TOUR: Time-sensitive Opportunistic Utility-based Routing in Delay Tolerant Networks

Mingjun Xiao*†, Jie Wu†, Cong Liu‡, and Liusheng Huang*
*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 Information Science and Technology, Sun Yat-sen University, Guangzhou, P. R. China
Email: xiaomj@ustc.edu.cn, jiewu@temple.edu, gzcong@gmail.com, lshuang@ustc.edu.cn

***Abstract*—In this paper, we propose a time-sensitive utility model for delay tolerant networks (DTNs), in which each message has an attached time-sensitive *benefit* that decays over time. The *utility* of a message is the benefit minus the transmission *cost* incurred by delivering the message. This model is analogous to the postal service in the real world, which inherently provides a good balance between delay and cost. Under this model, we propose a Time-sensitive Opportunistic Utility-based Routing (TOUR) algorithm. TOUR is a single-copy opportunistic routing algorithm, in which a time-sensitive *forwarding set* is maintained for each node by considering the probabilistic contacts in DTNs. By forwarding messages via nodes in these sets, TOUR can achieve the optimal expected utilities. We show the outstanding performance of TOUR through extensive simulations with several real DTN traces. To the best of our knowledge, TOUR is the first utility-based routing algorithm in DTNs.**

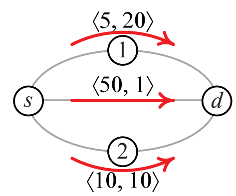
***Index Terms*—Delay tolerant networks, opportunistic routing, utility.**

I. Introduction

Utility-based routing is a special routing approach that is based on a so-called utility composite metric [1], [2]. Each message being delivered is assigned a fixed benefit as the delivery reward. The utility is in terms of the benefit minus the total transmission cost incurred by the message delivery. The objective of utility-based routing is to maximize the utility of each message delivery in a highly dynamic network. As a result, this type of routing allows more important messages to be delivered through more reliable routes at the expense of higher delivery costs. Such a routing scheme is analogous to the postal service in the real world (e.g., a high-value package usually uses registered mail for reliability at a higher cost), which provides a good balance between benefit and cost.

In this paper, we focus on utility-based routing in delay tolerant networks (DTNs). Compared to traditional wireless ad hoc networks, DTNs often experience intermittent connectivity and even long-lasting disconnections due to the mobility of the nodes. Probabilistic contact, time-varying benefit, and opportunistic forwarding are important factors in the DTN routing design. However, these factors have not been considered in current utility-based routing schemes.

To this end, we propose a time-sensitive utility model for DTN routing problems. Each message has a time-varying *benefit*, which linearly decays with time. The decay rate is called the *benefit decay coefficient*. The *utility* is the time-varying



benefit	preferred path
100-4t	$s \to 1 \to d$
100 -t	$s \to 2 \to d$

Fig. 1. A simple DTN composed of four nodes. Each path from node s to node d has an average delay and cost, labeled by $\langle average\ delay, cost\rangle$. There are two 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 will let the messages be forwarded along different paths.

benefit minus the transmission *cost* of the message. With each message delivery, the utility will gradually decrease due to continuous increases in delay and cost, and the message will be discarded when its utility becomes zero, which corresponds to the traditional deadline. *The concept of time-sensitive utility can thus be seen as a composite metric, which takes the benefit, delay, and cost into account*.

Like the previous utility model [1], [2], our time-sensitive utility model can also let messages with different benefits and benefit decay coefficients be delivered along different paths by maximizing their utilities. For example, in a simple DTN composed of four nodes, as shown in Fig.1, the message with a time-varying benefit of 100–$4t$ will be forwarded along path $s \to 1 \to d$. The corresponding utility, $100-4\cdot 5-20=60$, is larger than that of the other paths. Likewise, the messages with the benefit $100-t$ will be forwarded along path $s \to 2 \to d$.

Under the time-sensitive utility model, routing to maximize the utilities of message deliveries is very difficult, since several intractable factors (including the time-varying benefit, probabilistic contact, and cost) need to be considered. To address this unique problem, we propose a concept of time-varying optimal *forwarding set*. Each node only forwards messages to the encountered nodes in these time-varying sets, while ignoring the meetings with other nodes. Then, we design a novel Time-sensitive Opportunistic Utility-based Routing (TOUR) algorithm based on this concept, which can achieve the optimal performance. The main contributions are summarized as follows:

1) We introduce a time-sensitive utility model into DTN routing, which can inherently balance the routing process by letting important messages (with a higher ben-

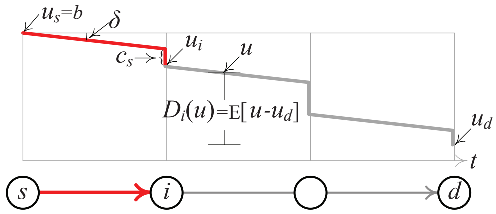


Fig. 2. Basic notions: arrival utilities u_s, u_i, and u_d; benefit decay coefficient δ; cost c_s for the message forwarding from s to i; utility difference $D_i(u)$.

efit) be delivered via short-delay, yet costly, paths.

2) We propose a distributed Time-sensitive Opportunistic Utility-based Routing (TOUR) algorithm, in which a continuous utility function needs to be maintained. A discrete sampling method is adopted to describe these functions in the implementation of TOUR in DTNs. We also analyze the approximation error incurred by the discrete process, and derive a bound.
3) We define a concept of time-sensitive optimal forwarding set, whereby TOUR achieves its optimal performance. We also design a greedy algorithm to determine the optimal forwarding set for each node.
4) We have conducted extensive simulations on multiple real DTN traces to evaluate the discrete-version of TOUR. The results show that this proposed algorithm can achieve a nearly optimal performance at a low cost. It also provides a good balance between delivery delay and transmission cost.

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

II. Model & Problem

We consider a DTN composed of probabilistically contacted mobile nodes $V=\{1,2,\cdots,i,\cdots\}$. The inter-contact time of nodes i and j is assumed to follow the exponential distribution with the parameter $\lambda_{i,j}$. If $\lambda_{i,j}>0$, we say that nodes i and j are neighbors. Let N_i denote the *neighbor set* of node i, then $j\in N_i$. Our assumption about the exponential distribution is reasonable, since previous research has proven that the exponential distribution can be seen as the approximate distribution of real DTNs for simplicity [3]–[5]. Moreover, we assume that there is a *cost* for each node i to forward a message, denoted by c_i. Based on the basic network model, we present the time-sensitive utility model in the following.

Each message contains a *benefit*, denoted by $b(t)$, to indicate the reward for delivering the message to its destination. The benefit decreases linearly as time t elapses. The initial maximum benefit is b. An important message has a large initial benefit. The decreased benefit value within each unit time interval is defined as the *benefit decay coefficient*, denoted by δ. Formally, the benefit satisfies the following formula.

TABLE I
Description of major notations.

Variable	Description
$\lambda_{i,j}$	the exponential distribution parameter of the inter-meeting time between nodes i and j.
N_i	the neighboring node set of node i: $\{j\vert\lambda_{i,j}>0\}$.
b	the initial benefit.
δ	the benefit decay coefficient.
u_i	the arrival utility of node i (Definition 1).
u, μ	a utility variable and a utility constant.
$D_i(u)$	the utility difference of node i for the utility u (Definition 2).
$R_i(u)$, $R_i^*(u)$	the forwarding set and the optimal forwarding set of node i (Definitions 3 and 4).
$D_i(u)\vert_{R(u)}$	the utility difference $D_i(u)$ for node i using the forwarding set $R(u)$ (Definition 4).
$\rho_{i,j}(u)$	the probability density function for node i forwarding the message to node $j\in R(u)$ (Eq. 4).
$p_f(\mu)$	the probability for node i to fail to forward the message before its utility becomes zero (Eq. 4).

$$b(t)=\begin{cases} b-t\cdot\delta, & t\leq b/\delta \\ 0 \quad\quad, & t>b/\delta \end{cases} \tag{1}$$

The *utility* is defined as the benefit minus the transmission cost, denoted by $u(t)$. Let c denote the total cost incurred by message forwarding until time t, then the utility satisfies:

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

In general, the utility can be seen as the remaining reward for delivering a message. When the utility of a message becomes zero, the message will be discarded.

Under the above time-sensitive utility model, we consider a message delivery from a source s to a destination d via node i. As shown in Fig. 2, we define two auxiliary notions for the simplicity of the following discussion.

Definition 1: The *arrival utility* of node i, denoted by u_i, is a virtual utility concept, which indicates the utility value as soon as the message reaches node i (even though node i has not yet charged the cost c_i).

Definition 2: The *utility difference* of node i for a utility u ($\in[0,b]$), denoted by $D_i(u)$, is the average utility reduction for a message of node i with the utility u to be delivered to (or be discarded before arriving at) node d. That is, $D_i(u)=E[u–u_d]$, where $E[\cdot]$ indicates the expected value, and u_d is the arrival utility of node d.

Here, the subscript i in $D_i(u)$ indicates that node i is the current message forwarder. In this case, u_d has not been determined yet due to the uncertain forwarding from node i to node d. Thus, $D_i(u)$ is an expected value. If there are multiple delivery paths, let $D_i(u)$ be the minimum one that is related to the best forwarding strategy, unless otherwise stated. Moreover, $D_i(u)$ is a function of u. A different u will lead to a different $D_i(u)$.

In this paper, we only consider *single-copy* message forwarding. Our objective is to maximize the expected utility of destination $E[u_d]$ when given an initial benefit b, a benefit decay coefficient δ, and pairwise source and destination nodes s and d. Since $D_s(b)=b-E[u_d]$, the objective becomes minimizing the utility difference $D_s(b)$. Without loss of generality, we aim at minimizing $D_i(u)$ for each node i in the following sections. Moreover, we only discuss the solution for the case with a single (b, δ, d), which can easily be extended to the case of multiple $(b, \delta, d)'$s.

III. The Basic Solution

In this section, we first give the overview of our solution, which includes the definition of a forwarding set and the corresponding opportunistic forwarding scheme. Then, as a part of the solution, we give a greedy search strategy, followed by an iterative computation process to determine the optimal forwarding sets of all nodes. Once the optimal forwarding sets are determined, the solution can achieve the optimal utility-based routing result.

A. Overview: Time-Sensitive Opportunistic Forwarding

We adopt the opportunistic forwarding strategy. Each node dynamically selects relays to forward messages according to their utilities. More specifically, each node i maintains a time-varying forwarding set, which is defined as follows.

Definition 3: The *forwarding set* of node i, denoted by $R_i(u)$, is a subset of N_i, which varies with a utility variable u ($\in [0, b]$); when node i meets a node in $R_i(u)$, node i will forward the message with the utility u to this node; otherwise, it ignores this contact.

This definition also implies the *opportunistic forwarding scheme*. That is, each node only forwards its messages to the encountered node in its forwarding set, while ignoring the other nodes outside of the forwarding set. Moreover, by using a different forwarding set, a node can get a different utility difference. The optimal forwarding set is defined as follows.

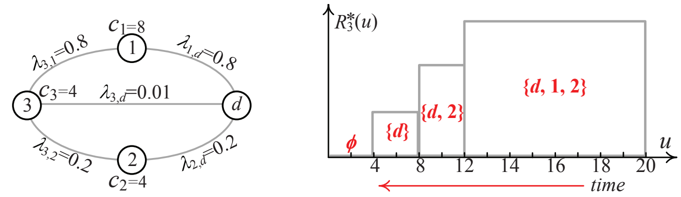
Definition 4: The *optimal forwarding set* of node i, denoted by $R_i^*(u)$, is a forwarding set whereby node i can get its minimum utility difference. Let $D_i(u)|_{R(u)}$ denote the utility difference for node i using the forwarding set $R(u)$. Then, $R_i^*(u)$ satisfies:

$$R_i^*(u) = \underset{R(u)\subseteq N_i}{argmin}\, D_i(u)|_{R(u)}. \quad (3)$$

Here, u is a utility variable that varies with time. The forwarding set is thus time-sensitive. We use u as a utility variable and μ as a utility constant in the following description. Moreover, $R(\mu)$ is used to indicate the function set $R(u)$ when $u = \mu$. $D_i(u)|_{R(\mu)}$ is used to denote the utility difference value for node i using a forwarding set $R(u)$, where $R(u) = R(\mu)$ when $u = \mu$; when $u \neq \mu$, $R(u)$ is assumed to be an unspecified forwarding set (i.e., a set we don't care about).

The optimal forwarding set $R_i^*(u)$ satisfies the rule that if node i meets any node in $R_i^*(u)$, then forwarding the message will be more beneficial than ignoring this forwarding opportunity to minimize $D_i(u)$. If node i meets a node outside of $R_i^*(u)$, then ignoring this forwarding opportunity will be better than forwarding the message. This characteristic ensures that, when all nodes forward messages according their optimal forwarding sets, the optimal utility difference will be achieved.

Based on the concept of a forwarding set, the *time-sensitive opportunistic forwarding* strategy is presented as follows. Each node first determines its optimal forwarding set (the method is given in Sections III-B and III-C). Then, each node forwards its messages to any encountered node in its optimal forwarding set.



(a) A simple DTN (b) A forwarding set

Fig. 3. An example of a time-sensitive forwarding set ($b = 20, \delta = 2$).

Fig. 3 shows an example of the forwarding set. The contact graph of a simple DTN is depicted in Fig. 3(a). The optimal forwarding set $R_3^*(u)$ (which is determined in Fig. 4) satisfies: $R_3^*(u \leq 4) = \emptyset$, $R_3^*(4 < u \leq 8) = \{d\}$, $R_3^*(8 < u \leq 12) = \{d, 2\}$, and $R_3^*(12 < u \leq 20) = \{d, 1, 2\}$, as shown in Fig. 3(b). Suppose that a message $(b = 20, \delta = 2)$ needs to be forwarded to the destination d. Along with the elapsed time, the utility will gradually decrease from 20 to 0. Accordingly, the optimal forwarding strategy for node 3 involves forwarding the message to any encountered node in $\{d, 1, 2\}$ when $u > 12$, then to any node in $\{d, 2\}$ when $8 < u \leq 12$, and so on, until the utility cannot afford the transmission cost when $u \leq 4$.

B. Determining the Optimal Forwarding Set: A Single Node

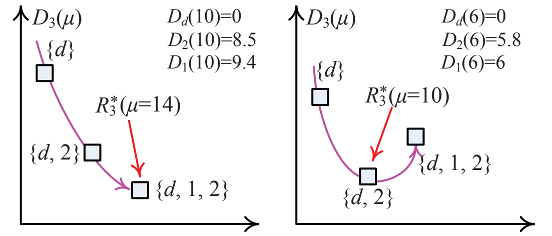
Here we determine the optimal forwarding set $R_i^*(\mu)$ and the corresponding utility difference $D_i(\mu)$ of a single node i for a given utility μ, by assuming that the utility differences of its neighboring nodes are known.

First, we present a basic formula to compute the utility difference of node i for an arbitrary forwarding set $R(u)$ ($0 \leq u \leq \mu$), i.e., $D_i(\mu)|_{R(u)}$. Consider that node i forwards a message with utility μ. Both a successful forwarding and a failed forwarding are considered. If node i encounters a node $j \in R(u)$ when the utility becomes u ($c_i \leq u \leq \mu$), node i will forward the message to node j. The corresponding utility difference is the sum of the utility reduction for this message to be forwarded from node i to node j, and the utility difference of node j about its arrival utility u_j, i.e., $\mu - u_j + D_j(u_j)$. If node i fails to meet any node in $R(u)$, the message will be discarded. The corresponding utility difference is thus μ. Then, we have:

$$D_i(\mu)|_{R(u)} = \int_0^\mu \sum_{j \in R(u)} \rho_{i,j}(u)(\mu - u_j + D_j(u_j))du + p_f(\mu)\mu, \quad (4)$$

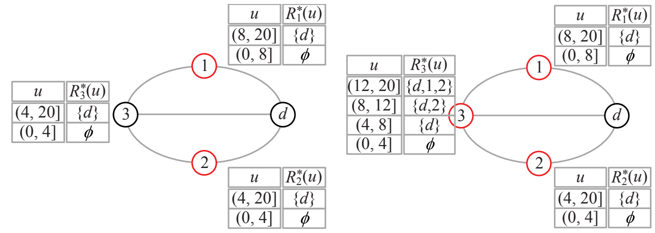
where $p_f(\mu)$ is the failed forwarding probability, and $\rho_{i,j}(u)$ is the probability density function of node i forwarding the message to node $j \in R(u)$. Here, $R(u) = \emptyset$ when $u \leq c_i$, since the utility cannot afford the cost, and the message will be directly discarded. In Eq. 4, the $\rho_{i,j}(u)$ and $p_f(\mu)$ can be calculated by the method in Section IV-B. The arrival utility of node j is $u_j = u - c_i$. Then, $D_i(\mu)|_{R(u)}$ can be calculated.

Now, based on Eqs. 3 and 4, we can determine $R_i^*(\mu)$ by searching all possible sets $R(\mu) \subseteq N_i$, computing the



(a) Determine $R_3^*(\mu=14)$ (b) Determine $R_3^*(\mu=10)$

Fig. 4. Determine $R_3^*(\mu)$ for the DTN in Fig.3(a) ($b\!=\!20, \delta\!=\!2$).



(a) Round 1 (b) Round 2

Fig. 5. Optimal forwarding sets for the DTN in Fig.3(a) ($b\!=\!20, \delta\!=\!2$).

corresponding $D_i(\mu)|_{R(\mu)}$, and finding the optimal $R(\mu)$, so as to minimize $D_i(\mu)$. Note that we only need to search $R(\mu)$. This is because the $R(u)$ with different u are independent of each other. Despite this, exhaustively searching all possible sets $R(\mu)\!\subseteq\!N_i$ will still lead to an exponential computational overhead. To this end, we propose a simple greedy method to efficiently determine the optimal forwarding set as follows.

For each given utility μ, node i compares the utility differences $D_j(\mu\!-\!c_i)$ for each neighboring node j, sorts these neighboring nodes in ascending order of their utility difference values, extends the forwarding set $R(\mu)$ by adding the ordered neighboring nodes one-by-one, and computes the value of $D_i(\mu)$ in the meantime. In the extension process of $R(\mu)$, $D_i(\mu)$ will increase after decreasing. The first inflection point is exactly the minimum value. Then, by stopping the extension process, we can deduce that the current forwarding set $R(\mu)$ is optimal. The correctness is ensured by a theorem, presented in Section IV-A. The detailed algorithm (Algorithm 1) is implemented in Section IV-C.

To illustrate, Fig. 4 shows an example of greedily determining the forwarding sets $R_3^*(\mu=14)$ and $R_3^*(\mu=10)$ for node 3 in the DTN, depicted in Fig. 3(a). For $\mu\!=\!14$, node i constructs $R_3^*(\mu\!=\!14)$ by adding nodes d, 2, 1 in turn, since $D_d(\mu-c_3) < D_2(\mu-c_3) < D_1(\mu-c_3)$, where $\mu - c_3 = 10$. Meanwhile, node i uses Eq. 4 to compute its own utility difference value for each step of the set extension. In this way, node i finds that $D_i(\mu)|_{R_3^*(14)=\{d,2\}} > D_i(\mu)|_{R_3^*(14)=\{d,2,1\}}$. Then, it can get that $R_3^*(14)\!=\!\{d, 2, 1\}$, as shown in Fig. 4(a). In the same way, node i can get $R_3^*(10) = \{d, 2\}$, as shown in Fig. 4(b). Here, $D_d(\mu\!-\!c_3)$, $D_2(\mu\!-\!c_3)$, and $D_1(\mu\!-\!c_3)$ are assumed to be known, which is derived in our next example.

C. Determining Optimal Forwarding Set: ALL Nodes

Now, we determine the optimal forwarding sets of all nodes by recursively executing the computation process in Section III-B. For generality, we compute the utility difference $D_i(u)$ of each node $i\!\in\!V$ for all possible $u\!\in\![0, b]$. The solution is just like a distributed Floyd-Warshall algorithm, which is only based on local information. Each node i records a local estimation about the utility difference of every neighboring node. These locally estimated utility differences are initialized to be the largest utility difference, i.e., $D_i(u)\!=\!u$. Then, node i initially sets its own current optimal forwarding sets to be $\{d\}$ if the destination $d \in N_i$; otherwise, it is set to be $\emptyset$. Meanwhile, node i computes the corresponding initial utility difference values by executing the local computation process in Section III-B. Moreover, upon encountering a neighboring node j, node i will update the local records about the utility difference $D_j(u)$. Then, node i re-computes its own utility difference $D_i(u)$ and updates its current optimal forwarding sets by executing the local computation process in Section III-B. By repeating this iterative process, each node will finally get its optimal forwarding set. The detailed algorithm is presented in Section IV-B.

It should be pointed out that the iterative computation process in this solution will not result in a loop. Moreover, it will converge within at most $|V|\!-\!1$ rounds of iterative computation (a round means that each pairwise neighboring node has encountered the other at least once). The convergence of the solution is ensured by a theorem, presented in Section IV-A.

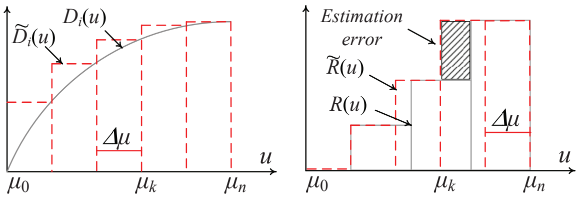
Here, we show an example to iteratively determine the optimal forwarding sets for the DTN in Fig. 3(a), where $b\!=\!20$ and $\delta = 2$. In the first round, node 1 first lets its forwarding set be $\{d\}$, or $\emptyset$ if the utility cannot afford the transmission cost. Then, it derives the corresponding utility difference by Eq. 4 to get $D_1(u)$ ($D_1(8 < u \leq 20) = 10.5 - 2.5e^{0.4(8-u)}$; $D_1(u\!\leq\!8)\!=\!8$). Likewise, nodes 2 and 3 also get their utility differences. After meeting all of their neighbors, nodes 1 and 2 will find out that their utility differences are the minimum among all encountered nodes (besides d), respectively. Then, they could ensure that their forwarding sets and utility differences are optimal. After the first round, node 3 knows $D_1(u)$ and $D_2(u)$ (e.g., $D_1(10)\!=\!9.4$, which is related to Fig. 4(a)). In the second round, it will update its own forwarding set and utility difference. Accordingly, it will get its optimal forwarding set. Fig. 5 lists the forwarding sets of all the nodes, excluding the destination.

IV. Solution Details

In this section, we first prove that our solution is optimal, and then we provide an implementation, i.e., the TOUR algorithm, in which a discrete process is adopted to approximately calculate the expected utility difference values. We also analyze the estimation error of this algorithm.

A. Proof of the Optimality

Our solution is based on the opportunistic forwarding strategy. Each node greedily determines a time-sensitive optimal forwarding set. Note that, according to the definition of a forwarding set, once the forwarding sets of all nodes are optimal, the optimal performance will be achieved. Thus, we only need to prove that our solution to greedily determining



(a) Discretization: $D_i(u)$ (b) Discretization: $R(u)$

Fig. 6. An example of discretization.

the optimal forwarding set in Section III-B is correct, and that the iterative computation process in Section III-C will converge. Once we have the two results, we can conclude that our solution is optimal.

First, to greedily determine the optimal forwarding set of each node in Section III-B, we have the following theorem.

Theorem 1: Assume that $N_i = \{1, 2, \cdots, m\}$ and $D_1(\mu - c_i) < \cdots < D_m(\mu - c_i)$ $(\mu \in (c_i, b])$. Then, we have:

1) $\exists k$ $(1 \leq k \leq m)$, $R_i^*(\mu) = \{1, 2, \cdots, k\}$;

2) $D_i(\mu)|_{R_i(\mu)=\{1,\cdots,h\}} > D_i(\mu)|_{R_i(\mu)=\{1,\cdots,h+1\}}$ for $\forall h \in [1, k)$.

Theorem 1 shows that if we add the neighboring nodes one-by-one from N_i to the forwarding set in an ascending order of their utility difference values, the utility difference of node i will continuously decrease until it reaches its minimum value. Then, the corresponding forwarding set is the optimal. Our solution, which is based exactly on this characteristic, is thus correct.

Second, we also have the following theorem about the convergence of the iterative computation in the whole network.

Theorem 2: The iterative computation will not lead to a loop and will converge within at most $|V| - 1$ rounds of computation.

According to Theorem 2, although all nodes determine their optimal forwarding sets based on their local information, they can still obtain the optimal results after at most $|V|-1$ rounds of computation.

B. Discrete Processing

Note that the utility differences and forwarding sets in our solution actually are the continuous functions for utility. Thus, we adopt a discrete sampling method to describe them in the detailed implementation of TOUR. We set $n+1$ discrete sampling points about utility $\{\mu_0 = 0, \cdots, \mu_k = k\Delta\mu, \cdots, \mu_n = n\Delta\mu\}$, where $\Delta\mu$ is the sampling utility interval, i.e., $\Delta\mu = \frac{b}{n}$. Then, an arbitrary utility difference $D_i(u)$ is approximated by the following utility difference, as shown in Fig. 6(a):

$$\tilde{D}_i(u \mid \mu_{k-1} < u \leq \mu_k) = D_i(\mu_k), \quad 1 \leq k \leq n. \tag{5}$$

In the same way, an arbitrary forwarding set $R(u)$ is approximated by the following forwarding set, as shown in Fig. 6(b):

$$\tilde{R}(u \mid \mu_{k-1} < u \leq \mu_k) = R(\mu_k), \quad 1 \leq k \leq n. \tag{6}$$

Based on the approximate utility difference $\tilde{D}_i(u)$ and forwarding set $\tilde{R}(u)$, we can derive the discrete-version formula of Eq. 4. Let the discrete-version formulas of the forwarding probability density function $\rho_{i,j}(u)$ and the failed forwarding probability $p_f(\mu)$ in Eq. 4 be denoted by $\tilde{\rho}_{i,j}(u)$ and $\tilde{p}_f(\mu)$. Then, we have:

$$\tilde{D}_i(\mu)|_{\tilde{R}(u)} = \int_0^{\mu} \sum_{j \in \tilde{R}(u)} \tilde{\rho}_{i,j}(u)(\mu - u_j + \tilde{D}_j(u_j))du + \tilde{p}_f(\mu)\mu. \tag{7}$$

Now, we derive the formulas to compute $\tilde{\rho}_{i,j}(u)$ and $\tilde{p}_f(\mu)$ in Eq. 7. Here, we only focus on the case where $\mu = \mu_m$ $(0 \leq m \leq n)$, since the other μ will not be used in the discrete-version computation. Note that $\tilde{p}_f(\mu)$ is the failed forwarding probability. This means that node i did not meet any node in the forwarding set $\tilde{R}(u)$ before the utility of the message decreases from μ to zero. For each utility reduction interval $(\mu_{h-1}, \mu_h]$ $(1 \leq h \leq m)$, the failed forwarding probability is $e^{-\sum_{j \in \tilde{R}(u_h)} \lambda_{i,j} \Delta t}$, where Δt is the time at which the utility decreases from μ_h to μ_{h-1}, i.e., $\Delta t = \frac{\Delta\mu}{\delta}$. Thus, $\tilde{p}_f(\mu)$ satisfies:

$$\tilde{p}_f(\mu) = e^{-\sum_{h=1}^{m} \sum_{j \in \tilde{R}(\mu_h)} \lambda_{i,j} \Delta\mu/\delta}. \tag{8}$$

In the same way, we derive the formula of $\tilde{\rho}_{i,j}(u)$, which is the probability density function of node i forwarding the message to node $j \in \tilde{R}(u)$. Without loss of generality, we let $u \in (\mu_{k-1}, \mu_k]$, where $1 \leq k \leq m$. Then, $\tilde{\rho}_{i,j}(u)$ satisfies:

$$\tilde{\rho}_{i,j}(u) = \lambda_{i,j}/\delta \cdot e^{-\sum_{h=k+1}^{m} \sum_{j \in \tilde{R}(\mu_h)} \lambda_{i,j} \Delta\mu/\delta - \sum_{j \in \tilde{R}(\mu_k)} \lambda_{i,j}(\mu_k - \mu)/\delta}. \tag{9}$$

Algorithm 1 Determining the optimal forwarding set

Require: b, δ, n, c_i, N_i, $\{\lambda_{i,j} | j \in N_i = \{1, \cdots, m\}\}$, $\tilde{D}_1(\mu_k) < \cdots < \tilde{D}_m(\mu_k)$ $(0 \leq k \leq n)$

Ensure: $\tilde{R}_i(\mu_k)$, $\tilde{D}_i(\mu_k)$ $(0 \leq k \leq n)$

1: **for** $k = 0, \cdots, n$ **do**
2: Initialize $\tilde{R}_i(\mu_k)$, and compute $\tilde{D}_i(\mu_k)$ by Eq.7;
3: **for** $j = 1, \cdots, m$ **do**
4: $\tilde{R}_i(\mu_k) = \tilde{R}_i(\mu_k) + \{j\}$;
5: Incrementally compute $\tilde{D}_i(\mu_k)$ by Eq. 7;
6: **if** $\tilde{D}_i(\mu_k)$ increases **then**
7: Break;
8: **return** $\tilde{R}_i(\mu_k)$, $\tilde{D}_i(\mu_k)$;

C. Algorithm Implementation

Based on our optimal solution and the discrete process, we now present the TOUR algorithm, i.e., Algorithm 2 and its sub-process, Algorithm 1.

Given the utility difference values of node i's neighbors, $\tilde{D}_j(\mu_k)$ $(0 \leq k \leq n,\ j \in N_i)$, Algorithm 1 outputs the optimal forwarding set $\tilde{R}_i(\mu_k)$ and utility difference $\tilde{D}_i(\mu_k)$ $(0 \leq k \leq n)$. $\tilde{R}_i(\mu_k)$ is determined by using the greedy search strategy that is based on Theorem 1. In Step 2, $\tilde{R}_i(\mu_k)$ and $\tilde{D}_i(\mu_k)$ are initialized. From Steps 3 to 7, $\tilde{R}_i(\mu_k)$ is extended by greedily adding a neighboring node. With the expansion of $\tilde{R}_i(\mu_k)$, the corresponding $\tilde{D}_i(\mu_k)$ decreases step-by-step until the optimal value is found in Step 7. Theorem 1 ensures the correctness of this algorithm.

In Algorithm 2, each node i initializes $\tilde{R}_i(\mu_k)$ and $\tilde{D}_i(\mu_k)$ $(0 \leq k \leq n)$ in Step 1. Then, in the routing phase, the node receives the new version of the utility difference from

Algorithm 2 TOUR

Require: b, δ, n, c_i, N_i, $\{\lambda_{i,j}|j\in N_i\}$
Ensure: $\tilde{R}_i(\mu_k)$, $\tilde{D}_i(\mu_k)$ $(0\le k\le n)$
  For each node i $(\neq d)$ **do**
1: **Initialize**:$\tilde{R}_i(\mu_k)=\emptyset$, $\tilde{D}_i(\mu_k)=\mu_k$ $(0\le k\le n)$
  Routing:
2: **while** node i encounters a neighbor j **do**
3:   Send $\tilde{D}_i(\mu_k)$ to and receive $\tilde{D}_j(\mu_k)$ $(0\le k\le n)$ from j;
4:   **if** $\tilde{D}_j(\mu_k)$ is different from the local version **then**
5:     Update local $\tilde{D}_j(\mu_k)$;
6:     $\tilde{R}_i(\mu_k)$, $\tilde{D}_i(\mu_k)$ $\leftarrow$ **Algorithm** 1;
7:   **for** each message *msg* in node i **do**
8:     Get the current utility u of *msg*;
9:     Derive out k to make $u\in(\frac{(k-1)b}{n},\frac{kb}{n}]$;
10:     **if** $j\in\tilde{R}_i(\mu_k)$ **then**
11:       Send *msg* to j;

its encountered neighbor in Step 3, and updates its own forwarding sets and utility differences, if needed, by executing Algorithm 1 in Steps 4-6. The message forwarding is executed in Steps 7-11. The forwarding set is first found in Step 9. Then, the message will be forwarded if the encountered node belongs to this forwarding set. The correctness of this algorithm is ensured by Theorem 1. Theorem 2 ensures the convergence of this algorithm. The computational overhead of TOUR is dominated by Step 6, which is $O(n|V|^2)$. Moreover, TOUR will converge within at most $|V|-1$ rounds of computation. If there is a neighboring node whose inter-meeting probability to node i is very small, a round of computation might take a long time to converge. Despite this, the results of TOUR are still good enough when we ignore this node's contribution to the results. This is because, the smaller the inter-meeting probability is, the smaller the weight of the node's contribution (to the results) will be.

D. Estimation Error

With regards to the discrete process, we make a detailed estimation error analysis and get the following results.

Theorem 3: Let $\Lambda = max\{\sum_{j\in N_i}\lambda_{i,j}|i\in V\}$; then, the estimation error of TOUR will be less than $|V|b(e^{\frac{\Lambda b}{n\delta}}-1)$.

Now, for a given arbitrary error ε, we can get a bound:

$$n_0=\frac{\Lambda b}{\delta\ln(1+\varepsilon/(|V|b))}. \tag{10}$$

When $n>n_0$, the estimation error of the TOUR algorithm will be less than ε, according to Theorem 3.

V. Evaluation

In this section, we conduct extensive real trace-driven simulations to evaluate the performance of the proposed algorithm. First, we present the compared algorithms, followed by a discussion of the real traces that we used. Second, we discuss our evaluation methods and settings. Finally, our evaluation results are shown in different perspectives to provide better insight.

TABLE II
Statistics in Three Cambridge Haggle Traces.

Trace	Contacts	Length(D) (d.h:m.s)	Routing nodes	External nodes
Intel	2,766	4.3:48.32	9	128
Cambridge	6,732	6.1:34.2	12	223
Infocom	28,216	2.22:52.56	41	264

A. Algorithms in Comparison

We first implement TOUR with 10 discrete sampling points and its optimal version, denoted by TOUR-OPT. Here, we use a TOUR with 100 discrete sampling points to imitate the optimal TOUR. In fact, the simulation results show that 100 discrete sampling points are sufficient. Without considering the delivery delay, the existing utility-based routing algorithms in traditional ad hoc networks cannot work in DTNs. Thus, we also carefully design three other utility-based routing algorithms to examine the compared performance of our algorithms: *SimpleUtility*, *MiniCost*, and *MinDelay*.

In SimpleUtility, the utility difference of each node is set to be the sum of the forwarding cost and the decreased benefit in terms of delay. Each node will forward the message to an encountered node if the utility difference of this node is greater than that of the encountered node. In MiniCost, each node forwards the message to an encountered node if the forwarding cost of this node is greater than that of the encountered node. A node in MinDelay will forward the message if the expected delivery delay of this node is greater than that of the encountered node.

B. Real-traces Used

The *Cambridge Haggle Trace* [6] includes a total of five traces of Bluetooth device connections by people carrying mobile devices (iMotes) over a certain number of days. These traces are collected by different groups of people in office environments, conference environments, and city environments, respectively. Bluetooth contacts were classified into two groups: iMote sightings of other iMotes are called *internal contacts*, while sightings of other Bluetooth devices are called *external contacts*. Since there is no record of contact between non-iMotes, we only use the iMotes as *forwarding nodes*. Other nodes, or *external nodes*, can only be assigned as destinations. Table II shows some statistics from the traces.

The *UMassDieselNet Trace* [7] contains the bus-to-bus contacts (the durations of which are relatively short) of 40 buses. Our simulations are performed on traces collected over 55 days during the Spring 2006 semester, with weekends, Spring break, and holidays removed due to reduced schedules. The bus system serves approximately ten routes. There are multiple shifts serving each of these routes. Shifts are further divided into morning (AM), midday (MID), afternoon (PM), and evening (EVE) sub-shifts. Drivers choose buses at random to run the AM sub-shifts. At the end of the AM sub-shift, the bus is often handed over to another driver to operate the next sub-shift on the same route, or on another route.

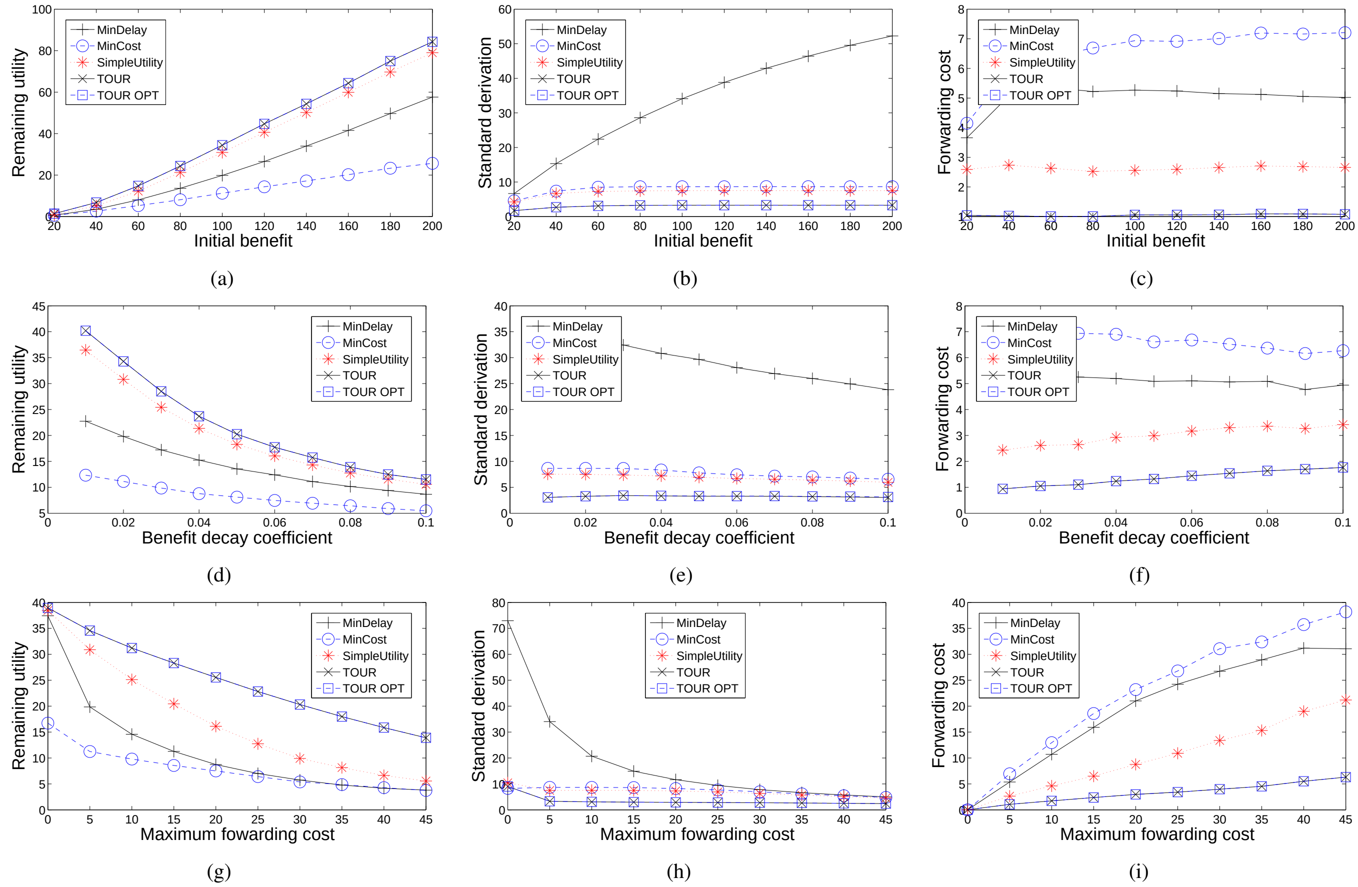


Fig. 7. Remaining utilities, standard derivations of node forwardings, and forwarding costs in the Cambridge Haggle Imote Infocom trace.

TABLE III
EVALUATION SETTINGS.

parameter name	default	range
initial benefit	100	20-200
maximum forwarding cost	5	0-45
benefit decay coefficient (per min)	0.02	0.01-0.1
number of messages	30,000	

C. Simulation Method and Settings

In our evaluations, each node is initialized with a virtual forwarding cost. We first define a *maximum forwarding cost*, which is the forwarding cost of the node with the highest contact probability with other nodes. Then, the forwarding cost of each node is defined to be proportional to its contact opportunity with other nodes. As a result, a node with a high contact probability to others has a large forwarding cost. In addition to the maximum forwarding cost, the other variables include the *benefit decay coefficient* and the *initial benefit*.

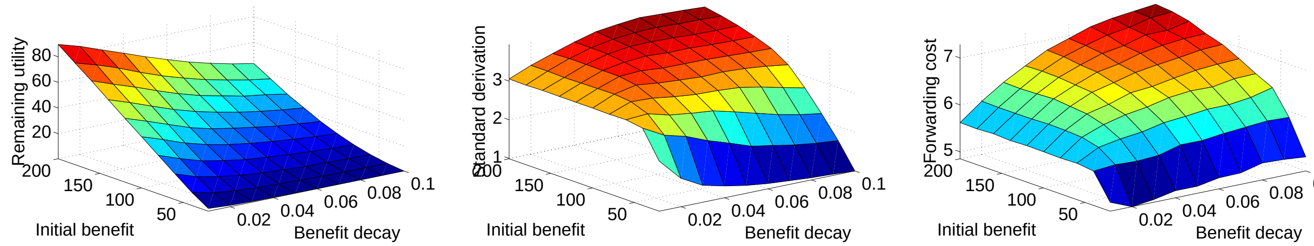
In each evaluation, some of the variables change in their range, while other variables are fixed at their default values. The default values and the changing ranges of the evaluation variables are shown in Table III.

Each simulation result is averaged from 30,000 randomly generated messages, whose sources are assigned evenly among internal nodes, and whose destinations are assigned evenly among all nodes.

D. Results on Utility, Derivation, and Cost

Unlike traditional DTN routing algorithms, utility-based routing algorithms mainly focus on the gain of a message delivery. Maximizing the *remaining utility* (i.e., the arrival utility of the destination) is thus the basic goal of our algorithms. Besides the metric, two additional metrics (standard derivation of node forwarding and forwarding cost) are adopted for our evaluation. The former metric measures the standard derivations of the number of messages forwarded by each node. This metric measures how well the algorithms can schedule different message deliveries (which are assigned different initial benefits, benefit decay coefficients, and maximum forwarding costs) to different forwarding paths. The latter metric measures how well the algorithms route the message to nodes with low transmission cost. Due to space limitations, we only provide the most representative evaluation results here.

It is shown in Figures 7(a)-7(i) that (1) TOUR and TOUR-OPT have the largest remaining utility, followed by SimpleUtility; all of these are much larger than MinDelay and MinCost, which do not maximize utility; (2) TOUR and TOUR-OPT have the smallest standard derivations in the number of forwardings at each node, which are less than half of SimpleUtility and MinCost, and are far smaller than MinDelay; (3) the forwarding cost of TOUR and TOUR-OPT is around one-third the cost of SimpleUtility, and is much



(a) (b) (c)

Fig. 8. Remaining utilities, standard derivations of node forwardings, and forwarding costs of TOUR in the Cambridge Haggle Imote Infocom trace.

smaller than MinCost and MinDelay; (4) TOUR is able to approximate TOUR-OPT with unobservable errors in terms of all evaluation metrics.

Figures 8(a)-8(c) show the following facts about TOUR. (1) The remaining utility increases as the initial benefit increases and as the benefit decay coefficient decreases, which shows that important messages and less urgent messages are more likely to be delivered. (2) The standard derivation of node forwardings increases as the initial benefit increases, and TOUR is able to keep the derivations low after the initial benefit reaches a certain value, which is about 50 in this evaluation. When the initial benefit is high, the standard derivation increases as the benefit decay increases, while when the benefit is low, the trend reverses. This is because, as the initial benefit decreases, most nodes avoid other nodes with high forwarding costs. (3) The forwarding cost increases as the benefit decay increases, since a shorter lifetime encourages a message to take a node with a higher forwarding cost. These results suggest that, in TOUR, the messages with a high initial benefit and a large decay coefficient will be delivered with high probabilities of success via the nodes that have high forwarding costs and high contact probabilities to others.

VI. Related work

Many DTN routing algorithms have been proposed, including flooding-based algorithms (e.g., [8], [9]), probability-based algorithms (e.g., [4], [10], [11]), and social-aware algorithms (e.g., [12]–[15]). Compared to the proposed TOUR algorithm, all of these algorithms do not consider the utilities of message forwarding, which will result in most messages being delivered via the nodes with high probabilities of contact with others. The resources of those nodes might be occupied by unimportant message deliveries, making them unable to serve more important delivery requests.

The concept of utility-based routing is first proposed in ad hoc networks [1]. The motivation lies in finding the trade-off between the reliability and the cost of the message delivery, so let the more important messages be delivered through the more reliable paths at the expense of higher transmission costs. This utility model is also extended to opportunistic transmission in [2]. Unlike the previous utility-based model and algorithms, the most important factors in DTN routing design, including delay, and opportunistic forwarding, are considered in our scheme.

VII. Conclusion

In this paper, we first introduce the concept of utility-based routing in DTNs, and we propose a time-sensitive utility model for DTNs, which takes benefit, delay, and cost into account. Under this model, we propose a time-sensitive opportunistic utility-based routing algorithm, i.e., TOUR, which can achieve the maximum expected utility for each message delivery. This algorithm creates a good balance between benefit, delay, and cost, which inherently allows important messages to be delivered along paths with a high probability of success, but at a large cost, much like the postal service in the real world. Simulations with real DTN traces prove the significant performance of the TOUR algorithm.

Acknowledgment

This research was supported in part by the National Grand Fundamental Research 973 Program of China (Grant No.2011CB302905); the National Science and Technology Major Project (Grant No. 2011ZX03005-004-04, 2012ZX03005009); the NSF of China (Grant No. 60803009, 61003044), the NSF of Jiangsu Province in China (Grant No. BK2009150); the Fundamental Research Funds for the Central Universities; and NSF grants ECCS 1231461, ECCS 1128209, CNS 1138963, CNS 1065444, and CCF 1028167.

References

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

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

[3] H. Cai and D. Y. Eun, "Crossing over the bounded domain: From exponential to power-law inter-meeting time in manet," in *ACM MobiCom*, 2007.

[4] V. Conan, J. Leguay, and T. Friedman, "Fixed point opportunistic routing in delay tolerant networks," *IEEE Journal on Selected Areas in Communications*, vol. 26, no. 5, pp. 773–782, 2008.

[5] W. Gao, Q. Li, B. Zhao, and G. Cao, "Multicasting in delay tolerant networks: A social network perspective," in *ACM MobiHoc*, 2009.

[6] James Scott et al., "CRAWDAD data set cambridge/ haggle (v. 2009-05-29)," Downloaded from http://crawdad. cs.dartmouth. edu/cambridge/haggle, May 2009.

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

[8] A. Vahdate and D. Becker, "Epidemic routing for partially-connected ad hoc networks," Duke University, Tech. Rep. CS-2000-06, June 2000.

[9] T. Spyropoulos, K. Psounis, and C. Raghavendra, "Efficient routing in intermittently connected mobile networks: The multiple-copy case," *IEEE/ACM Transactions on Networking*, vol. 16, no. 1, pp. 77–90, 2008.

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

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

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

[13] E. Daly and M. Haahr, "Social network analysis for routing in disconnected delay-tolerant manets," in *ACM MobiHoc*, 2007.

[14] J. Wu and Y. Wang, "Social feature-based multi-path routing in delay tolerant networks," in *IEEE INFOCOM*, 2012.

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

Appendix

A. Proof of Theorem 1

Before the proof, we first prove a lemma:

Lemma 4: For $\forall k \in N_i$, $R(\mu) \subseteq N_i$, and $R^+(\mu) = R(\mu) + \{k\}$, we have:

$$D_i(\mu)|_{R^+(u)} < D_i(\mu)|_{R(u)} \Leftrightarrow D_k(\mu - c_i) + c_i < D_i|_{R(\mu)=\emptyset}.$$

Proof: According to Eq. 4, we have:

$$D_i(\mu)|_{R(u)} = \int_0^{\mu} \sum_{j \in R(u)} \rho_{i,j}(u)(\mu - u_j + D_j(u_j))du + p_f(\mu)\mu,$$

$$D_i(\mu)|_{R^+(u)} = \int_0^{\mu} \sum_{j \in R^+(u)} \rho^+_{i,j}(u)(\mu - u_j + D_j(u_j))du + p^+_f(\mu)\mu.$$

Here, $\rho_{i,j}(u)$ and $\rho^+_{i,j}(u)$ (resp. $p_f(\mu)$ and $p^+_f(\mu)$) represent the (resp. failed) forwarding probabilities under the forwarding sets $R(u)$ and $R^+(u)$, respectively. Note that $\mu - u_j = c_i$, $\rho^+_{i,j}(u) = \rho_{i,j}(u)(1 - \rho_{i,k}(\mu)du)$ for each u ($0 \le u < \mu$), and $p^+_f(\mu) = p_f(\mu)(1 - \rho_{i,k}(\mu)du)$, where $\rho_{i,k}(\mu)du$ is the probability that node i encounters node j during the time interval when $u \in [\mu - du, \mu]$. Then, by comparing $D_i(\mu)|_{R(u)}$ and $D_i(\mu)|_{R^+(u)}$, we can get:

$$\begin{aligned} D_i(\mu)|_{R^+(u)} - D_i(\mu)|_{R(u)} = \\ (D_k(\mu - c_i) + c_i - D_i(\mu)|_{R(\mu)=\emptyset})\rho_{i,k}(\mu)du \end{aligned} \quad (11)$$

This means that the lemma is correct. Note that $D_i(\mu)|_{R(\mu)=\emptyset}$ in Eq. 11 is the utility difference for node i that does not forward the message when $u = \mu$. Its value is independent of $R(\mu)$. This shows that adding a node k into an arbitrary forwarding set can achieve a better utility difference than before adding the node to the set, if and only if node k can achieve a better utility difference than node i when $u < \mu$. ■

Now, we can prove this theorem as follows based on the above result.

1) Without loss of generality, assume node k to be the node in $R^*_i(\mu)$ that has the maximum utility difference with regards to $\mu - c_i$. Then, according to Lemma 4, we have $D_k(\mu - c_i) + c_i < D_i(\mu)|_{R_i(\mu)=\emptyset}$. On the other hand, $D_h(\mu - c_i) < D_k(\mu - c_i)$ for an arbitrary $h \in [1, k)$. Then, $D_h(\mu - c_i) + c_i < D_i(\mu)|_{R_i(\mu)=\emptyset}$. Using Lemma 4 again, we can get a smaller $D_i(\mu)$ when we add node h into $R^*_i(\mu)$ if $h \notin R^*_i(\mu)$. This is a contradiction to the optimality of $R^*_i(\mu)$. Thus, we can get $h \in R^*_i(\mu)$. Since h is a arbitrary node in $[1, k)$, we have $R^*_i(\mu) = \{1, 2, \cdots, k\}$.

2) Since $h + 1 \in R^*_i(\mu)$, we have $D_{h+1}(\mu - c_i) + c_i < D_i(\mu)|_{R_i(\mu)=\emptyset}$ according to part one. Using Lemma 4 again, we have $D_i(\mu)|_{R_i(\mu)=\{1,\cdots,h+1\}} < D_i(\mu)|_{R_i(\mu)=\{1,\cdots,h\}}$. Thus, the theorem is correct.

B. Proof of Theorem 2

First, according to Lemma 4, we have that a node j belongs to the optimal forwarding set of another node i, if and only if $D_j(\mu - c_i) + c_i < D_i(\mu)|_{R_i(\mu)=\emptyset}$. Thus, there will be no loop in the iterative computation. Moreover, for a given utility μ, there must be at least one node that can successfully get its optimal forwarding set at each round of iterative computation.

In the first round, each pairwise neighboring node compares their current utility difference values. There must exist a minimum. Without loss of generality, let this node be node 1. Then, $R_1(\mu) = \{d\}$.

After the first round, each neighbor will update their local records about node 1. Then, in the second round, the node with the second minimum value of utility difference will determine its optimal forwarding set. Let this node be node 2. Just like node 1, no other nodes except 1 and/or d can be taken as the forwarding node of node 2. Then, node 2 can determine its forwarding set $R_2(\mu)$.

Similarly, the third node will determine its optimal forwarding set in the third round, and so on. Therefore, after at most $|V| - 1$ rounds, the whole iterative computation process completes, and all nodes obtain their optimal forwarding sets.

C. Proof of Theorem 3

For a utility $u \in (\mu_{k-1}, \mu_k]$, we first compare $R(u)$ and $\tilde{R}(u)$ in Eq. 7. Fig. 6(b) shows the function distributions of $R(u)$ and $\tilde{R}(u)$, in which the shadow illustrates an estimation error between them. According to Eq. 6, we have:

$$(R(u) - \tilde{R}(u)) \subseteq (R(\mu_k) - R(\mu_{k-1})). \quad (12)$$

Moreover, according to Eqs. 9 and 8, we can further derive the estimation errors of $\tilde{\rho}_{i,j}(u)$ and $\tilde{p}_f(u)$ from Eq. 12:

$$\begin{aligned} \frac{\tilde{\rho}_{i,j}(u)}{\rho_{i,j}(u)}, \frac{\tilde{p}_f(u)}{p_f(u)} &\le e^{\sum_{h=1}^{k} \sum_{j \in (R(\mu_h) - R(\mu_{h-1}))} \lambda_{i,j} \Delta\mu/\delta} \\ &= e^{\sum_{j \in (R(\mu_k) - R(\mu_0))} \lambda_{i,j} \Delta\mu/\delta} \end{aligned} \quad (13)$$

Now, according to Eqs. 4 and 7, we can get the approximate ratio of $D_i(u)$ when we only consider the one-hop estimation error by ignoring the estimation error of $D_j(u)$:

$$\frac{\tilde{D}_i(u)}{D_i(u)} \le e^{\sum_{j \in (R(\mu_k) - R(\mu_0))} \lambda_{i,j} \Delta\mu/\delta} = e^{\sum_{j \in R(\mu_k)} \lambda_{i,j} \Delta\mu/\delta}. \quad (14)$$

Since $\Delta\mu = \frac{b}{n}$, $b \ge max\{D_i(u)|i \in V\}$, $\Lambda = max\{\sum_{j \in N_i} \lambda_{i,j} | i \in V\}$. Then, we can get the one-hop estimation error:

$$|\tilde{D}_i(u) - D_i(u)|_{one-hop} \le b(e^{\frac{\Lambda b}{n\delta}} - 1). \quad (15)$$

According to Eqs. 4 and 7, the total estimation error is no more than the sum of the estimation error in each hop. Thus, we have the total estimation error:

$$|\tilde{D}_i(u) - D_i(u)| \le |V| b(e^{\frac{\Lambda b}{n\delta}} - 1). \quad (16)$$