

Research Article

A General and Specific Utility-Based Adaptive Routing for Delay Tolerant Networks

Lei You, Jianbo Li, Changjiang Wei, Chenqu Dai, and Jixing Xu

Information Engineering College, Qingdao University, Ningxia Road 308, Qingdao, Shandong 266071, China

Correspondence should be addressed to Jianbo Li; lijianboqdu@gmail.com

Received 2 August 2013; Revised 22 November 2013; Accepted 5 December 2013; Published 20 January 2014

Academic Editor: Ai Chen

Copyright © 2014 Lei You et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The nonexistence of an end-to-end path poses great challenges in directly adapting the traditional routing algorithms for ad hoc or mobile ad hoc networks (MANETs) to delay tolerant networks (DTNs). In this paper, an adaptive routing protocol that makes use of both self-aware and destination-aware utility metrics to evaluate the potential throughput between every pair of nodes is proposed. We theoretically analyze the nodes' spraying process and discover the relationship between the initial number of copies and the latency of the message distribution process, by which we can further control the expected latency under a certain level. Compared with Epidemic, Spray And Wait Spray, And Focus and Encounter-based routing, our proposed algorithm is the only one that achieves comparable performance under both Helsinki City Scenario and Cambridge-iMote Scenario.

1. Introduction

Delay/disruption tolerant networks are networks of self-organizing wireless nodes, where end-to-end connectivity is intermittent [1]. The communication model of the Internet is based on some inherent networking assumptions, for example, the existence of a continuous end-to-end path between two nodes, the relatively short round-trip delays, the symmetric data rates, and the low error rates [2]. However, in DTNs these assumptions usually fail, which leads to the fact that the TCP/IP protocol does not work [3]. Hence, many application protocols designed for the Internet architecture cannot operate well in the DTN scenarios such as InterPlanetary interNet (IPN) [4]. In [5], Khabbaz et al. pointed out that we need a new kind of network model when modeling existing and recently emerging wireless networks, especially those deployed in extreme environments (e.g., battlefields, volcanic regions, deep oceans, deep space, developing regions, etc.) where they suffer challenging conditions (e.g., military wars and conflicts, terrorist attacks, earthquakes, volcanic eruptions, floods, storms, hurricanes, severe electromagnetic interferences, congested usage, etc.) resulting in excessive delays, severe bandwidth restrictions, remarkable node mobility, frequent power outages, and

recurring communication obstructions. Based on the observation that quite a few terrestrial networks exhibit delay-tolerant properties, albeit different nature: from sparse mobile ad hoc to sensor networks to mobile Internet access, it is found that delay tolerance exists as an important element to describe communication behavior and to design protocols suitable for operation in the corresponding challenged networking environment [6].

Most research achievements on DTNs focus on the design of routing protocols. Since there are some common characteristics between most terrestrial DTNs and mobile ad hoc networks (MANETs), for example, nodes mobility and "store-carry-forward" paradigm, many researches work on routing in DTNs aiming to solve the newly arisen difficulties in MANETs that address the "delay-tolerant" property. Epidemic protocol [7] makes use of naive replication strategy letting each node replicate the message to all encountered nodes, so as to try to achieve the maximum message delivery ratio. However, the buffer size and energy of nodes are limited thus constraining its practical performance. Spray And Wait [8] takes the cost into consideration and confined the maximum number of copies and hop counts of each message. On the foundations of these two classic routing algorithms, many multicopy routing schemes focusing on evaluating contacts

opportunities among nodes have been proposed [9–13]. All these research achievements are devoted to efficient utilizing of the possible contacts between pair of nodes, thus making messages delivered through multiple hops between nodes in a “store-carry-forward” paradigm. All of these algorithms essentially utilize various utility metrics to replicate or to forward the message in a gradient way to the destination. In DTNs, the utility metrics can be categorized as follows:

- (i) *destination-aware utility*, which evaluates how well-suited a device is for routing towards a specific destination device,
- (ii) *self-aware utility*, which evaluates the applicability of a device as forwarder in general, irrespective of the message’s destination,
- (iii) *unaware utility*, which is a per-message utility of (controlled) flooding, irrespective of any node.

However, as far as our information goes, only [14] takes both self-aware and destination-aware factors into consideration. We denote self-aware utility as the “*general utility*” and destination-aware utility as the “*specific utility*”, respectively. Starting from this point, we design a routing protocol capturing both generic and specific utilities (GSUs) of nodes. Based on this principle, we implement an asymmetric message replicas distribution approach “GSU-Spray” to boost the messages dissemination towards their destinations. The primary motivation is to replicate more copies to candidate nodes as quickly as possible. Besides, since network resources in DTNs are highly constrained, we take the buffer resource issues into consideration. Furthermore, the number of replicas for each message is adjusted between nodes in each transmission by referring to the dynamically updated utility metrics. From this perspective, GSU-Spray has the ability to adaptively cater to the various DTN environments. Our contributions are listed as follows.

- (i) *General and Specific Utility Metrics for Nodes and Messages*. We propose general and specific utilities that take the node buffer issue into consideration. On one hand, we utilize each node’s contact frequency to all other nodes over a time period to design the general utility. On the other hand, for which a longer connection duration between two nodes usually indicates a higher message transmission throughput, we use both contact frequency and contact duration time between each pair of nodes to define the specific utility which evaluates each node’s fitness of being a relay node to the destination node.
- (ii) *Utility-Based Adaptive Asymmetrical Spray Approach*. We design a utility-based spray routing algorithm. We asymmetrically spray the copies of each message by referring to the utility metric. This method dynamically reallocates the network resource for each message, thus adaptively balancing the number of copies between each pair of nodes according to the utility metric.
- (iii) *Combining Specific Utility and Delegation Scheme to Implement Forwarding*. In delegation forwarding

(DF) [15], each message updates a threshold value with the maximum value of all encountered node(s)’ utility metrics for the destination. Different from the traditional controlled-flooding strategies, DF only replicates the message if the encountered node has a better utility value than the threshold value cached in this message. This approach is employed in our routing algorithm by letting each message record the maximum specific utility value encountered before, thus lowering the total costs of routing.

- (iv) *Adjustable Sensitivity of the Contact Duration Time*. Rather than simply accumulating every contact duration time for each node, we design a function that integrates an exponential function $e^{-\lambda t}$ in each duration time interval. Consequently, the sensitivity to the contact duration time of the utility metric is controlled by the value of the parameter λ .
- (v) *Using Markov Chain to Guarantee a Desirable Latency*. The distribution process for a certain message can be modeled as a discrete Markov Chain, where each state is denoted by the number of message replicas. We theoretically analyze the replication process of spraying stage in GSU-Spray by building a Markov Chain. Then we discover the relationship between the initial number of copies of a message and the desired latency. By dynamically adjusting the preassigned number of copies of each message, we can keep the expected latency under a certain level.

The paper is organized as follows. Section 2 introduces the network model and describes the routing problem. In Section 3 we put forward details of routing design. Section 4 analyses the spraying process by using a Markov Chain Model. In Section 5 we show the simulation results. In Section 6 we report on previous works in the field of DTNs. Section 7 eventually concludes this paper.

2. Preliminary

The notations relative to our network model are listed in Table 1. To illustrate the general picture of our routing in the network, we use the following model. As stated in the Introduction, we focus on mobile opportunistic networking scenarios where nodes communicate using the DTN bundle protocol [16]. All nodes are willing to cooperate and to assist other nodes in forwarding bundles to their destinations. Moreover, we assume that node n_s holds a variable $L_s(k)$ for each message m_k to record the number of remaining “tickets.” Such ticket denotes how many copies of m_k can be distributed to other nodes by n_s in the spraying stage. Our primary goal is to efficiently allocate the number of tickets between two meeting nodes to accelerate the message distribution process. In the source Spray And Wait routing [8], the source node is the only node to spray the messages to others, while in its binary edition all nodes equally spray the message copies.

As illustrated in Figure 1, the source Spray And Wait routing shows up a little stiffly, since it only makes use of the contacts between the source node and others to distribute

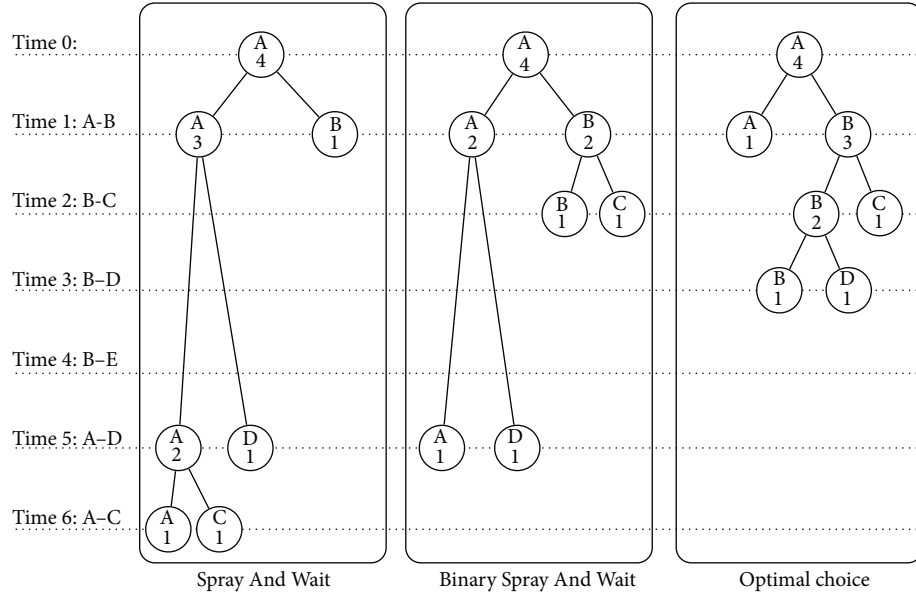


FIGURE 1: Message duplication tree formed by spray operations.

message copies. Binary Spray And Wait is not the optimal choice either, since the source node always equally sprays the message copies with unawareness of each node's capability. Though [8] proves that binary Spray And Wait is optimal when node movement is independent identically distributed (IID), in most cases we do not have such a strict IID movement that the binary distribution method fails to get the optimal choice, as shown in Figure 1.

In this paper, we propose an adaptive routing scheme to satisfy the various DTNs. Based on the Spray And Wait routing, we adaptively adjust the number of tickets between each pair of nodes during the whole spraying process, so as to efficiently allocate network resources. In addition, since spray-based routing algorithms can only generate limited number of message copies, in forwarding stage, we employ delegation forwarding scheme to further replicate messages to other nodes in order to keep sufficient copies, thus achieving the purpose of improving the delivery performance.

3. Routing Details

In DTN routing, the utility of a node is a measure of its contribution to improve a routing metric such as throughput or delay [17]. Let us briefly review the objectives of DTN routing algorithms. Firstly we should maximize the message delivery ratio. Then the average latency and the overhead ratio should be controlled under an acceptable level. There are two stages in our proposed routing. In the spraying stage, GSU-Spray sprays message copies to others. Different from traditional spray-based routing, our scheme adaptively adjusts the number of message copies between each pair of contacting nodes. For each message, when there is only one copy held by a node, the node then enters the delegation forwarding stage where nodes only replicate the message to other nodes with a higher utility metric value than

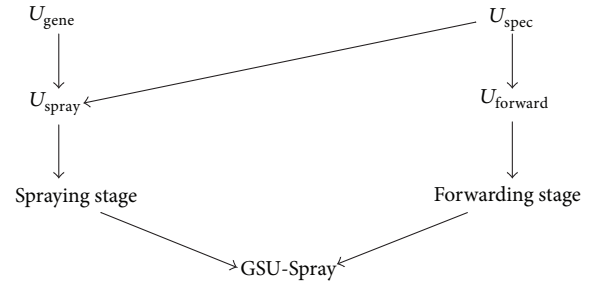


FIGURE 2: The relationship of utility metrics dependency.

the delegation threshold. In this stage, when a message is successfully relayed to n_b by n_a , the delegation threshold in each message copy will be updated to the utility of n_b .

In this section, we firstly give the definition of both general metric and specific metric. We then describe how to adaptively spray the message copies to nodes by using both general and specific utility metrics. Finally we combine the specific utility metric with the node state variable γ to implement delegation forwarding. Either stage takes the node buffer space issues into consideration. The relationship of utility metrics dependency is illustrated in Figure 2. U_{gene} and U_{spec} represent the general and specific utilities, respectively, and both of them are the atomic utilities, which are updated based on the history records of contact duration and frequency. U_{spray} is the metric to evaluate a node's capability of spraying message copies, which is composed by both U_{gene} and U_{spec} . The other metric U_{forward} is used in the delegation forwarding stage, by only resorting to the help of U_{spec} . Then based on these two stages comes our GSU-Spray algorithm.

3.1. Utility-Based Asymmetrical Spraying Stage. One way to increase the performance of routing is to spray message

TABLE 1: Mathematical notations for network model.

Notation	Meaning
n_s	The node with the EID s
m_k	The message identified by k
$\gamma(s)$	Current resource state of n_s ($\gamma(s) \in [0, 1]$)
$\phi(s)$	Total meeting frequency of n_s
$L_s(k)$	The number of tickets of message m_k for current node n_s
$\tau(k)$	The delegation threshold for message m_k
$TS_{s,d}(k)$	Start time of the k th connecting between n_s and n_d
$TD_{s,d}(k)$	End time of the k th connecting between n_s and n_d
$U_{\text{gene}}(s)$	The general utility for n_s
$U_{\text{spec}}(s, d)$	The specific utility for n_s and n_d
$U_{\text{spray}}(s, d)$	Utility for allocating replicas between n_s and n_d in spraying state
$U_{\text{forward}}(s, d)$	Utility for replicating message from n_s to n_d in forwarding stage

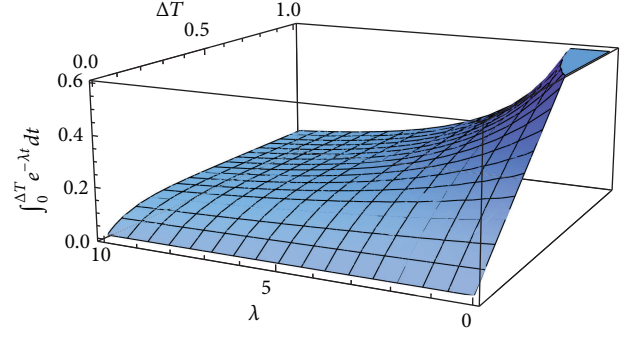
copies to other nodes in the network, so as to raise the parallelism of the message [18]. The first stage of GSU-Spray is to asymmetrically adjust message copies between every pair of nodes. As addressed before that both general and specific utilities should be taken into consideration, we calculate the generic utility and specific utility for node n_s by

$$U_{\text{gene}}(s) = \gamma(s) \phi(s), \quad (\gamma(s) \in [0, 1]), \quad (1)$$

$$U_{\text{spec}}(s, d) = \sum_{i=1}^{i=n} \int_{TS_{s,d}(i)}^{TD_{s,d}(i)} e^{-\lambda t} dt, \quad (\lambda > 0). \quad (2)$$

Equation (1) evaluates a node's capability of being the relay node in a "self-aware" perspective, of which $\phi(s)$ records the contact frequency between n_s and all other nodes and $\gamma(s)$ reflects the available buffer resource in n_s with the range interval $[0, 1]$.

Equation (2) is used to calculate the specific utility for each pair of nodes in a "destination-aware" perspective. Different from general utility that reflects the state of a node itself, a specific utility metric measures the qualification of prospective forwarding throughput between a pair of nodes. In (2), $TS_{s,d}(i)$ and $TD_{s,d}(i)$ represent the start time and the end time of i th contact between n_s and n_d , respectively. In addition, we design a function that integrates an exponential function $e^{-\lambda t}$ in each duration time interval, by which the sensitivity of the utility to the contact duration time can be adjusted. As illustrated in Figure 3, a larger λ value leads to a curve decreasing more quickly, which leads to the fact that the utility is less sensitive to the contact duration time. This is because the increment of the contact duration time interval ΔT has little influence on the value of $\int_0^{\Delta T} e^{-\lambda t} dt$, thus making our evaluation criteria degenerate near to the metric relying only on nodes' contact frequency. Nevertheless, the smaller λ value we set, the larger increment shows on the curve of the calculus, which means that the evaluation criteria (2) becomes more sensitive to the nodes' contact duration time.

FIGURE 3: The relationship between $\int_0^{\Delta T} e^{-\lambda t} dt$, λ , and ΔT .TABLE 2: Effects of λ values on node(s)'s sensitivity to the contact duration time.

λ	ΔT	$\int_0^{\Delta T} e^{-\lambda t} dt$	Ratio ¹
$\lambda = 10.0$	0.20	0.0865	
	0.40	0.0982	1.1353
	0.60	0.0997	1.0153
	0.80	0.1000	1.0030
	1.00	0.1000	1.0000
$\lambda = 1.0$	0.20	0.1813	
	0.40	0.3297	1.8185
	0.60	0.4512	1.3685
	0.80	0.5507	1.2205
	1.00	0.6321	1.1478

¹The ratio between $\int_0^{\Delta T} e^{-\lambda t} dt$ in the current line and that in the above line.

Table 2 shows the exact function values we obtain by setting different λ and ΔT . Though the function gets different values at different λ , the rank stays the same. We can see from the fourth column that the ratio increase is different between the two cases of λ , and it is obvious that compared with the function with a larger λ value the contact duration time ΔT has a stronger influence on that than with a smaller λ value.

The details of the utility-based spraying stage are depicted in Algorithm 1, where our primary goal is to spray message copies to the network as efficiently as possible. We achieve this target by adaptively allocating or reallocating the number of copies for each message when a pair of nodes get into connection. Algorithm 1 takes the information of each pair of nodes (in our case, n_a and n_b) as input and then outputs the adjusted results, respectively represented as $L_{a_{\text{new}}}$ and $L_{b_{\text{new}}}$. The two metrics U_{gene} and U_{spec} are calculated according to (1) and (2) separately. Since the general metric U_{gene} is only relative to the state of current node itself, we only need to calculate it once per algorithm running time. Both n_a and n_b compute their own U_{gene} and then exchange them with each other. Then as shown in lines 2-3, the algorithm obtains g_a and g_b that can reflect the prospective throughput between n_a and n_b . Both g_a and g_b are less than 1 and we always maintain that $g_a + g_b = 1$. When $L_{a_{\text{new}}}(k) = 0$, it is undesirable for n_a to continue holding message m_k , and thus n_a deletes the copy of m_k from its buffer. The same rule also works on $L_{b_{\text{new}}}$.

Input: node information of n_a, n_b
Output: $L_{a_{\text{new}}}, L_{b_{\text{new}}}$
 when n_a get in contact with n_b
 (1) local variables declaration: g_a, g_b, s_a, s_b, d
 (2) $g_a \leftarrow \frac{U_{\text{gene}}(a)}{U_{\text{gene}}(a) + U_{\text{gene}}(b)}$
 (3) $g_b \leftarrow 1 - g_a$
 (4) **for** $m_k \in n_a.\text{messageCollection}()$ **do**
 (5) $d \leftarrow \text{destination node of } m_k$
 (6) $s_a \leftarrow \frac{U_{\text{spec}}(a, d)}{U_{\text{spec}}(a, d) + U_{\text{spec}}(b, d)}$
 (7) $s_b \leftarrow 1 - s_a$
 (8) $U_{\text{spray}}(a, d) \leftarrow \alpha g_a + (1 - \alpha) s_a$
 (9) $U_{\text{spray}}(b, d) \leftarrow \alpha g_b + (1 - \alpha) s_b$
 (10) $L_{a_{\text{new}}}(k) \leftarrow (L_{a_{\text{old}}}(k) + L_{b_{\text{old}}}(k)) \left[\frac{U_{\text{spray}}(a, d)}{U_{\text{spray}}(a, d) + U_{\text{spray}}(b, d)} \right]$
 (11) $L_{b_{\text{new}}}(k) \leftarrow L_{a_{\text{old}}}(k) + L_{b_{\text{old}}}(k) - L_{a_{\text{new}}}(k)$
 (12) **end for**
 (13) **return** $(L_{a_{\text{new}}}, L_{b_{\text{new}}})$;

ALGORITHM 1: Utility-based spraying algorithm.

and n_b . It should be noted that though the number of replicas decreases in the above case, the number of total prospective spraying copies stays the same, since the responsibility of n_b to spray the message m_k shifts to n_a .

Different from the generic utility metric that is only relative to one node, the specific utility metric U_{spec} reflects the prospective throughput between one pair of nodes. For the purpose of sending the message to a node “nearer” to its destination, U_{gene} and U_{spec} should be combined together. We compute the metric U_{spec} for all the messages and also maintain two variables s_a and s_b which reflect the prospective throughput of n_a and n_b to m_k 's destination n_d . Both s_a and s_b are less than 1 and we also maintain that $S(a, d) + S(b, d) = 1$.

3.2. Utility-Based Forwarding Stage. After the spraying operation, the node enters the forwarding stage, at which the primary goal is to effectively deliver messages to their respective destinations. At this stage, the task of each node shifts from reasonably distributing replicas to quickly seeking for the destination node. The corresponding utility is calculated by

$$U_{\text{forward}}(s, d) = \gamma(s) U_{\text{spec}}(s, d). \quad (3)$$

As shown in (3), we only take U_{spec} into account when considering the capability of forwarding. The reason behind this is that, different from the spraying stage that tends to spray copies as efficiently as possible, at the forwarding stage we desire each message copy to be relayed along a way similar to the shortest path to the destination. However, since it is difficult to mathematically get the strict shortest path, we employ the delegation forwarding strategy [15] to make each message replicated in a gradient direction to its destination, while introducing acceptable network costs. Besides, we still value the effect of buffer space by using the coefficient $\gamma(s)$ in (3), in the same way as (1).

Delegation forwarding (DF) [15] caught significant attention in the research community because of its simplicity and impressive performance. The main idea of DF is to replicate the copies of a packet only to the individual that has a higher utility value to the destination. With the increase of the utility level, a message holder's forwarding opportunity is expected to be decreased, which means that the number of copies duplicated for a message and its total number of forwarding operations are expected to be reduced, thus reducing the network cost. Reference [15] analyses that when the total number of nodes is N , delegation forwarding has the cost of $O(\sqrt{N})$, while the naive method of forwarding achieves the cost of $O(N)$.

As mentioned above, since delegation forwarding can efficiently reduce the routing costs, we devise the GSU-Spray forwarding strategy by referring the delegation forwarding strategy. Algorithm 2 shows the detailed information of the utility-based delegation forwarding process. The input of the algorithm is the information of n_a and n_b that includes their message collection and the respective utility metrics. The output is a list X with n elements that indicates the expected relay choice of all the n messages. When the algorithms are finished on the two nodes, the messages will be exchanged according to their corresponding list X . To further explain Algorithm 2, we assume that the algorithm is running on n_a . Line 1 initializes the array X with each element to be set false, which indicates that none of the messages would be replicated to n_b . In lines 2–9, x_k is calculated for each message m_k , according to (3). The highest encountered utility value for each message is recorded by τ_k , which is set to be the larger value of $U_{\text{forward}}(a, d)$ and $U_{\text{forward}}(b, d)$. Finally, the algorithm returns the array X , which includes the expected replication choice for every message.

```

Input:  $n_a, n_b$ 
Output:  $X = [x_1, x_2, \dots, x_n]$  ( $\forall i, x_i \in \{\text{true}, \text{false}\}$ )
  when  $n_a$  gets in contact with  $n_b$ 
  (1) local variables declaration:  $d$ 
  (2)  $X \leftarrow [\text{false}, \text{false}, \dots, \text{false}]$ 
  (3) for  $m_k \in N_a.\text{messageCollection}()$  do
  (4)    $d \leftarrow \text{destination node of } m_k$ 
  (5)    $U_{\text{forward}}(a, d) \leftarrow \gamma(a) U_{\text{spec}}(a, d)$ 
  (6)    $U_{\text{forward}}(b, d) \leftarrow \gamma(b) U_{\text{spec}}(b, d)$ 
  (7)   if  $U_{\text{forward}}(b, d) > \tau_k$  then
  (8)      $x_k \leftarrow \text{true}$ 
  (9)   end if
  (10)   $\tau_k \leftarrow \max \{U_{\text{forward}}(a, d), U_{\text{forward}}(b, d)\}$ 
  (11) end for
  (12) return  $X$ ;

```

ALGORITHM 2: Utility-based delegation forwarding algorithm.

TABLE 3: Mathematical notations for Markov Chain analysis.

Notation	Meaning
T	The mean time interval of all connections
M	Total number of different generated messages
N	Total number of nodes in the network
B	Total buffer size of all nodes in the network
m	The average size of each message replica
c	The initial number of copies for each message
$\pi(n)$	The state that there are n replicas
END	The state that a replica meets the destination

4. Analysis of Message Spraying Process

In this section, a Markov Chain Model is built to analyze the message spraying process of GSU-Spray. All the notations in this section are listed in Table 3. We assume that the mean contact time interval between any pair of nodes is T . The total time of spraying stage is split into many short time slots, each with the length T . $k \cdot T$ stands for the length of the passed k time intervals. Our primary goal is to get the latency for a message to generate n copies during the spraying process. The Markov Chain is illustrated in Figure 4, where $\pi(n)$ denotes the state that there are n replicas for the message in the whole network, m and c are the average message size and the initial number of maximum copies of each message, respectively, and END denotes the state that a replica meets the destination and the message transmission is completed. All other states have the opportunity to transfer to the state END, which means that at least one of the replicas reaches the destination. There are bidirectional transition opportunities between $\pi(n)$ and $\pi(n-1)$, because GSU-Spray keeps adjusting the number of messages copies between each pair of nodes. As stated in Section 3.1, either n_a or n_b should delete m_k from the buffer if its $L(k)$ value is zero. Finally, there exists a case that after two nodes coming into contact, there is no change in the number of replicas of the message. This happens when

both of the nodes do not hold any replica of the message or both of them hold the replica but both $L_{a_{\text{new}}}$ and $L_{b_{\text{new}}}$ are more than zero, which indicates that there is no need for transmission.

We firstly prove four lemmas, and then from these lemmas we get the transition probability of the Markov Chain. Finally we calculate $T(n)$ that represents the latency before generating n replicas of the message by using the transition probability of the Markov Chain. In our analysis, we give the assumption that $0 < \alpha$ and $\beta < 1$. Besides, we exclude the factors of node energy and define that $\gamma(s) = \text{unengaged buffer/total buffer}$.

Lemma 1. The transition probability from state $\pi(n)$ to state $\pi(n+1)$ is

$$2 \cdot \frac{n}{N} \cdot \frac{N-n-1}{N-1} \cdot p^2 \left(n < \frac{B}{Mm} \right) \cdot \left(\int_0^{T \cdot k} \frac{1}{NT} e^{-x/NT} dx \right)^2. \quad (4)$$

Proof. When there are n nodes with replicas of the message, the probability for some node to hold a replica is n/N . Excluding this node, there are $N-1$ nodes left, of which there are $N-n-1$ nodes (here the “1” means the destination node) that currently do not hold any replica of this message. When a node n_a with a replica meets some node n_b with no replica, the total number of replicas increases if and only if we have both $L_{a_{\text{new}}} \neq 0$ and $L_{b_{\text{new}}} \neq 0$. Otherwise either n_a keeps holding the replica or forwards the replica to n_b and then deletes it from its own buffer. Thus we have the following probability equation:

$$p[\pi(n), \pi(n+1)] = 2 \cdot \frac{n}{N} \cdot \frac{N-n-1}{N-1} \cdot P \{L_{a_{\text{new}}} \neq 0, L_{b_{\text{new}}} \neq 0\}, \quad (5)$$

where we have the following:

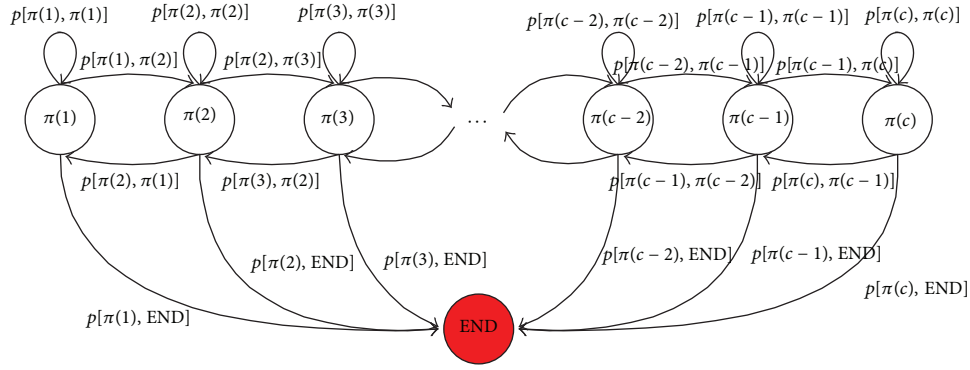


FIGURE 4: A Markov Chain that models a message replication process.

$$\begin{aligned}
P\{L_{a_{\text{new}}} \neq 0, L_{b_{\text{new}}} \neq 0\} &= P\{U_{\text{spray}}(a, d) \neq 0, U_{\text{spray}}(b, d) \neq 0\} \\
&= P\{\alpha g_a + (1 - \alpha) s_a \neq 0, \alpha g_b + (1 - \alpha) s_b \neq 0\} \\
&\quad (0 < \alpha < 1) \\
&= P\{U_{\text{gene}}(a) + U_{\text{spec}}(a, d) \neq 0, U_{\text{gene}}(b) + U_{\text{spec}}(b, d) \neq 0\} \\
&= P\{\gamma(a) \neq 0, \phi(a) \neq 0\} \cdot P\{\gamma(b) \neq 0, \phi(b) \neq 0\} \\
&= P\{\gamma(a) \neq 0\} \cdot P\{\gamma(b) \neq 0\} \cdot P\{\phi(a) \neq 0\} \cdot P\{\phi(b) \neq 0\} \\
&= P^2\{\gamma(a) > 0\} \cdot P^2\{\phi(a) > 0\}.
\end{aligned} \tag{6}$$

The total buffer size of all nodes is represented as B . We denote the number of different kinds of generated messages as M . Then the average buffer size for each kind of message is B/M . The number of current message replicas is n , and we know that the average size of each message replica is m , so the total size of all generated replicas of the current message can be represented as $m \cdot n$. We can conclude that the buffer space for the current message runs out if and only if $(n \cdot m)/(B/M) < 1$. So we have the following:

$$P\{\gamma(a) > 0\} = P\left\{\frac{n \cdot m}{B/M} < 1\right\} = P\left(n < \frac{B}{Mm}\right). \tag{7}$$

Because the mean time interval between any pair of nodes contacting is T , the mean time for each node to meet any another node is NT , and thus we assume that the waiting time for each node to meet another node obeys the exponential distribution with mean value $\lambda = NT$. And consequently we have the following:

$$P\{\phi(a) > 0\} = p(t < Tk) = \int_0^{T \cdot k} \frac{1}{NT} e^{-t/NT} dt, \tag{8}$$

where k stands for the number of passed time slots since the starting time of spraying and $k \cdot T$ is the whole passed time period. Thus we have

$$\begin{aligned}
P\{L_{a_{\text{new}}} \neq 0, L_{b_{\text{new}}} \neq 0\} &= p^2 \left(n < \frac{B}{Mm}\right) \left(\int_0^{T \cdot k} \frac{1}{NT} e^{-t/NT} dt\right)^2, \\
p[\pi(n), \pi(n+1)] &= 2 \cdot \frac{n}{N} \cdot \frac{N-n-1}{N-1} \\
&\quad \cdot p^2 \left(n < \frac{B}{Mm}\right) \cdot \left(\int_0^{T \cdot k} \frac{1}{NT} e^{-x/NT} dx\right)^2.
\end{aligned} \tag{9}$$

□

Lemma 2. The transition probability from state $\pi(n)$ to state $\pi(n-1)$ is

$$\begin{aligned}
2 \cdot \frac{n}{N} \cdot \frac{N-n-1}{N-1} \\
\cdot \left[1 - p^2 \left(n < \frac{B}{Mm}\right) \left(\int_0^{T \cdot k} \frac{1}{T} e^{-x/T} dx\right)^2\right].
\end{aligned} \tag{10}$$

Proof. As stated in the proof process of Lemma 1, when there are two nodes contacting, the probability that one holds a replica and the other does not is $2 \cdot (n/N) \cdot (N-n-1)/(N-1)$. Then the number of replicas decreases if and only if either $L_{a_{\text{new}}} = 0$ or $L_{b_{\text{new}}} = 0$. Since $P\{L_{a_{\text{new}}} = 0 \text{ or } L_{b_{\text{new}}} = 0\} = 1 - P\{L_{a_{\text{new}}} \neq 0 \text{ and } L_{b_{\text{new}}} \neq 0\}$, we have the following:

$$\begin{aligned}
p[\pi(n), \pi(n-1)] &= 2 \cdot \frac{n}{N} \cdot \frac{N-n-1}{N-1} \\
&\quad \cdot \left[1 - p^2 \left(n < \frac{B}{Mm}\right) \left(\int_0^{T \cdot k} \frac{1}{T} e^{-x/T} dx\right)^2\right].
\end{aligned} \tag{11}$$

□

Lemma 3. The transition probability from the state $\pi(n)$ to the state END is

$$2 \cdot \frac{n}{N} \cdot \frac{1}{N-1}. \tag{12}$$

Proof. When two nodes meet, the probability of either node to hold a replica is n/N , and the probability of the other to be the destination node is $1/(N-1)$, so we have

$$p[\pi(n), \text{END}] = 2 \cdot \frac{n}{N} \cdot \frac{1}{N-1}. \quad (13)$$

Lemma 4. *The transition probability for the state $\pi(n)$ to stay the same is*

$$1 - p[\pi(n), \pi(n+1)] - p[\pi(n), \pi(n-1)] - p[\pi(n), \text{END}]. \quad (14)$$

Proof. Since this concludes all other cases, we can directly get it from the normative theory of probability. \square

Theorem 5. *The transition probability in the Markov Chain between two states is*

$$\begin{aligned} p_1 &= p[\pi(n), \pi(n+1)] = 2 \cdot \frac{n}{N} \cdot \frac{N-n-1}{N-1} \\ &\quad \cdot p^2\left(n < \frac{B}{Mm}\right) \cdot \left(\int_0^{T \cdot k} \frac{1}{NT} e^{-x/NT} dx\right)^2, \\ p_2 &= p[\pi(n), \pi(n-1)] = 2 \cdot \frac{n}{N} \cdot \frac{N-n-1}{N-1} \\ &\quad \cdot \left[1 - p^2\left(n < \frac{B}{Mm}\right) \left(\int_0^{T \cdot k} \frac{1}{NT} e^{-x/NT} dx\right)^2\right], \\ p_3 &= p[\pi(n), \text{END}] = 2 \cdot \frac{n}{N} \cdot \frac{1}{N-1}, \\ p_4 &= p[\pi(n), \pi(n)] = 1 - p[\pi(n), \pi(n+1)] \\ &\quad - p[\pi(n), \pi(n-1)] - p[\pi(n), \text{END}]. \end{aligned} \quad (15)$$

Proof. From Lemmas 1–4, we get it immediately. \square

The minimum latency before generating n replica of a certain message is represented as $T(n)$, which is calculated based on Theorem 5. Consider

$$\begin{aligned} T(n) &= p[\pi(n-1), \pi(n)] \cdot [T(n-1) + T] \\ &\quad + p[\pi(n), \pi(n)] \cdot [T(n) + T] \Rightarrow T(n) \\ &= (p[\pi(n-1), \pi(n)] \cdot [T(n-1) + T] \\ &\quad + p[\pi(n), \pi(n)] T) \\ &\quad \times (1 - p[\pi(n), \pi(n)])^{-1}. \end{aligned} \quad (16)$$

By solving this recursive equation, we have the following:

$$T(n) = \left\{ \frac{\{1 - (p_1/(1 - p_4))^{n-1}\} \{p_1 + p_4\}}{1 - p_4 - p_1} \right\} T. \quad (17)$$

If we set the initial number of copies for each message c to be less than the constant B/Mm , then we have $c < B/Mm$,

because the inequality $n \leq c$ always holds in the spraying process. Consequently, we have $p(n < B/Mm) = 1$ in the meaning of the expected value. And thus given the parameters N and T , we can calculate the transition probability in each time slot and adaptively adjust the value c to guarantee the latency $T(n)$ under an acceptable level.

5. Simulation

The simulation is evaluated by the opportunistic network environment (ONE) [10]. In detail, we evaluate the Epidemic, binary Spray And Wait, Spray And Focus, and Encounter-based routing for performance comparison, using both synthetic mobility model and real trace. The simulation is grouped into the following categories: (1) varying buffer size in Helsinki City Model; (2) varying buffer size in Cambridge-iMote real trace; (3) varying message time-to-live in Helsinki City Model; (4) varying message time-to-live in Cambridge-iMote. The four comparison routing algorithms are listed as follows.

- (1) *Epidemic* [7]: in this routing scheme, packets received at intermediate nodes are forwarded to all the nodes neighbors (except the one who sends the packet) without employing any flooding control strategy.
- (2) *Spray and Wait (S & W)* [8]: spray stage: each node with more than one copy forwards half of the copies to the encountered node with no copy; Wait stage: if the destination is not found in the spray stage, the copy carriers wait for the destination. This algorithm only employs an unaware utility that assigns a maximum number of replicas for each message.
- (3) *Spray and Focus (S & F)* [9]: the spray stage of S & F is the same as in binary S & W that symmetrically distributes the message replicas. The difference is that S & F adopts the focus phase instead of wait phase, decreasing the delivery delay via a utility forwarding approach. The employed utility is only destination aware (specific).
- (4) *Encounter-Based Routing (EBR)* [10]: EBR is an asymmetrical spray-based routing protocol, which takes advantage of the observed mobility property of certain network, assuming the future rate of node encounters can be roughly predicted by historical information. The assumption behind it is that nodes experience a large number of encounters that would have a higher potential to relay the message to final destination. The employed utility for distributing the replicas is only self aware (general).

For simplification, in our simulation, the self-aware variable $\gamma(s)$ for node n_s is set to be unengaged buffer/total buffer. For the purpose of fairness, the initial number of copies for the four algorithms S & W, S & F, EBR, and our proposed routing is set to be the same, which is a recommended value between 10% and 15% of the total number of nodes in each scenario. We compare the five different routing protocols based on the following criteria.

- (1) *Delivery Ratio*: normally, the ultimate goal of routing in DTNs is to achieve great delivery performance. This criteria is the measure of delivery capability for each protocol. When the network resource is sufficient, Epidemic routing usually has the best delivery performance. This is because Epidemic routing always finds the best possible path to the destination. Therefore it represents the baseline for the best possible delivery performance.
- (2) *Average Latency*: end-to-end latency is another important concern in DTN routing design. Long average latency means that the message must occupy valuable buffer space for a longer time, and consequently we desire a low latency value.
- (3) *Overhead Ratio*: it is desirable to have a low overhead ratio, since it reflects the efficiency of message transmission. Overhead ratio is defined to be the number of relay operations (excluding the delivery action) over the number of total delivered messages.
- (4) *Average Hop Count*: we expected to minimize the number of hops that a message must take in order to reach the destination.

5.1. Simulation in Helsinki City Scenario

5.1.1. Performance Evaluation by Varying Buffer Size. In Helsinki City Scenario [19], the nodes are assumed to be users with mobile phones or similar devices, using Bluetooth interface at 250 KBps bandwidth and 10 m transmission range. In this case, the initial free buffer size of each node is set to be small, which ranges from 5 M to 55 M. There are six trams following predefined routes, and there is an extra high-speed interface at 10 MBps bandwidth and 1000 m transmission range for the communication between trams. Two thirds of the remaining nodes are pedestrians and one third are cars. The speed of cars is set to be 10–50 km/h and the speed of trams 25–36 km/h, with the pause time of 10–120 s and 10–30 s, respectively. Both pedestrians and cars randomly choose their destinations on the map and move along the shortest path. The parameters settings are listed in Table 4.

Regarding the results in Figures 5(a), 5(b), 5(c), and 5(d), GSU-Spray achieves the highest delivery ratio and the lowest average latency. The overhead ratio of GSU-Spray protocol is almost as low as S & W and S & F and is much lower than that of Epidemic and EBR. However the average hop count is higher than Epidemic and EBR when the buffer size is more than 15 M.

The result in Figure 5(a) shows that GSU-Spray significantly outperforms Epidemic and EBR but has a slightly higher delivery ratio than S & W and S & F. However, in Figure 5(b), it is obvious that the average latency of GSU-Spray is the lowest among all protocols. This is because GSU-Spray adaptively adjusts the number of message copies between each pair of nodes, thus making the message copies have a higher probability to meet the destination. Another reason is that GSU-Spray employs the delegation forwarding method to replicate the message in a gradient way to the destination, thus lowering the average latency of each

TABLE 4: Simulation settings of Helsinki City Scenario.

Parameter name	Range (default value)
Number of nodes	126
World size ($m \times m$)	4500 \times 3000
Initial tickets number	13
Message TTL (min)	180–480 (300)
Simulation time (hours)	12
Message size (KB)	500–1024
Pedestrian buffer (MB)	5–55 (5)
Tram buffer (MB)	500
Bluetooth range (m)	10
Highspeed range (m)	1000
Bluetooth bandwidth (KBps)	250
Highspeed bandwidth (MBps)	10
Pedestrian speed (m/s)	0.5–1.5
Message interval (s)	35–40
GSU-Spray- α	0.7
GSU-Spray- λ	10.0

message. However, from Figure 5(d), we see that the average hop count of GSU-Spray is relatively higher than S & W and is approximately equal to S & F. The reason is that, compared with S & W, our proposed routing employs delegation forwarding instead of waiting in the second stage of routing and this strategy is similar to S & F. When the buffer size is more than 15 M, the average hop count is higher than both Epidemic and EBR. Usually, there is sometimes a trade-off between latency and hop count, and our proposed algorithm tends to give priority to optimize the former criteria.

There are two stages of GSU-Spray protocol. In the spraying stage, the maximum number of replicas for each message is fixed. In the forwarding stage, the delegation forwarding method is employed to control the cost in $O(\sqrt{N})$. Consequently, GSU-Spray generates much fewer copies for each message than Epidemic that employs the flooding strategy. The smaller number of message copies leads to the fewer relay operations, which leads to the greater efficiency per transmission operation. So GSU-Spray has much lower overhead ratio than Epidemic, as illustrated in Figure 5(c). Though the maximal number of copies of EBR is constrained, the overhead ratio of EBR is also very high. Since the utility function in EBR only considers the encounter rate of nodes, the simulation results show that the relay operation of EBR is relatively inefficient. S & W distributes the copies by using the least relay operations (not the least time), and thus it can be regarded as the lowest bound of overhead for GSU-Spray algorithm. We can see from Figure 5(c) that the overhead ratio of GSU-Spray is almost the same as S & W and S & F. However GSU-Spray outperforms S & W and S & F in both delivery ratio and average latency.

5.1.2. Performance Evaluation by Varying Message TTL. In the simulation of varying message time-to-live, we set the node buffer size (only for cars and pedestrians, not for trams) constant at 5 MB. The result in Figure 6(a) shows that

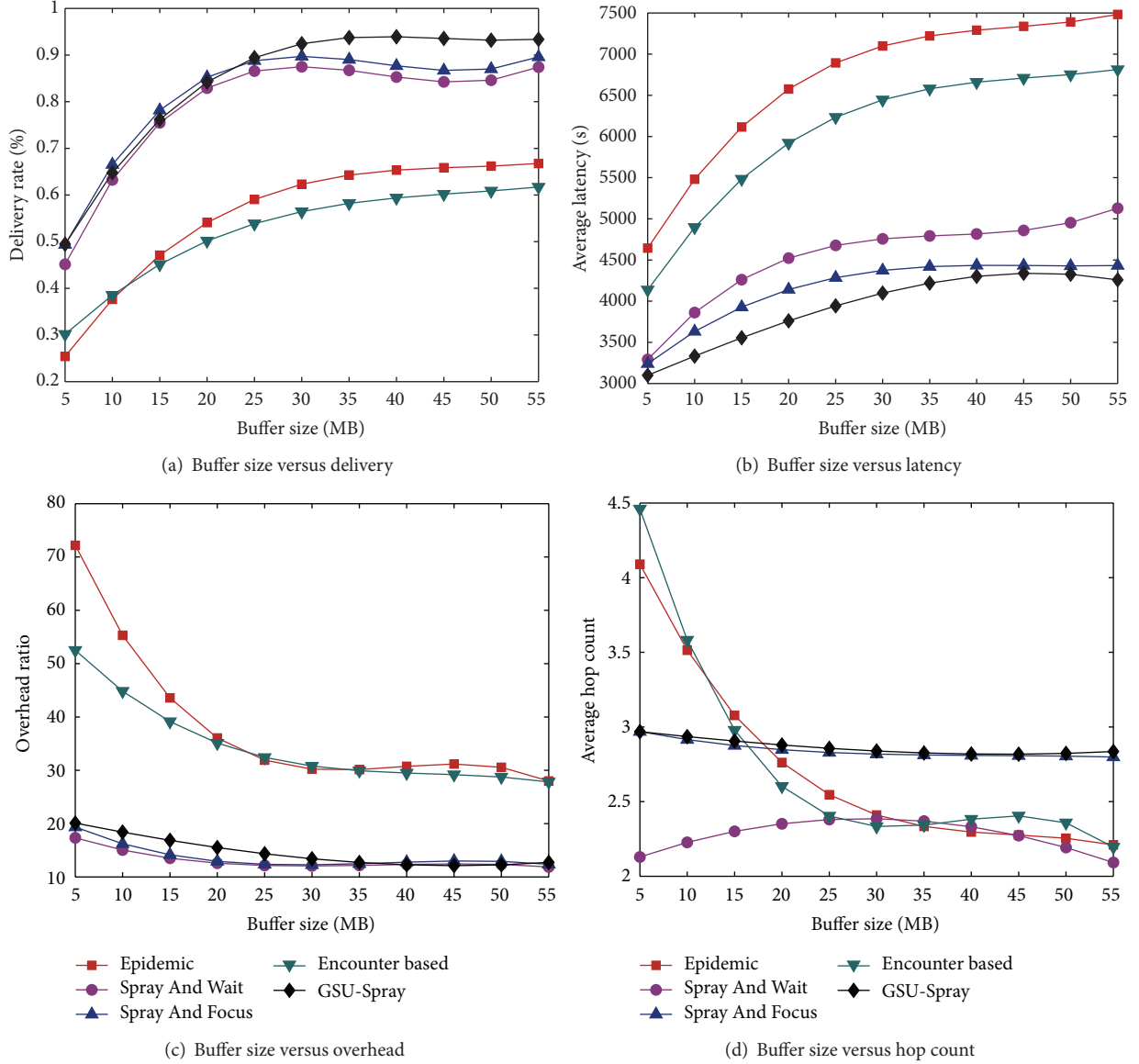


FIGURE 5: Buffer size versus delivery ratio, average latency, overhead ratio, and average hop count (*Helsinki City Scenario*).

GSU-Spray outperforms the other four routing algorithms in message delivery ratio. As same as illustrated in Figure 5(b), the average latency of GSU-Spray also keeps in the lowest level among all protocols in Figure 6(b). In addition, GSU-Spray performs well in both overhead and average hop count, as shown in Figures 6(c) and 6(d).

The result in Figure 6(a) shows that GSU-Spray routing significantly outperforms the other protocols. The performances of S & W and S & F are approximately equal. In the forwarding stage, since we employ delegation strategy to confine the average cost of message copies, the number of generated message copies is much smaller than Epidemic, thus avoiding the buffer resource from running out quickly. As a general utility-based routing, EBR has a relatively lower delivery ratio than the others. GSU-Spray has a comparably better performance than S & W and S & F, mostly because it

is an asymmetrical spray routing algorithm that utilizes both general and specific utility metrics. Besides, in the second stage of GSU-Spray, the delegation forwarding scheme is employed. And that is also the reason that the average latency of GSU-Spray is the lowest among all protocols, as illustrated in Figure 6(b).

From Figures 6(a), 6(b), 6(c), and 6(d) we know that the message time-to-live property does not have apparent influence on the routing performance. As shown in Figure 6(d), GSU-Spray performs well in average hop count. In this simulation scenario, the bottle neck factor of routing is the buffer size. The simulation result shows that GSU-Spray wins out in three criteria and has the same average hop count performance as S & F. Thus we can conclude that GSU-Spray is a good choice in the network scenario of scarce buffer resource.

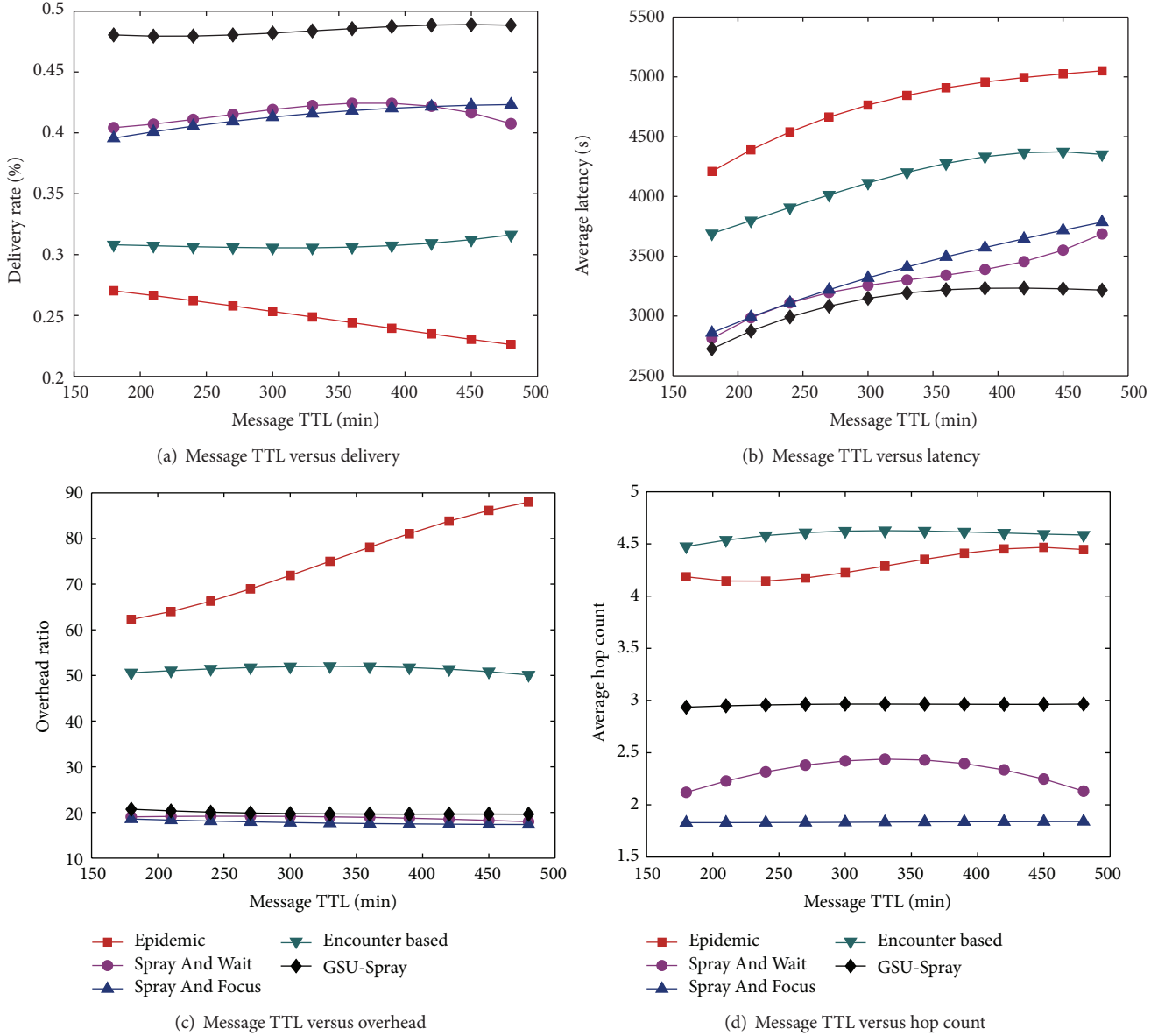


FIGURE 6: Message TTL versus delivery ratio, average latency, overhead ratio, and average hop count (*Helsinki City Scenario*).

5.2. Simulation in Cambridge-iMote Trace. To evaluate the performance of our proposed routing algorithm in the real world scenario, we use the Cambridge Huggle trace dataset. This trace includes Bluetooth sightings by groups of users carrying small devices (iMotes) for two months in various locations that we expected many people to visit such as grocery stores, pubs, market places, and shopping centers in and around the city of Cambridge, UK. Mobile users in this experiment mainly consisted of students from Cambridge University who were asked to carry these iMotes with them at all times for the duration of the experiment. In addition to this, the experiment deployed a number of stationary nodes in various locations that we expected many people to visit such as grocery stores, pubs, market places, and shopping centers in and around the city of Cambridge, UK. A stationary iMote was also placed at the reception of the Computer Lab

in Cambridge University, in which most of the experiment participants are students. The data was gathered from 36 mobile participants and 18 fixed locations. The settings of this simulation are listed in Table 5.

5.2.1. Performance Evaluation by Varying Buffer Size. In this real trace simulation, the buffer size is set to be much larger than that in Helsinki City Scenario. Figure 7(a) shows that the delivery ratio of GSU-Spray is the highest among all these protocols. When the buffer size is larger than 75 MB, GSU-Spray wins out EBR in delivery performance. The results show that in social mobile DTN scenarios the contact duration is a more accurate property that reflects the prospective throughput among nodes than contact frequency. Notice that in this real trace simulation, we set the parameter λ to be 1.0, which is nine times less than the value of 10.0 in Helsinki

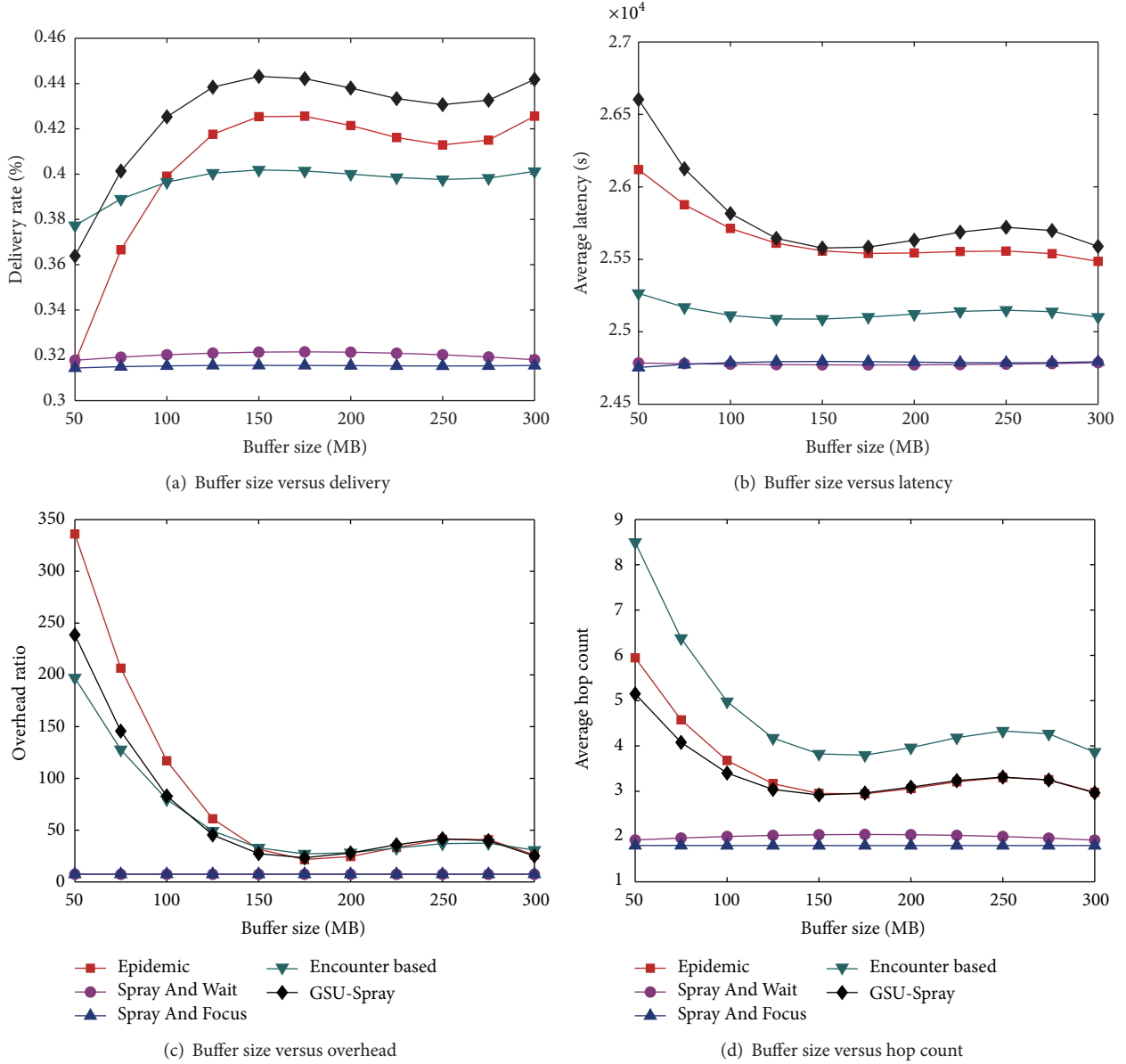


FIGURE 7: Buffer size versus delivery ratio, average latency, overhead ratio, and average hop count (*Cambridge-iMote*).

City Scenario, thus raising the metric's sensitivity to contact duration time of GSU-Spray.

Regarding the result in Figure 7(b), GSU-Spray has the highest latency. S & W stays in the lowest latency level, while its delivery performance is unacceptable. The reason is that the messages in statistics are mainly composed of those that can be delivered quickly. A number of messages that cannot be delivered in a short period are dropped during the routing process. From all the four subfigures of Figure 7, we can see that the buffer resource is not the bottle neck factor of routing performance when it is larger than 150 MB, and in this case GSU-Spray still outperforms Epidemic in delivery performance. As illustrated in Figures 7(a) and 7(b), the result data of GSU-Spray is near to that of Epidemic. Our future work in this field is to further reduce the latency of our proposed algorithm.

Now we focus on the comparison of the three algorithms, Epidemic, EBR, and our proposed GSU-Spray. If a high delivery ratio is expected with the need of an acceptable overhead cost, GSU-Spray is expected to be employed. If we want a lower end-to-end latency, it is more desirable to employ EBR, which outperforms Epidemic and GSU-Spray in average latency performance and is simpler and easier to implement.

5.2.2. Performance Evaluation by Varying Message TTL. In the simulation shown by Figure 8, we set the buffer size constant at 200 MB. As shown in Figure 8(a), with the increase of preassigned time-to-live value, the delivery performance of all these algorithms enhances. Our proposed GSU-Spray still keeps a high delivery ratio approximate to Epidemic. As shown in Figure 8(c), the overhead ratio increases more

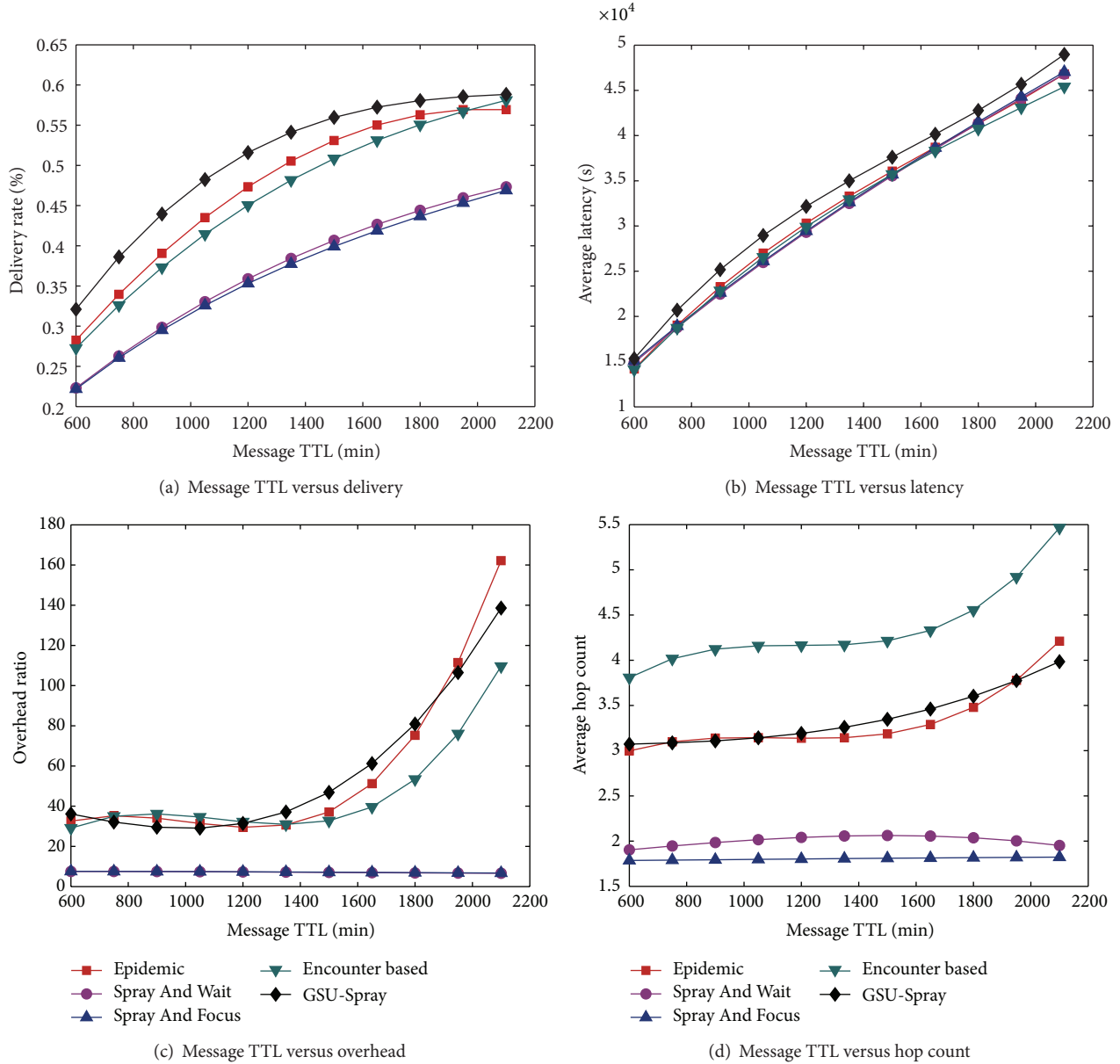


FIGURE 8: Message TTL versus delivery ratio, average latency, overhead ratio, and average hop count (*Cambridge-iMote*).

slowly than that of Epidemic. When there is no need for keeping low average hop count, GSU-Spray has an overall good performance in this real trace scenario.

Figure 8(b) depicts that all four algorithms have slight differences in the performance of average latency. Besides, the growth trend of latency keeps the same. When the message TTL is set to be larger, all these algorithms have a relatively higher average latency. Nevertheless the delivery ratio also rises. However, by comparing Figures 8(a) and 8(b), the latency rises much quicker than delivery ratio with the increase of message TTL. Thus we can infer that there are some messages that are hard to be delivered in a short period. Even so, this can be alleviated by using the Markov Model to adaptively adjust the initial number of copies c , as

stated in Section 4. In Figure 8(d), GSU-Spray has the highest average hop count value, which is however slightly higher than Epidemic.

In conclusion, by referring to the Helsinki City Scenario, we know that our proposed GSU-Spray has relatively better performance when the buffer resource is scarce. As illustrated in the simulation result of *Cambridge-iMote*, since the buffer resource is no longer the bottle neck of routing performance, Epidemic and EBR also have good performance. However, when there is no need for keeping low average latency, if we want to promote the delivery performance to a higher level, GSU-Spray is a better choice. What is important is that GSU-Spray is the only routing algorithm that has overall good performance in the two different scenarios.

TABLE 5: Simulation settings of Cambridge-iMote trace.

Parameter name	Range
Number of nodes	36
Initial tickets number	5
Message TTL (min)	600–2100 (1200)
Simulation time (days)	11.5
Message size (KB)	500–1024
Device buffer (MB)	50–300 (200)
Interface bandwidth (KBps)	250
Message interval (s)	35–40
GSU-Spray- α	0.3
GSU-Spray- λ	1.0

6. Related Work

Reference [20] presents two multicopy forwarding protocols called optimal opportunistic forwarding (OOF) and OOF-, which maximize the expected delivery rate and minimize the expected delay, respectively, while requiring that the number of forwarding operations per message does not exceed a certain threshold. Reference [21] applies the evolutionary games to noncooperative forwarding control in MDTNs, of which the main focus is on mechanisms to rule the participation of the relays to the delivery of messages in DTNs. Reference [12] provides a reliable data delivery scheme for mobile sensor networks with an enhanced delaying technique. Nodes estimate connectivity and expect interencounter time with sink nodes. Connectivity is estimated based on ratio of past and present connections. When the connectivity is unreliable, nodes delay the transmission for the remaining interencounter duration or per-hop lifetime. Reference [22] theoretically proves that considering both factors leads to higher throughput than considering only contact frequency. To fully exploit a social network for high throughput and low routing delay, the authors propose a social network oriented and duration utility-based distributed multicopy routing protocol for DTNs. In [11], the authors find that it is wise to wait till much better opportunities arise to minimize the communication cost without degrading the delivery ratio and latency. Consequently a universal scheme, named E-Scheme, is proposed to improve routing on the delivery probability metric. In [23], the authors propose a distributed optimal community-aware opportunistic routing (CAOR) algorithm that computes the minimum expected delivery delays of nodes through a reverse Dijkstra algorithm and achieves the optimal opportunistic routing performance. By proposing a home-aware community model, whereby turning an MON into a network that only includes community homes, the computational cost and maintenance cost of contact information are greatly reduced.

7. Conclusion

In this paper, we propose a general and specific utility-based adaptive routing algorithm GSU-Spray for mobile

DTNs. Different from the current stochastic routing protocols that forward messages according to a contact frequency-based utility, our proposed algorithm utilizes the contact duration time as the criterion to evaluate the relationship between nodes. Consequently by combining the self-aware and destination-aware utility metric, we design two synthetic metrics to indicate the suitability of a node to be the relay for a certain message, which are employed in the spraying stage and forwarding stage, respectively. When calculating the destination-aware utility, rather than simply accumulating every contact duration time for each node, we design a function by integrating an exponential function $e^{-\lambda t}$ in each duration time interval. Consequently the sensitivity to the contact duration is controlled by the parameter λ . In order to quickly complete the task of message copies spreading, we theoretically analyze the replication process of spraying stage of GSU-Spray and derive a function of the relationship between the expected latency and the number of message copies. Thus the expected latency can be kept under a certain level by dynamically adjusting the preassigned number of copies of each message during the routing process. Simulation results show that GSU-Spray outperforms Epidemic, S & W, S & F, and EBR under the synthetic mobility model and real trace dataset. Observed from the simulation results, GSU-Spray is the only algorithm that achieves comparable performances under the two different scenarios. Our future work will focus on further reducing the average latency of GSU-Spray and evaluating its performance on real-world testbed.

Conflict of Interests

All authors do not have any possible conflict of interests.

Acknowledgments

This research was supported in part by Foundation Research Project of Qingdao Science and Technology Plan under Grant no. 12-1-4-2-(14)-jch and Natural Science Foundation of Shandong Province under Grant no. ZR2013FQ022.

References

- [1] J. Ott, "404 not found," in *Proceedings of the 3rd ACM International Workshop on Mobile Opportunistic Networks (MobiOpp '12)*, p. 3, ACM Press, New York, NY, USA, March 2012.
- [2] K. Wei, X. Liang, and K. Xu, "A survey of social-aware routing protocols in delay tolerant networks: applications, taxonomy and design-related issues," *IEEE Communications Surveys & Tutorials*, no. 99, pp. 1–23, 2013.
- [3] Y. Cao, H. Cruickshank, and Z. Sun, "Active congestion control based routing for opportunistic delay tolerant networks," in *Proceedings of the IEEE 73rd Vehicular Technology Conference (VTC '11)*, pp. 1–5, IEEE, May 2011.
- [4] R. C. Durst, P. D. Feighery, and K. L. Scott, "Why Not Use the standard Internet suite for the interplanetary Internet?" *InterPlanetary Internet (IPN) Technical Information*, 2000.

- [5] M. J. Khabbaz, C. M. Assi, and W. F. Fawaz, "Disruption-tolerant networking: a comprehensive survey on recent developments and persisting challenges," *IEEE Communications Surveys and Tutorials*, vol. 14, pp. 607–640, 2011.
- [6] H. Gong and L. Yu, "Study on routing protocols for delay tolerant mobile networks," *International Journal of Distributed Sensor Networks*, vol. 2013, no. 4, pp. 1–16, 2013.
- [7] A. Vahdat and D. Becker, "Epidemic Routing for Partially Connected ad hoc Networks," Tech. Rep. Cs-2000-06, Duke University, 2000.
- [8] T. Spyropoulos, K. Psounis, and C. S. Raghavendra, "Spray and wait: an efficient routing scheme for intermittently connected mobile networks," in *Proceeding of the ACM SIGCOMM Workshop on Delay-Tolerant Networking (WDTN '05)*, pp. 252–259, ACM Press, New York, NY, USA, August 2005.
- [9] T. Spyropoulos, K. Psounis, and C. S. 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] S. C. Nelson, M. Bakht, and R. Kravets, "Encounter-based routing in DTNs," in *Proceedings of the 28th Conference on Computer Communications (IEEE INFOCOM '09)*, pp. 846–854, April 2009.
- [11] Z. Lin and X. Jiang, "Universal scheme improving probabilistic routing in delay-tolerant networks," *Computer Communications*, vol. 36, pp. 849–860, 2013.
- [12] S. Cha, E. Talipov, and H. Cha, "Data delivery scheme for intermittently connected mobile sensor networks," *Computer Communications*, vol. 36, pp. 504–519, 2013.
- [13] W. Gao and G. Cao, "On exploiting transient social contact patterns for data forwarding in delay-tolerant networks," *IEEE Transactions on Mobile Computing*, vol. 12, no. 1, pp. 151–165, 2013.
- [14] E. M. Daly and M. Haahr, "Social network analysis for routing in disconnected delay-tolerant MANETs," in *Proceedings of the 8th ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc '07)*, pp. 32–40, ACM Press, New York, NY, USA, September 2007.
- [15] V. Erramilli, M. Crovella, A. Chaintreau, and C. Diot, "Delegation forwarding," in *Proceedings of the 9th ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc '08)*, pp. 251–259, ACM Press, New York, NY, USA, May 2008.
- [16] K. L. Scott and S. Burleigh, "Bundle protocol specification," *RFC*, vol. 5050, pp. 1–51, 2007.
- [17] A. Balasubramanian, B. N. Levine, and A. Venkataramani, "Replication routing in DTNs: a resource allocation approach," *IEEE/ACM Transactions on Networking*, vol. 18, pp. 596–609, 2010.
- [18] I. Psaras, N. Wang, and R. Tafazolli, "Six years since first DTN papers. Is there a clear target," *Proceedings of the 1st Extreme Conference on Communications (ExtremeCom '09)*, 2009.
- [19] A. Keränen, J. Ott, and T. Kärkkäinen, "The ONE simulator for DTN protocol evaluation," in *Proceedings of the 2nd International ICST Conference on Simulation Tools and Techniques*, p. 55, ICST, 2009.
- [20] C. Liu and J. Wu, "On multicopy opportunistic forwarding protocols in nondeterministic delay tolerant networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 23, pp. 1121–1128, 2012.
- [21] R. El-Azouzi, F. De Pellegrini, H. B. Sidi, and V. Kamble, "Evolutionary forwarding games in delay tolerant networks: equilibria, mechanism design and stochastic approximation," *Computer Networks*, vol. 57, pp. 1003–1018, 2013.
- [22] Z. Li and H. Shen, "SEDUM: exploiting social networks in utility-based distributed routing for DTNs," *IEEE Transactions on Computers*, vol. 62, pp. 83–97, 2013.
- [23] M. Xiao, I. J. Wu, and L. Huang, "Community-aware opportunistic routing in mobile social networks," *IEEE Transactions on Computers*, pp. 1–13, 2013.

