1. Introduction

Recent years have seen the development of mobile devices such as smartphones, laptops, and tablet PCs, which makes it easier for people to contact and share data with each other in a cheap way. Many researchers use the term “mobile opportunistic networks (MONs)” to describe this special kind of delay-tolerant network (DTN), in which mobile users move around and communicate with each other via their carried short-distance wireless communication devices. Since MONs experience intermittent connectivity incurred by the mobility of users, routing is a mainly concerning and challenging problem [1]. In traditional data networks such as Internet, there are usually some assumptions of the network model, for example, the existence of at least one end-to-end path between source-destination pairs. Any arbitrary link connecting two nodes is assumed to be bidirectional supporting symmetric data rates with low error probability and latency. In addition, the power of each node is considered to be sufficient and thus irrelevant to the node throughput. Messages are buffered in intermediate nodes (e.g., routers) and further forwarded to the next-hop relay or successfully received by the destination. In this case, each message is not expected to occupy the buffer of nodes for a long period of time. However, all the above assumptions usually fail in the context of DTNs. A part of applications in DTNs stresses the delivery ratio while being tolerant of an acceptable end-to-end latency, which is known as the “delay-tolerant” property. For further popularizing these kinds of applications, we have to reconsider the widely used network architecture so as to relax the assumption based by the traditional TCP/IP, that the end-to-end connectivity always exists [2].

There are many research achievements of the routing problem in DTNs, which highly improve the performance in the network scenarios like MONs. Most of them focused on exploiting the strategies for choosing relay node(s) during the routing process [3–5], while few literatures [6–8] considered the effect of data item selection on the routing performance. However, the combination of long-term storage and the message replication performed by many DTN routing protocols imposes a high bandwidth and storage overhead on wireless nodes [9]. Moreover, the data units disseminated in this context, called bundles, are self-contained. What is more, the application-level data units can often be large [10]. As a result, the nodes’ buffer, in this context, usually works at a full load status. Similarly, the available bandwidth for a certain connection is likely to be insufficient to have all the buffered messages forwarded. Consequently, regardless of the specific routing algorithm used, it is important to have efficient scheduling policies to decide which message(s)
should be chosen to exchange with another encountered node when bandwidth and connection duration are limited.

In this paper, we try to improve the classical probability routing protocol PRoPHET [11] from this point. By defining the concept of “Transmission Profit,” we introduce the “Optimal Throughput-Aware Probabilistic Routing,” which is consequently modeled as an optimal decision-making problem and is solved by the dynamic programming technique. Based on this model, a data item selection mechanism for PRoPHET is devised in this paper. The data item selection mechanism in this paper applies to the network scenarios with the two following characteristics.

(1) The average throughput of the connection between each pair of nodes is far smaller than the size of messages in their buffer.

(2) The energy consumption for each transmission may not be ignored, which highlights the importance of every successful relay operation.

For example, a transmission operation for a message would fail when the available throughput of the used connection is smaller than the size of this message. However, a solution to this problem is to send a message of which the size is not exceeding the connection throughput, thus avoiding the waste of the transmission opportunity. To say the least, even if the throughput of the connection is sufficient for sending either of the messages, the whole profit we gain for each transmission (e.g., the delivery probability or the end-to-end latency) would be distinctive when different messages are selected to send. From this perspective, an efficient data item selection mechanism is expected to be employed in challenged network scenarios.

The rest of this paper is organized as follows. In Section 2 we introduce the system model and the routing model. In Section 3 we formally define the key problem. The improved protocol with the data item selection is given in Section 4. In Section 5 we analyze the simulation result. Section 6 reports on previous similar work. We conclude the paper and discuss future work in Section 7.

2. Preliminary

The mathematical notations are listed in Table 1. In our model, we use a discrete timeline that is divided into many small time slots of which the length is defined as a unit time. We denote the whole nodes set as $\mathcal{N} = \{n_i \mid n_i \in \mathcal{N}, 1 \leq i < |\mathcal{N}|\}$. There is a message time-to-live value $TTL(k)$ for each generated message $m_k$. When a message is generated by a node, the $TTL(k)$ is preassigned by the corresponding application. Messages would be dropped if their TTLs are exhausted. The size of each message $m_k$ is denoted by $S(k)$. We assume that each node $n_i$ knows its own bandwidth $B(i)$. This assumption is acceptable due to the fact that the network interface of each device is usually immovable. To say the least, even the bandwidth of each node is not stable, and we can easily measure the average value by a slide window. For any encountered node $n_j$ of $n_i$, we let $n_i$ record two values $t_{(i,j)}^s$ and $t_{(i,j)}^e$ that record the start and the end time of the most recently happening contact, respectively. Then $t_{(i,j)}^e - t_{(i,j)}^s$ is the duration time of this contact, and we will simply use $t_i$ and $t_e$ when there would be no ambiguousness in the context.

PRoPHET routing algorithm records history of encounters and transitivity, and the utility metric is based on an encounter probability with the transitivity. PRoPHET estimates a probabilistic metric called delivery predictability, $P_{(a,b)}$, at every node $n_a$, for each known destination $n_b$. This indicates how likely it is that this node will be able to deliver a message to that destination. The calculation process is listed as follows:

\[
P_{(a,b)} = P_{(a,b)} + \left(1 - P_{(a,b)}\right) \times P_{init},
\]

\[
P_{(a,b)} = P_{(a,b)} \times \gamma^k,
\]

\[
P_{(a,c)} = P_{(a,c)} + \left(1 - P_{(a,c)}\right) \times P_{(a,b)} \times P_{(b,c)} \times \beta,
\]

where $P_{(a,b)}$ denotes the delivery predictability of reaching $n_b$ from $n_a$ and $P_{init}$, $\beta$, and $\gamma$ are initialization constants chosen from the range $[0,1]$. Each node maintains a $1 \times |\mathcal{N}|$ vector, with $|\mathcal{N}|$ representing the number of nodes, where each element $i$ records the delivery predictability between $n_i$ and $n_j$.

3. Problem Formalization

In this section, we give the details of our proposed data item selection scheme.

3.1. Objective. The objective is to maximize the delivery probability of each message to the destination. In MONs, each node routes the message in a “store-carry-forward” way. We can choose to always forward the message to the node with higher meeting probability to its destination. However, this simple strategy does not take the throughput issue into
consideration, which is highly relative to the bandwidth and the connection duration time. Since the bandwidth and the connection duration time between each pair of nodes are limited, the forward sequence of messages in the queue has great effect on the routing performance. Assume that $n_a$ has three messages $m_1$, $m_2$, and $m_3$ for transmission to $n_b$, of which the sizes are 150 k, 200 k, and 100 k. However the current connection can only carry a data flow maximal to 120 k in total. Thus neither $m_1$ nor $m_2$ can be successfully relayed from $n_a$ to $n_b$ due to the highly constrained throughput of the connection. It is not rational yet to let node $n_a$ forward the message with the smallest size to $n_b$. The reason is that there is no explicit optimization objective, which might take away the relay opportunity of those messages that have a little larger size, but great improvement on the delivery probability after being replicated to $n_b$. Consequently, it is necessary to choose an explicit optimization objective for our selection. In this paper, we primarily focus on how to efficiently enhance the delivery performance. In the next part we give analysis process of maximizing the profit on delivery probability.

If the message is forwarded to $n_b$, then the delivery would fail only if both $n_a$ and $n_b$ fail to deliver the message, and the delivery probability for this case can be computed by the following equation:

$$P_{(a,b)} = 1 − (1 − P_{(a,d)}) (1 − P_{(b,d)}).$$ (4)

Thus we have the following definition.

**Definition 1** (transmission profit). The transmission profit $\zeta(i)$ is the magnitude of improvement on delivery ratio for message $m_i$, where

$$\zeta(i) = p^{(a,b)} - p^{(a,d)} = 1 - (1 - P_{(a,d)}) (1 - P_{(b,d)}) - P_{(a,d)}.$$ (5)

The value of variable $\zeta(i)$ reflects the improved result of the delivery probability for message $m_i$. From this point, one consequently defines one's "Optimal Throughput-Aware Probabilistic Routing" as follows.

**Definition 2** (optimal throughput-aware probabilistic routing). The optimal probabilistic routing always tries to maximally improve the delivery probability for each message to the destination, by taking the estimated bandwidth of current connection into consideration, that is, each node $n_i$ forwards several selected messages to a node $n_j$ with corresponding top improved $\zeta$ values by making the most use of the estimated available throughput of current connection.

For example, as shown in Table 2, there are five messages in the buffer of $n_a$. For simplification, we number the five messages as $m_1$ to $m_5$. And the destination node of $m_i$ is represented as $d_i$. For any message $m_i$ ($1 \leq i \leq 5$), we have $P_{(b,d_i)} > P_{(a,d_i)}$, which indicates that, when $n_a$ meets $n_b$, all these five messages should be forwarded from $n_a$ to $n_b$. By using (5) we can get the $\zeta$ value for each message. In the next part of this section, we firstly show the method to estimate the available connection throughput, and then we give the formal expression for our problem.

**3.2. Estimating the Available Throughput.** Consider the following definition.

**Definition 3** (throughput of connection). Given a connection duration time $t$ between two nodes and the bandwidth $B$ (KB/unit) of the connection between them, the throughput $TH$ of the connection is

$$TH = B \cdot t.$$ (6)

Since we assume that the bandwidth $B$ of the connection is known in advance, the remaining task of estimating the throughput is to get the connection duration time $t$. We can use the following equation to estimate the duration time for the next upcoming connection between $n_a$ and $n_b$, where $\alpha \in [0, 1]$ is the scaling constant and $t_e - t_s$ is the duration time of the most recently happening connection, of which the impact is controlled by the parameter $\alpha$. When $\alpha$ is set to be relatively large, the impact of the second part of the following equation would be enhanced and vise versa. Consider

$$\tau_{(a,b)} = (1 - \alpha) \tau_{(a,b)}^t + \alpha (t_e - t_s).$$ (7)

When $n_a$ has not met $n_b$ for a while, we use the following equation to update $\tau_{(a,b)}$, where $\gamma \in [0, 1]$ is the same aging constant in (2) and $k$ is the number of time units that have elapsed since the last time the metric was aged:

$$\tau_{(a,b)} = \tau_{(a,b)}^t \gamma^k.$$ (8)

In each time unit, the node checks the status of the connection between $n_a$ and $n_b$. The updating process for $\tau$ is shown in Algorithm 1. Then according to Definition 3,

| Message size | $m_1$ | $m_2$ | $m_3$ | $m_4$ | $m_5$ |
|--------------|-------|-------|-------|-------|-------|
| $P_{(a,d)}$  | 0.0   | 0.2   | 0.1   | 0.12  | 0.2   |
| $P_{(b,d)}$  | 0.1   | 0.75  | 0.2   | 0.25  | 0.35  |
| Value of $\zeta$ | 0.1 | 0.6 | 0.18 | 0.22 | 0.28 |

**Table 2: Five messages in $n_a$'s buffer.**

**Algorithm 1**: Updating the $\tau$ value.
the throughput of the connection between \( n_a \) and \( n_b \) can be estimated by the following equation:

\[
ETH_{(a,b)} = B_{(a,b)} \cdot \tau_{(a,b)}. \tag{9}
\]

An example is shown in Figure 1. The connection between \( n_a \) and \( n_b \) has shown up 3 times in the time interval \([0, 11]\). The variable \( k \) is reset to be 1 in the black square and increases by 1 in the white square. At the starting time 0, we set \( \tau = 0 \) and \( k = 1 \). In the red square, the variable \( \tau \) is updated by (8), while in the green square \( \tau \) is updated by (7). The whole computing process conforms Algorithm 1. Finally we obtain the estimated value \( \tau = 1.62 \). Assuming the bandwidth of the connection is 6.8 KB/unit (this numerical value is easier for the discussion below), then the throughput of the connection can be estimated by (9) as

\[
ETH_{(a,b)} = B_{(a,b)} \cdot \tau_{(a,b)} = 6.8 \times 1.62 = 11 \text{ KB}. \tag{10}
\]

3.3. Formalization. We first show that the routing problem can be viewed as a 0-1 knapsack problem, and then we give the formalization of our routing problem.

**Theorem 4.** The optimal bandwidth-aware probabilistic routing problem can be formalized to be an optimal decision-making problem and, furthermore, can be viewed as a 0-1 knapsack problem.

**Proof.** If viewing the estimated throughput \( ETH_{(a,b)} \) as the maximum weight that knapsack can carry, each message as an item in the knapsack problem, the improvement value \( \xi(i) \) for each message as the value of each item, and the size of each message \( S(i) \) as the weight of ith item, then the routing problem is equivalent to the corresponding 0-1 knapsack problem, where decisions are made on each item to achieve the maximal profit.

Theorem 4 shows that the routing problem is an optimal decision-making \( NP-complete \) problem. The problem is formally defined as follows.

**Definition 5** (formalization of routing problem). Assume that the optimal bandwidth-aware probabilistic routing problem is an optimal decision-making problem; that is,

\[
\text{Max} \sum_{i=1}^{n} \xi(i) x_i \tag{11}
\]

\[
s.t. \sum_{i=1}^{n} S(i) x_i \leq ETH_{(a,b)}, \quad x_i \in \{0, 1\}, \quad i = 1, \ldots, n.
\]

4. Improved Routing Protocol with Data Item Selection

In this section, we give the detailed information about the improved routing protocol. The key problem of routing is formalized in Section 3. We first apply dynamic programming to the data item selection problem. Then we illustrate how to maintain the needed information during the entire routing process. Finally we show the entire procedure of our improved routing protocol.

4.1. Solving the Optimal Decision Problem. The scheme to solve the optimal decision-making problem is stated in Algorithm 2. The calculating process is shown in Lines 1–5. And the process of solution construction is stated in Lines 6–19.

Overviewing the example through this paper, the value of corresponding \( \xi \) and the size of messages are shown in Table 2. In Section 3.2 the estimated throughput for the connection between \( n_a \) and \( n_b \), that is, \( ETH_{(a,b)} \), is calculated. Based on all the above, the calculation process of the example is shown in Table 3.

4.2. Protocol Description. Now we focus on the routing protocol. The same as that in PRoPHET, first of all we need to maintain a table recording the meeting probability. Besides, since the estimation of connection throughput is based on the connection duration time, we also need to let each node record the variables \( t_a \) and \( t_e \) and the corresponding estimation value \( \tau \) for the most recently happening connection. Finally, we need to record the number of time units that have elapsed since the last time the metric was aged, that is, \( k \), for each connection. We denote the routing information table of \( n_a \) by \( TABLE[a] \), which is shown in Table 4, and the space complexity of \( TABLE[a] \) is \( O(N^2) \).

There are two parts of our entire protocol. The information exchange protocol is shown in Algorithm 3 and the data transmission protocol is shown in Algorithm 4.

In Algorithm 3, the primary task is to update the needed information for timely routing and then to exchange it with neighbor nodes. The updating process is stated in Lines 1–4, where the equation in PRoPHET and our updating algorithm are used. In Line 5 the request for \( TABLE \) is broadcast to all the neighbor nodes of current node \( n_a \). As shown in Lines 6–8, if current node \( n_a \) receives the request from any other node, \( TABLE[a] \) will be transmitted to that node. In Lines 9–11, when \( n_a \) receives \( TABLE[b] \) from any node \( n_b \), then the data transmission protocol will be called. In other words, which is the triggering condition of the data transmission protocol.

In Algorithm 4, the current node \( n_a \) scans its buffer, adding all messages that let \( P(b,d_i) > P(a,d_i) \) hold to the message list. In our scheme, the forward strategy is the same as PRoPHET, that a message will be transferred from...
**Algorithm 2:** Get the solution by dynamic programming.

**Triggering condition:**

In each time unit

**n_a Executes:**

1. for each column record $i$ of $n_a$ in Table 4 do
2. $\mathcal{P}(a,b)$ by (1)–(3).
3. call Algorithm 1 to update $\tau_{(a,b)}$.
4. end for
5. broadcast the request for TABLE to $n_a$.neighbors
6. if received request for TABLE from any node $n_b$ then
7. forward TABLE[a] to $n_b$
8. end if
9. if received TABLE[b] from any node $n_b$ then
10. call Algorithm 4
11. end if

**Algorithm 3:** Information exchange protocol.

**Algorithm 4:** Data transmission protocol.
node a to node b only if the b's contact predictability to the destination node is higher than that of the other node. However, the throughput of each connection is taken into consideration, as shown in the remaining part of this algorithm. We firstly estimate the throughput of the connection between $n_a$ and $n_b$ in Line 7. Then the transmission profit value $\xi(i)$ is calculated for each message $m_i$ by (5). Finally we employ Algorithm 2 to obtain the ForwardList, where all elements are extracted from the MessageList. We sort all the messages in ForwardList by ascending order according to the TTL to give the expiring messages a higher priority for transmission, so as to lower the number of dropped messages. Then in Line 13 all the messages in ForwardList are transmitted by $n_a$ to $n_b$ in the ascending order of message TTL.

5. Simulation

The results are evaluated by the ONE simulator [12]. We firstly adopt the real experimental trace of the Cambridge-iMote dataset, since it is one of the most extensive and widely exploited data traces. This trace includes Bluetooth sightings by groups of users carrying small devices (iMotes) for two months in various locations that we expected many sightings by groups of users carrying small devices (iMotes) widely exploited data traces. This trace includes Bluetooth iMote dataset, since it is one of the most extensive and first to adopt the real experimental trace of the Cambridge-iMote real trace. The results are evaluated by the ONE simulator [12]. We firstly estimate the throughput of the connection between $n_a$ and $n_b$ in Line 7. Then the transmission profit value $\xi(i)$ is calculated for each message $m_i$ by (5). Finally we employ Algorithm 2 to obtain the ForwardList, where all elements are extracted from the MessageList. We sort all the messages in ForwardList by ascending order according to the TTL to give the expiring messages a higher priority for transmission, so as to lower the number of dropped messages. Then in Line 13 all the messages in ForwardList are transmitted by $n_a$ to $n_b$ in the ascending order of message TTL.

Table 3: Calculation process using dynamic programming.

| $i$ | 1   | 2   | 3   | 4   | 5   |
|-----|-----|-----|-----|-----|-----|
| $S(i)$ | 1K  | 2K  | 5K  | 6K  | 7K  |
| $\xi(i)$ | 0.10 | 0.60 | 0.18 | 0.22 | 0.28 |

Table 4: The routing information table.

| $P_{(a,n_1)}$ | $P_{(a,n_2)}$ | $P_{(a,n_3)}$ | ... | $P_{(a,n_N)}$ |
|---------------|---------------|---------------|-----|---------------|
| $k_1$         | $k_2$         | $k_3$         | ... | $k_{N+1}$     |
| $t_{(a,n_1)}$ | $t_{(a,n_2)}$ | $t_{(a,n_3)}$ | ... | $t_{(a,n_N)}$ |
| $t_{(a,n_1)^1}$ | $t_{(a,n_2)^1}$ | $t_{(a,n_3)^1}$ | ... | $t_{(a,n_N)^1}$ |

Table 5: Simulation settings of Cambridge-iMote trace.

| Parameter name | Range       |
|----------------|-------------|
| Number of nodes | 36          |
| Entire simulation time (days) | 11.5         |
| Message size (KB) | 500–1024     |
| Bandwidth (KBps) | 250          |
| $P_{init}$ | 0.75        |
| $\alpha$ | 0.5         |
| $\beta$ | 0.25        |
| $\gamma$ | 0.98        |

mobility model. The simulation is grouped into the following categories: (1) varying buffer size in Cambridge-iMote real trace; (2) varying message time-to-live in Cambridge-iMote real trace; (3) varying buffer size in Helsinki City Model; (4) varying message time-to-live in Helsinki City Model. We compare five different routing protocols based on delivery ratio, overhead ratio, and average latency.

5.1. Simulation in Cambridge-iMote Real Trace. We conduct the simulations by generating 3,300 messages for randomly selected source nodes and by executing the above-mentioned algorithms to forward these messages to their destinations, while recording the delivery ratio, average latency, and average hop count. In simulations on evaluating all these metrics, we set the simulation time as 25%, 50%, 75%, and 100% of the entire time of the dataset. Figure 2 shows the simulation results on varying buffer size, with the message TTL constant at 300 minutes. In Figure 3, we show the simulation results on varying the message TTL, with the node buffer size constant at 50 M. The other settings of the simulation are listed in Table 5.

Regarding all the figures in Figure 2, the results show that our improved throughput-aware routing significantly outperforms PRoPHET and Epidemic routing in delivery ratio and average latency and reduces the overhead to an extent. Though the delivery performance between MaxProp and our scheme varies a little, our proposed routing outperforms MaxProp in either of the remaining two criteria. More specifically, from Figures 2(a), 2(d), 2(g), and 2(j), we can see that the effect on the improvement of delivery performance increases with the simulation time prolonged. As shown in the second column of Figure 2, the improved throughput-aware routing has a comparably lower latency than PRoPHET. By referring to the third column of Figure 2, our proposed routing has greater improvement on overhead ratio with the simulation time set longer. From all the subfigures, we can see that the throughput-aware routing has the overall best performance among all the five protocols.

Regarding all the figures in Figure 3, the results show that our proposed routing significantly outperforms the other two protocols in delivery and overhead, while having a slight improvement on average latency. With the simulation time prolonged, the influence of our proposed scheme has greater improvement on all of the three metrics. However, we do not see much improvement on the average latency. From
all the subfigures in Figure 3, our proposed routing has a relatively better performance than Epidemic, PRoPHET, and EncounterBased routing. Compared with MaxProp, our proposed routing performs much better when the whole simulation time is short, which indicates that the proposed routing reaches the best status more quickly in real network scenarios.

5.2. Simulation in Helsinki City Scenario. In Helsinki City Scenario [12], 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 is cars. The speed of cars is set to be 1050 km/h and the speed of trams 2536 km/h, with the pause time of 10120 s and 1030 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 6.

Regarding all the figures in Figure 4, the results show that our improved throughput-aware routing significantly outperforms EncounterBased, PRoPHET, and Epidemic routing in delivery ratio. With the number of active pedestrians and cars increasing, MaxProp routing gradually outperforms the others in all the three criteria. But when there are not enough active pedestrians and cars in the network, the performance of MaxProp routing is almost the same as PRoPHET. In this case, our proposed scheme significantly outperforms the others in all the three criteria. More specifically, from Figures 4(a), 4(d), and 4(g), we can see that the effect on the improvement of delivery performance increases
with the number of pedestrians/cars. As shown in the second column of Figure 4, the improved throughput-aware routing has almost the same latency as PRoPHET. By referring to the third column of Figure 4, our proposed routing has greater improvement on overhead ratio. From all the subfigures, we can see that the throughput-aware routing has the overall best performance when the number of nodes in the network is set to be relatively small.

Figure 5 shows a similar result as in Figure 4 which is that, when the number of nodes is relatively small, our proposed routing outperforms the other four protocols in delivery and overhead and has a slight improvement on average latency. From all the subfigures in Figure 3, our proposed routing performs better than its original edition PRoPHET, which reflects that the same relay node choosing strategy with different data item selection mechanism has totally variant performance. In all, when the nodes are relatively abundant, that is, the density of nodes is large in the network, we prefer to choose MaxProp. On the other hand, our proposed scheme has a comparable improvement on the PRoPHET routing thus making it suitable to work in the network with relatively low density of nodes.

![Figure 4: Helsinki City Scenario] Buffer size versus overhead with different number of nodes.

![Table 6: Simulation settings of Helsinki City Scenario.]

| Parameter name               | Range (default value) |
|------------------------------|-----------------------|
| Pedestrians/cars             | 30–90                 |
| World size (m x m)           | 4500 x 3000           |
| Initial tickets number       | 6                     |
| Message TTL (min)            | 180–425 (300)         |
| Simulation time (hours)      | 12                    |
| Message size (KB)            | 500–1024              |
| Pedestrian buffer (MB)       | 5–55 (25)             |
| Tram buffer (MB)             | 500                   |
| Bluetooth range (m)          | 10                    |
| High-speed range (m)         | 1000                  |
| Bluetooth (Kbps)             | 250                   |
| High-speed (MBps)            | 10                    |
| Pedestrian speed (m/s)       | 0.5–1.5               |
| Message interval (s)         | 35–40                 |

Figure 4: Helsinki City Scenario Buffer size versus overhead with different number of nodes.

Table 6: Simulation settings of Helsinki City Scenario.
6. Related Works

In [8], Zhu et al. proposed a routing algorithm taking full advantage of predicted probabilistic vehicular trajectories by which the packet delivery probability was theoretically derived. This paper demonstrates that predicted trajectories do help data delivery in vehicular networks. One of the most classical probabilistic routing schemes is probabilistic routing protocol using history of encounters and transitivity (PRoPHET) [11]. In PRoPHET, the utility metric is based on an encounter probability with the transitivity property. For example, given that \( n_a \) most likely encounters \( n_b \) and in similar manner that \( n_b \) encounters \( n_c \), then \( n_c \) may be a good candidate node for node A even if its encounter is least likely. Therefore, messages carried by \( n_a \) would also be replicated to \( n_c \), in addition to \( n_b \), alleviating the buffer space exhaustion at \( n_b \). In particular, the aging factor is also taken into account for the outdated information.

Reference [13] 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 of per message does not exceed a certain threshold. Reference [14] 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 in the delivery of messages in DTNs. Reference [15] 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 [16] theoretically proves that considering both factors leads to higher throughput than considering only contact frequency. And, to fully exploit a social network for high throughput and low routing delay, the authors propose a social network oriented routing protocol for DTNs, in which a duration utility-based metric is utilized for evaluating the most suitable the relay node for each message.
In [4], 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 [3], the authors propose a distributed optimal community-aware opportunistic routing (CAOR) algorithm, where a reverse Dijkstra algorithm is devised so as to compute the minimum expected delivery delays of nodes, thus achieving 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 try to improve the routing performance by resorting to an efficient data item selection mechanism in MONs. Our motivation is that, due to the fact that the bandwidth and contact duration time of each connection are highly constrained, a routing protocol would perform very differently with various data selection strategies. By defining the concept of “Transmission Profit,” the concept of “Optimal Throughput-Aware Probabilistic Routing” is introduced, which is consequently modeled as a dynamic programming problem. Then a data item selection algorithm for PROPHET is devised in this paper. The simulation results show that our data item selection mechanism greatly reduces the number of aborted transmissions thus enhancing the routing performance in aspects of delivery probability, average latency, and overhead ratio. Besides, it is possible to apply the proposed scheme in improving other metrics by redefining the “Transmission Profit.” Our future work will be focused on evaluating the improvement of the data item selection mechanism on various routing protocols.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

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] 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, pp. 1–23, 2013.

[2] J. Ott, “404 not found? A quest for DTN applications,” in Proceedings of the 3rd ACM International Workshop on Mobile Opportunistic Networks (MobiOpp ’12), pp. 3–4, ACM Press, New York, NY, USA, March 2012.

[3] M. Xiao, I. J. Wu, and L. Huang, “Community-aware opportunistic routing in mobile social networks,” IEEE Transactions on Computers, pp. 1–13, 2013.

[4] Z. Lin and X. Jiang, ”Universal scheme improving probabilistic routing in delay-tolerant networks,” Computer Communications, vol. 36, pp. 849–860, 2013.

[5] J. Wu, M. Xiao, and L. Huang, “Homing spread: community home-based multi-copy routing in mobile social networks,” in Proceedings of the IEEE International Conference on Computer Communications (INFOCOM ’13), pp. 2319–2327, April 2013.

[6] J. Burgess, B. Gallagher, D. Jensen, and B. N. Levine, “MaxProp: routing for vehicle-based disruption-tolerant networks,” in Proceedings of the 25th IEEE International Conference on Computer Communications (INFOCOM ’06), pp. 1–11, April 2006.

[7] Y. Zhang and J. Zhao, “Social network analysis on data diffusion in delay tolerant networks,” in Proceedings of the 10th ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc ’09), pp. 345–346, ACM Press, New York, NY, USA, May 2009.

[8] Y. Zhu, Y. Wu, and B. Li, “Trajectory improves data delivery in urban vehicular networks,” IEEE Transactions on Parallel and Distributed Systems, vol. 25, pp. 1089–1100, 2014.

[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] A. Krifa, C. Barakat, and T. Spyropoulos, “Message drop and scheduling in DTNs: theory and practice,” IEEE Transactions on Mobile Computing, vol. 11, pp. 1470–1483, 2012.

[11] A. Lindgren, A. Doria, and O. Schelen, “Probabilistic routing in intermittently connected networks,” ACM SIGMOBILE Mobile Computing and Communications Review, vol. 7, no. 3, pp. 19–34, 2003.

[12] A. Keränen, J. Ott, and T. Kärkkäinen, “The ONE simulator for DTN protocol evaluation,” in Proceedings of the Second International ICST Conference on Simulation Tools and Techniques (ICST ’09), p. 55, 2009.

[13] C. Liu and J. Wu, “On multicopy opportunistic forwarding protocols in nondeterministic delay tolerant networks,” IEEE Transactions on Parallel and Distributed Systems, vol. 23, no. 6, pp. 1121–1128, 2012.

[14] 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.

[15] S. Cha, E. Talipov, and H. Cha, “Data delivery scheme for intermittently connected mobile sensor networks,” Computer Communications, vol. 36, pp. 504–519, 2013.

[16] Z. Li and H. Shen, “SEDUM: exploiting social networks in utility-based distributed routing for DTNs,” IEEE Transactions on Computers, vol. 62, no. 1, pp. 83–97, 2013.