2006.

[14] X. Tie, A. Venkataramani, and A. Balasubramanian. R3: robust replication routing in wireless networks with diverse connectivity characteristics. In *ACM SIGCOMM*, 2011.

[15] X. Wang, M. Chen, Z. Han, D. O. Wu, and T. Kwon. Toss: Traffic offloading by social network service-based opportunistic sharing in mobile social networks. In *IEEE INFOCOM*, 2014.

[16] M. Xiao, J. Wu, C. Liu, and L. Huang. Tour: Time-sensitive opportunistic utility-based routing in delay tolerant networks. In *IEEE INFOCOM*, 2013.

Appendix

A. Proof of Theorem 1

Consider the case that node i successfully forwards the message to a node v_j $(1 \leq j \leq m)$. This means that node i fails to forward the message via the forwarding opportunities $\langle \tau_1, v_1, p_1 \rangle, \cdots, \langle \tau_{j-1}, v_{j-1}, p_{j-1} \rangle$. Then, the corresponding successful forwarding probability is $\prod_{h=1}^{j-1}(1-p_h)p_j$. Moreover, the deadline of the message delivery becomes $t-\tau_j$ when node v_j receives the message. The expected utility for the message delivery from node v_j to the destination is $u_j^*(t-\tau_j)$. Thus, the expected utility for the message delivery from node i to the destination via node v_j is $\prod_{h=1}^{j-1}(1-p_h)p_j(u_j^*(t-\tau_j)-c_i)$. The failed probability of the whole forwarding sequence is $\prod_{j=1}^{m}(1-p_j)$, and the corresponding utility of the message delivery is $0-c_i$. Then, the expected utility of node i satisfies:

$$
\begin{aligned}
u_i(t)|_{S_i(t)} &= \sum_{j=1}^{m}\prod_{h=1}^{j-1}(1-p_h)p_j\big(u_j^*(t-\tau_j)-c_i\big) \\
&\quad + \prod_{j=1}^{m}(1-p_j)(0-c_i) \\
&= \sum_{j=1}^{m}\prod_{h=1}^{j-1}(1-p_h)p_j u_j^*(t-\tau_j)-c_i.
\end{aligned}
$$

B. Proof of Theorem 2

We first prove a lemma:

Lemma 5: Consider an arbitrary forwarding sequence $S_i(t) = \{\langle \tau_1, v_1, p_1 \rangle, \langle \tau_2, v_2, p_2 \rangle, \cdots, \langle \tau_m, v_m, p_m \rangle\}$, and let $S_i^-(t) = S_i(t) - \{\langle \tau_j, v_j, p_j \rangle\}$, where the forwarding opportunity $\langle \tau_j, v_j, p_j \rangle \in S_i(t)$. Then, we have:

$$
u_i(t)|_{S_i(t)} > u_i(t)|_{S_i^-(t)} \Leftrightarrow u_j^*(t-\tau_j)-c_i > u_i(t)|_{S_i(t>\tau_j)}. \quad (9)
$$

Proof: According to Eq.(4), we compute $u_i(t)|_{S_i(t)}$ and $u_i(t)|_{S_i^-(t)}$ as follows.

$$
\begin{aligned}
u_i(t)|_{S_i(t)} &= \sum_{k=1}^{m}\prod_{h=1}^{k-1}(1-p_h)p_k u_k^*(t-\tau_k)-c_i \\
&= \sum_{k=1}^{j-1}\prod_{h=1}^{k-1}(1-p_h)p_k u_k^*(t-\tau_k) \\
&\quad + \prod_{h=1}^{j-1}(1-p_h)\Bigg(\sum_{k=j}^{m}\prod_{h=j}^{k-1}(1-p_h)p_k u_k^*(t-\tau_k)\Bigg)-c_i. \quad (10)
\end{aligned}
$$

$$
\begin{aligned}
u_i(t)|_{S_i^-(t)} &= \sum_{k=1}^{j-1}\prod_{h=1}^{k-1}(1-p_h)p_k u_k^*(t-\tau_k) \\
&\quad + \prod_{h=1}^{j-1}(1-p_h)\Bigg(\sum_{k=j+1}^{m}\prod_{h=j+1}^{k-1}(1-p_h)p_k u_k^*(t-\tau_k)\Bigg)-c_i. \quad (11)
\end{aligned}
$$

By comparing Eqs.(10) and (11), we have:

$$
\begin{aligned}
&u_i(t)|_{S_i(t)} - u_i(t)|_{S_i^-(t)} \\
&= \prod_{h=1}^{j-1}(1-p_h)p_j\Big(u_j^*(t-\tau_j) - u_i(t)|_{S_i(t>\tau_j)} - c_i\Big). \quad (12)
\end{aligned}
$$

Then, Eq.(9) holds. ■

Now, we derive Theorem 2 from Lemma 5. First, we prove

$$
\langle \tau_j, v_j, p_j \rangle \in S_i^*(t) \Rightarrow u_j^*(t-\tau_j) - c_i > u_i(t)|_{S_i^*(t>\tau_j)}.
$$

In fact, $\langle \tau_j, v_j, p_j \rangle \in S_i^*(t)$ means $u_i(t)|_{S_i^*(t)} > u_i(t)|_{S_i^*(t)-\{\langle \tau_j, v_j, p_j \rangle\}}$ due to the optimality of $S_i^*(t)$. Then, according to Lemma 5, we have $u_j^*(t-\tau_j)-c_i > u_i(t)|_{S_i^*(t>\tau_j)}$. Second, we prove

$$
u_j^*(t-\tau_j)-c_i > u_i(t)|_{S_i^*(t>\tau_j)} \Rightarrow \langle \tau_j, v_j, p_j \rangle \in S_i^*(t)
$$

by a contradiction. Assume that $\langle \tau_j, v_j, p_j \rangle \notin S_i^*(t)$. Let $S_i(t) = S_i^*(t) \cup \{\langle \tau_j, v_j, p_j \rangle\}$. Then, we have $u_i(t)|_{S_i(t)} < u_i(t)|_{S_i(t)-\{\langle \tau_j, v_j, p_j \rangle\}}$ due to the optimality of $S_i^*(t) = S_i(t) - \{\langle \tau_j, v_j, p_j \rangle\}$. Thus, according to Lemma 5, we can get $u_j^*(t-\tau_j) - c_i < u_i(t)|_{S_i(t>\tau_j)} = u_i(t)|_{S_i^*(t>\tau_j)}$. This contradicts the condition $u_j^*(t-\tau_j)-c_i > u_i(t)|_{S_i^*(t>\tau_j)}$. Thus, the assumption about $\langle \tau_j, v_j, p_j \rangle \notin S_i^*(t)$ is wrong. Then, the theorem is correct.

C. Proof of Theorem 3

We first show that the iterative computation will not lead to a loop. Consider an arbitrary node i and a neighboring node $v_j \in N_i$, and assume that the forwarding opportunity $\langle \tau_j, v_j, p_j \rangle \in S_i^*(t)$. According to Theorem 2, we can get $u_j^*(t-\tau_j)-c_i > u_i(t)|_{S_i^*(t>\tau_j)}$. Note that $S_i^*(t>\tau_j)$ is the optimal forwarding sequence, in that the contact time is larger than τ_j. This means $u_i(t)|_{S_i^*(t>\tau_j)} = u_i^*(t-\tau_j)$. Thus, we have $u_j^*(t-\tau_j)-c_i > u_i^*(t-\tau_j)$. That is, messages are always forwarded from the nodes with low expected utilities to the nodes with high expected utilities. In other words, when we compute the expected utility of a node, we only need the information of the nodes with higher expected utilities. Thus, the iterative computation will not result in a loop.

Now, we show that the iterative process will converge within at most $|V|$ rounds of computation. In fact, there must be at least one node whose optimal expected utility and optimal forwarding sequence for a given deadline can be determined at each round of iterative computation. In the first round, neighboring nodes exchange information. Then, the destination can determine $S_d^*(t) = \emptyset$, and $u_d^*(t) = b$. Other nodes will add the destination into their forwarding sequences or let their forwarding sequences be empty. In the second round, each pair of nodes will exchange information again. For example, nodes i and v_j will compare their current $u_i^*(t-\tau_j)$ and $u_j^*(t-\tau_j)$ to determine whether $\langle \tau_j, v_j, p_j \rangle \in S_i^*(t)$ or not. There must be a node whose expected utility is the largest. Then, the forwarding sequence and expected utility of this node is optimal. Likewise, at least a node in the third round can determine its optimal expected utility and optimal forwarding sequence for an arbitrary deadline, and so on. In each round, at least one node can determine its optimal expected utility and optimal forwarding sequence. Thus, the theorem is correct.