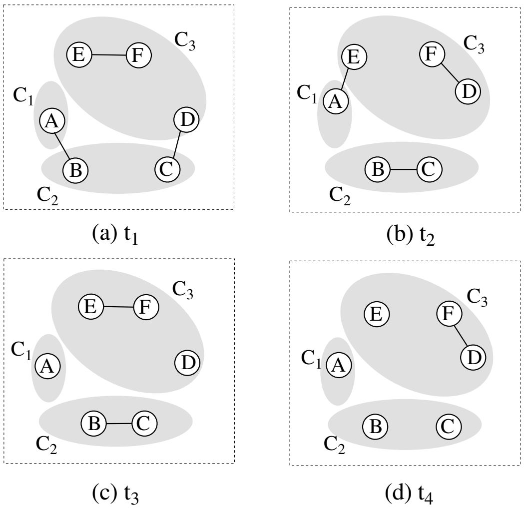
On Using Contact Expectation for Routing in Delay Tolerant Networks

Honglong Chen and Wei Lou
Department of Computing, The Hong Kong Polytechnic University, Kowloon, Hong Kong
Email: {cshlchen, csweilou}@comp.polyu.edu.hk

***Abstract*—Conventional routing algorithms rely on the existence of persistent end-to-end paths for the delivery of a message to its destination via a predesigned path. However, in a delay tolerant network (DTN), nodes are intermittently connected, and thus the network topology is dynamic in nature, which makes the routing become one of the most challenging problems. A promising solution is to predict the nodes' future contacts based on their contact histories. In this paper, we first propose an expected encounter based routing protocol (EER) which distributes multiple replicas of a message proportionally between two encounters according to their expected encounter values. In case of single replica of a message, EER makes the routing decision by comparing the minimum expected meeting delay to the destination. We further propose a community based routing protocol (CR) which takes advantages of the high contact frequency property of the community. The simulations demonstrate the effectiveness of our proposed routing protocols under different network parameters.**

***Keywords*-Delay Tolerant Networks; Expected Encounter Value; Minimum Expected Meeting Delay; Routing Protocols.**



(a) t_1 (b) t_2 (c) t_3 (d) t_4

Figure 1. A sample delay tolerant network with six intermittently connected nodes. C_1, C_2 and C_3 denote three different communities in the network.

I. INTRODUCTION

In a conventional network, messages are routed along persistent end-to-end paths predesigned on the always-connected network topology. However, this kind of routing strategy is not applicable to delay tolerant networks (DTNs), since the predesigned end-to-end path does not always exist. As the nodes may be mobile, the contact between each pair of nodes is intermittent and the network topology changes over time. Therefore, it becomes a most challenging task to design an efficient routing protocol in DTNs.

As the nodes in a DTN are intermittently connected, it is very difficult, if not impossible, to determine a persistent source-to-destination route for each message. The nodes can adopt the store-carry-and-forward mechanism to deliver the messages. However, it is still hard for a node to obtain the global network connectivity as it is time-varying. Figure 1 shows a simple network with six nodes. The network topology varies from time t_1 to t_4, making the routing in this network challenging. For instance, if node A wants to send a message to node D at t_1, according to the global network connectivity, the optimal path for this message is from node A to node E at t_2, then from node E to node F at t_3 and finally from node F to node D at t_4. However, node A may apply the best effort strategy to deliver the message to node B at t_1 since it meets node B firstly, resulting in failing to deliver the message to node D finally. Fortunately, by referring to the historical mobility, a node can predict its future contacts with other nodes, which are useful in making routing decisions.

A promising solution is proposed in [1] that predicts nodes' future contacts based on their contact histories: Each node estimates its future encounter value (EV) based on its contact history. When two nodes meet, the replicas of a message are distributed between them according to the proportion of their estimated EVs. This approach can achieve good performance with a low overhead. However, the EV estimated in [1] is identical to all messages and independent of the time-to-live (TTL) values of the messages. Since each message has its own TTL and should be delivered to the final destination before its TTL expires, the TTL of the message should be taken into consideration for estimating EV. For example, if a node estimates its EV as e per day, which suggests that this node will meet e other nodes in one day. However, if the residual TTL of the message is only one hour, then it is unwise to make the replicas distribution according to e. A better solution is to predict the EV based on the message's TTL.

Embedded this idea, in this paper, we propose an expected encounter based routing protocol (EER) to solve this problem. EER has two phases: multiple replicas distribution and single replica forwarding. In the multiple replicas distribution phase, each node disseminates the replicas of a message to different nodes as soon as possible, which can be achieved by distributing the replicas of the message according to their expected EVs. The expected EV is calculated as a function of the message's TTL, which is more accurate in predicting the future EV in a fixed future time interval. In the single replica forwarding phase, each node decides whether to forward the message to its current encounter by comparing their minimum expected meeting delays (MEMDs) to the destination. The MEMD is calculated based on the past meeting intervals between each pair of nodes and the elapsed time since their last contact. We further propose a community based routing protocol (CR) which takes advantages of the high contact frequency property of the community. CR includes inter-community routing and intra-community routing. In the inter-community routing, each node disseminates the multiple replicas of a message to the nodes from different communities as soon as possible, in which the distribution of the replicas of this message is proportional to the expected numbers of encountering communities of any pair of encounters. In case of the single replica of the message left during the propagation, the message is delivered to the node which has a higher probability to encounter the destination community. In the intra-community routing, a node in the destination community distributes the replicas of a message to its encounter in the same community according to the proportion of their expected EVs within the community. In case of the single replica of the message, the node in the destination community decides whether to forward the message to its encounter in the same community by comparing their MEMDs within the community, which leads the message to be forwarded to its final destination.

Our key contributions of this paper are summarized as follows:

- We formulate the calculation of the expected EV of each node and the minimum expected meeting delay between each node and the destination, then propose the expected encounter based routing protocol;
- We propose the community based routing protocol using the expected number of encountering communities which takes advantages of the community property and can achieve high delivery ratio with less information exchange overhead;
- We conduct simulations to illustrate the effectiveness of our proposed routing protocols under different network parameters.

The rest of this paper is organized as follows. In Section II, we discuss the existing DTN routing protocols. Section III describes our proposed expected encounter based routing protocol. In Section IV, we propose the community based routing protocol. The performance evaluation is conducted in Section V. Section VI concludes this paper and puts forward the future work.

II. RELATED WORK

Routing protocols in DTNs attempt to deliver the message through the intermittently connected nodes to the destination. The property of contemporaneous links makes routing in DTNs a challenging issue. The DTN routing has become an active research topic and many protocols [2]–[5] have been proposed in the past few years.

To obtain a high message delivery ratio, Vahdat *et al.* propose the epidemic routing protocol [6], in which each message is replicated and flooded to all the nodes in the network. However, in this protocol, the number of replicas of each message in the network increases rapidly, which greatly consumes the limited buffer space and bandwidth. To reduce the overhead, some improved epidemic-based routing schemes [2] [7] [8] are proposed. In Prophet [2], a node will replicate and forward a message to its encounter only if its encounter has a higher likelihood of meeting the destination. In MaxProp [7], due to the limitation of buffer space, each node schedules both the packets to be transmitted to other nodes and the packets to be dropped. Thus, the most likely to be successfully delivered packets will be replicated and transmitted to other nodes with the highest priority, and the node will drop the packet with the smallest probability to be successfully delivered when the buffer is full. In the delegation forwarding [8], the message is replicated and delivered by a node to its encounter only if its encounter has a better quality metric. This method can reduce the cost of the network to $O(\sqrt{n})$, compared to the cost $O(n)$ of the epidemic routing, where n is the number of nodes in the network.

To avoid a high network overhead, some forwarding-based routing schemes with single copy [9]–[11] are also proposed. In [9], the delay tolerant network routing problem is formulated, and several routing algorithms corresponding to the percentage of knowledge are proposed. Jones *et al.* [10] design a practical single copy routing mechanism based on the minimum estimated expected delay, which is calculated based on the average meeting interval between each pair of nodes in the network using the Dijstra's algorithm. While it, as a link-state routing algorithm, requires the pair-wise routing information exchange among encountering nodes which introduces additional overhead, it can provide the complete topology at each node to achieve good performance. The predict and relay (PER) [11] scheme relies on predicting the future contacts based on the semi-markov process. Gao *et al.* [12] propose a forwarding approach by exploiting the transient node contact patterns, based on which each node can make a more accurate prediction for data forwarding decision.

A tradeoff between the epidemic-based routing and the forwarding-based routing with single copy is the quota-based routing protocols [13]–[16] [1], which implant predefined replicas of each message into the network to improve the delivery ratio without greatly increasing the overhead. Spray-and-Wait [13] disseminates the replicas of each message in the *spray* phase. When the node has only one replica of the message, it will be in the *wait* phase, in which it waits to meet the destination and then delivers the message directly to the destination. Spray-and-Focus [14] adopts the *spray* phase in [13], and when the node has only one replica of the message, it will be in the *focus* phase, in which the message can be forwarded to its encounter with a higher utility to improve the performance. Based on Spray-and-Wait, Liu and Wu propose an optimal probabilistic forwarding protocol [15]. However, it assumes each node knows the mean inter-meeting times of all pairs of nodes in the network, which is impractical. In [16], the DTN routing approach is blackhole attacks resistent which uses the encounter tickets to secure the evidence of each contact. An encounter based routing scheme called EBR is proposed in [1], which distributes the replicas of a message between two encounters according to the proportion of their estimated EVs. However, the estimated EV in [1] is identical to all messages and independent of the TTL of each message. In this paper, we propose the EER using the expected EV which is a function of the message's TTL and is more accurate in predicting the future EV in a fixed future time interval. The simulation results show that the EER can achieve much better performance than the EBR.

The small world dynamics have been proposed for the economics and social studies, and the researchers have proved that some properties of the social network can be well utilized in the DTN routing [17]–[19]. The centrality, similarity and betweenness are borrowed from the social network to DTNs as the utility metrics in [17], based on which the proposed routing approach obtains good performance. BUBBLE [18] is a social-based forwarding algorithm using the properties of social network and it is designed for pocket switched networks. In [19], the multicast in DTNs is well studied from the social network perspective. The concept of community introduced in [18] [19] will be employed in our proposed CR, in which the property of community is used and the routing is divided into the inter-community routing and intra-community routing.

III. EXPECTED ENCOUNTER BASED ROUTING PROTOCOL

In this section, we describe the proposed expected encounter based routing protocol (EER), which adopts the link-state routing [10] and is one of the quota-based routing protocols. The EER includes two phases: the multiple replicas distribution and the single replica forwarding. We will first describe the multiple replicas distribution phase and the single replica forwarding phase in detail respectively, after which we will elaborate the EER algorithm.

A. Multiple Replicas Distribution Phase

In the EER, each message in the network is initiated with a predefined number of replicas. In the multiple replicas distribution phase, a node holds more than one replica of a message. To achieve a high message delivery ratio, the node can disseminate the replicas to different nodes as soon as possible. Therefore, when a node encounters any other node, it splits the replicas between them proportionally according to their expected EVs in a fixed future time interval, which can be calculated based on their contact histories.

1) Expected Encounter Value: As the previous work [10] [20] has shown that the mobility observations can make predictions with a very high accuracy, each node can make a prediction based on its previous contact history. According to the node's contact history, it can predict its future contact information between itself and any other node. One of such contact information is the expected EV, i.e., the number of nodes a node expects to meet, which will be used in the replicas distribution.

To calculate the expected EV, each node needs to record the encounter time of each contact between itself and any other node. Assume that there are total n nodes in the network. Each node maintains a set of sliding windows to record the contact histories, e.g., the past meeting intervals between itself and any other encountering node. The set of recorded past meeting intervals between nodes u_i and u_j is $R_{ij} = \{\Delta t_1^{ij}, \Delta t_2^{ij}, ..., \Delta t_{r_{ij}}^{ij}\}$, where Δt_k^{ij} is the recorded past k^{th} meeting interval between u_i and u_j, and r_{ij} is the total number of recorded meeting intervals between u_i and u_j. The last contact between u_i and u_j occurred at time t_0^{ij}. Then, u_i can calculate its expected EV using Theorem 1.

Theorem 1. *At time t ($t \geq t_0^{ij}$), the expected encounter value of node u_i within $(t, t+\tau]$ is:*

$$EEV_i(t,\tau) = \sum_{1 \leq j \leq n, j \neq i} \frac{m_{ij}^{\tau}}{m_{ij}}, \tag{1}$$

where $M_{ij} = \{\Delta t_k^{ij} | \Delta t_k^{ij} \in R_{ij}, \Delta t_k^{ij} > t - t_0^{ij}\}$ and $m_{ij} = |M_{ij}|$, $M_{ij}^{\tau} = \{\Delta t_k^{ij} | \Delta t_k^{ij} \in M_{ij}, \Delta t_k^{ij} \leq t + \tau - t_0^{ij}\}$ and $m_{ij}^{\tau} = |M_{ij}^{\tau}|$.

Proof: Please see Appendix A for a proof. ■

According to Theorem 1, each node in the network can calculate its expected EV when it meets any other node.

2) Replicas Distribution: In the EER, each message is initiated with a predefined number of replicas in the network. Assume that the initial number of replicas of each message is λ. A message is considered to be successfully delivered if at least one replica arrives at the destination within the TTL of the message. Thus, to obtain a high message delivery ratio, an effective strategy is to disseminate the λ replicas of each message to λ different nodes firstly, and then let each of the λ different nodes deliver the single-copy message to the destination respectively.

As each node can calculate its expected EV in a fixed future time interval based on its contact history, when two nodes meet, the distribution of the replicas of each message can be conducted according to the proportion of their expected EVs. For example, if u_i holds a message m_k with M_k replicas ($M_k > 1$), and its current TTL is TTL$_k$, u_j has no replica of m_k. When u_i meets u_j at time t, after exchanging and updating the routing information, u_i will pass

$$\lfloor M_k \cdot \frac{EEV_j(t, \alpha \cdot TTL_k)}{EEV_i(t, \alpha \cdot TTL_k) + EEV_j(t, \alpha \cdot TTL_k)} \rfloor$$

replicas of message m_k to u_j, here α is a network parameter and $0 \leq \alpha \leq 1$. That is, u_i and u_j will distribute the M_k replicas of m_k according to the proportion of their expected EVs in $(t, t + \alpha \cdot TTL_k]$.

B. Single Replica Forwarding Phase

In the EER, each message initially has λ replicas. In the multiple replicas distribution phase, each node disseminates all the replicas of the message to different nodes as soon as possible. When the number of replicas of the message in one node reduces to 1, the single replica forwarding phase starts.

In the single replica forwarding phase, each node needs to decide whether or not to forward the message it holds to its current encounter. Previous research has shown that using the contact history can make the prediction of the meeting delays to other nodes with a high accuracy [10], which is useful in making a routing decision. Thus, each node can firstly take advantage of its contact history to predict the one-hop meeting delays to other nodes, and then estimate the multi-hop meeting delay, which is the minimum expected meeting delay (MEMD) to the destination. Finally, the node can decide whether to forward the message it holds to its current encounter by comparing their MEMDs to the destination.

1) One-Hop Meeting Delay Prediction: The one-hop meeting delay can be calculated based on the past contact information. In the previous work [10], the average meeting interval between two encounters is used as their expected meeting delay. For example, if node u_i has a set of recorded past meeting intervals $\{\Delta t_1^{ij}, \Delta t_2^{ij}, ..., \Delta t_{r_{ij}}^{ij}\}$ to node u_j. Then at any moment before u_i encounters u_j, it will predict the expected meeting delay to u_j as $\frac{1}{r_{ij}} \sum_{k=1}^{r_{ij}} \Delta t_k^{ij}$. However, this average meeting interval is not always appropriate to be the prediction of the meeting delay. For instance, if two nodes periodically meet every Δt, and the last moment these two nodes meet is t_0^{ij}, then at $t_0^{ij} + \frac{1}{2}\Delta t$, the expected meeting delay between these two nodes should be $\frac{1}{2}\Delta t$, but not the average meeting interval Δt. Thus, the elapsed time since last contact between two nodes does impact their expected meeting delay.

We can use Theorem 2 to calculate the expected meeting delay (EMD) between two nodes which is related to the elapsed time since their last contact.

Theorem 2. *At time t ($t \geq t_0^{ij}$), the expected meeting delay (EMD) between nodes u_i and u_j is:*

$$EMD_{ij}(t) = \frac{1}{m_{ij}} \sum_{\Delta t_k^{ij} \in M_{ij}} \Delta t_k^{ij} - (t - t_0^{ij}), \qquad (2)$$

where $M_{ij} = \{\Delta t_k^{ij} | \Delta t_k^{ij} \in R_{ij}, \Delta t_k^{ij} > t - t_0^{ij}\}$, and $m_{ij} = |M_{ij}|$.

Proof: Please see Appendix B for a proof. ■

According to Eq. 2, each node in the network can predict the one-hop meeting delays between itself and other nodes. However, the one-hop meeting delay only includes partial connectivity information of the network. The global network connectivity information is more useful for the message delivery in the single replica forwarding.

2) Multi-Hop Meeting Delay Prediction: To make the message efficiently delivered to its destination in a DTN, each node can estimate the multi-hop meeting delay from itself to the destination, which is used to determine whether to forward the message to the current encounter. Before calculating the multi-hop meeting delay, each node can make its one-hop meeting delay prediction and exchange it with other encounters, through which it can get the network connectivity information. In the EER, each node maintains an $n \times n$ meeting interval matrix MI. The MI is defined as:

$$\mathbf{MI} = \begin{pmatrix} 0 & I_{12} & \dots & I_{1n} \\ I_{21} & 0 & \dots & I_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ I_{n1} & I_{n2} & \dots & 0 \end{pmatrix},$$

where I_{ij} denotes the average meeting interval between nodes u_i and u_j and it is updated by u_i. Obviously, $I_{ij} = 0$ when $i = j$. For the MI of u_i, I_{ij} ($1 \leq j \leq n, j \neq i$) can be obtained by $I_{ij} = \frac{1}{r_{ij}} \sum_{k=1}^{r_{ij}} \Delta t_k^{ij}$. The other elements in the MI of u_i can be obtained via information exchange when it meets other nodes. For the convenience of exchanging the meeting interval information when two nodes meet, each node has to maintain the last update time for each row in its MI. When two nodes meet at another time, they will exchange and update their MIs with each other according to the last update time of each row.[1]

As mentioned above, the prediction of the meeting delay based on the average meeting intervals may not be always

---

[1] In the implementation of our protocols, only the rows with the fresher update time need to be exchanged between the two encountering nodes, which can reduce the routing information exchange overhead greatly.

accurate. We let each node build an $n \times n$ expected meeting delay matrix MD whenever it meets another node, where

$$\mathbf{MD} = \begin{pmatrix} 0 & D_{12} & \dots & D_{1n} \\ D_{21} & 0 & \dots & D_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ D_{n1} & D_{n2} & \dots & 0 \end{pmatrix}.$$

D_{ij} is the expected meeting delay (EMD) between nodes u_i and u_j, which is updated by u_i. Also, $D_{ij} = 0$ when $i = j$. In the MD of u_i, D_{ij} $(1 \leq j \leq n, j \neq i)$ can be obtained by Eq. 2. As it is difficult for u_i to get the EMD between u_j and u_k when $j \neq i$ and $k \neq i$, u_i can replace it with I_{jk} for simplicity, which can be acquired from its MI. After building the MD, u_i can calculate the multi-hop meeting delay from itself to the destination of the message using the Dijkstra's algorithm.

Theorem 3. *The multi-hop meeting delay calculated using the Dijkstra's algorithm based on the above MD is the minimum expected meeting delay (MEMD).*

Proof: Please see Appendix C for a proof. ■

C. Expected Encounter Based Routing Algorithm

Algorithm 1 Expected Encounter Based Routing Algorithm

```
Let $m_1, m_2, \dots, m_M$ be the messages in $u_i$'s local buffer.
if $u_i$ meets $u_j$ at $t$ then
    $u_i$ and $u_j$ update their contact histories and calculate the up-to-date average meeting interval.
    $u_i$ and $u_j$ exchange their $MIs$ with each other to form an identical $MI$.
    $u_i$ and $u_j$ build their $MDs$.
    for $k = 1, 2, \dots, M$ do
        if $u_j$ does not hold $m_k$ then
            $M_k \leftarrow m_k.numOfReplicas$
            if $M_k > 1$ then
                $u_i$ sends $\lfloor M_k \cdot \frac{EEV_j(t,\alpha\cdot TTL_k)}{EEV_i(t,\alpha\cdot TTL_k)+EEV_j(t,\alpha\cdot TTL_k)} \rfloor$ replicas of $m_k$ to $u_j$.
            else
                $u_d \leftarrow m_k.destination$
                if $MEMD(u_i, u_d) > MEMD(u_j, u_d)$ then
                    $u_i$ forwards $m_k$ to $u_j$.
                end if
            end if
        end if
    end for
end if
```

The procedure of the expected encounter based routing algorithm is described in Algorithm 1. When nodes u_i and u_j meet, they update their contact histories for the meeting intervals between them and calculate the up-to-date average meeting interval. Then u_i and u_j exchange their MIs with each other to form an identical MI. If either one of these two nodes has a message to be delivered, each of them will build a new MD based on its MI and the one-hop meeting delay prediction. For each message m_k which is held by u_i but not u_j, if u_i has M_k replicas of m_k $(M_k > 1)$, it will send $\lfloor M_k \cdot \frac{EEV_j(t,\alpha\cdot TTL_k)}{EEV_i(t,\alpha\cdot TTL_k)+EEV_j(t,\alpha\cdot TTL_k)} \rfloor$ replicas of m_k to u_j, and keep the rest replicas of m_k. Otherwise, if u_i has only one replica of m_k, it compares its MEMD to the destination with that of u_j, i.e., it compares $MEMD(u_i, u_d)$ with $MEMD(u_j, u_d)$ where $MEMD(u_*, u_d)$ denotes the minimum expected meeting delay from u_* to u_d. If u_i has a longer MEMD, it will forward m_k to u_j. Note that the replicas of each message will not be redistributed between two encounters if both of them have at least one replica of this message.

IV. COMMUNITY BASED ROUTING PROTOCOL

In the EER, each node maintains an MI which includes the global network connectivity information. When a pair of nodes meet, they will exchange and update their MIs which may cause some routing information exchange overhead. In this section, we will propose the community based routing protocol (CR) which employs the concept of social network and can further reduce the information exchange overhead by diminishing the scale of the MIs. We will first introduce the concept of community. Then we will describe the calculation of expected number of encountering communities for each node in a fixed future time interval. Finally, we will elaborate the community based routing algorithm.

A. Community — A Social Network Concept

A social network is a structured human society which consists of individuals (called nodes) connected by socially meaningful relationships, such as common interest or social relations. Such social relationships can also partition the social network into several communities naturally. Community is an important attribute of a social network. Generally, the social relationship within the same community is stronger than that between different communities. For instance, the contact frequency between a pair of nodes in the same community is much higher than that from different communities. More specifically, for example, all the students in a school are divided into different classes (i.e., communities). The students from the same class will meet with each other frequently as they are classmates and they attend similar classes together. On the other hand, the meeting frequency between the students from different classes will be much lower.

The concept of community can be used in the DTN routing. In a DTN, all the nodes are divided into several communities according to their relationships. Then the routing in the DTN can be conducted in two phases — inter-community routing and intra-community routing. In the inter-community routing, each node distributes the multiple replicas of a message to the nodes from different communities as soon as possible, which can be achieved by distributing the replicas of the message according to the proportion of the two encounters' expected numbers of encountering communities. In case of the single replica of the message, it will be forwarded to the node which has a higher probability to encounter the destination community (i.e., the community which the destination of the message belongs to). In the intra-community routing, a node in the destination community distributes the replicas of a message to its encounter in the same community according to the proportion of their intra-community expected EVs, which are calculated based on the nodes only in the same community. Note that the intra-community MEMD, intra-community MI and intra-community MD that will be discussed in the following sections are also calculated based on the nodes only in the same community. In case of the single replica of the message, the node in the destination community decides whether to forward the message to its encounter in the same community by comparing their intra-community MEMDs.

There are a lot of research work on the construction of community, including the centralized algorithms, such as the *k*-clique [21] and weighted network analysis (WNA) [22], and distributed algorithms such as the construction method in [23]. While the construction of community is not our focus in this paper, we take advantages of the community property and propose the community based routing protocol (CR).[2]

B. Expected Number of Encountering Communities

In a community based DTN, each node maintains the intra-community *MI* and *MD*. In addition, each node also needs to maintain $n-1$ sliding windows to record contact histories, i.e., the past meeting intervals between itself and any other $n-1$ nodes. The node can use the recorded contract histories to calculate its expected number of encountering communities in a fixed future time interval.

We assume a network is partitioned into l communities $\{C_1, C_2, \cdots, C_l\}$ and each node only belongs to one of the l communities.[3] C_k denotes the set of nodes inside the k^{th} community, and CID_{u_i} denotes the ID of the community which node u_i belongs to. u_i is considered to encounter community C_k if it meets at least one node in C_k. Then u_i can calculate its expected number of encountering communities using Theorem 4.

Theorem 4. *At time t ($t \geq t_0^{ij}$), the expected number of encountering communities for node u_i within $(t, t+\tau]$ is:*

$$ENEC_i(t,\tau) = \sum_{1 \leq k \leq l, k \neq CID_{u_i}} (1 - \prod_{u_j \in C_k} (1 - \frac{m_{ij}^{\tau}}{m_{ij}})), \qquad (3)$$

where $M_{ij} = \{\Delta t_k^{ij} | \Delta t_k^{ij} \in R_{ij}, \Delta t_k^{ij} > t - t_0^{ij}\}$ and $m_{ij} = |M_{ij}|$, $M_{ij}^{\tau} = \{\Delta t_k^{ij} | \Delta t_k^{ij} \in M_{ij}, \Delta t_k^{ij} \leq t + \tau - t_0^{ij}\}$ and $m_{ij}^{\tau} = |M_{ij}^{\tau}|$.

Proof: Please see Appendix D for a proof. ■

Based on Theorem 4, each node in the network can calculate its expected number of encountering communities when it meets any other node.

C. Community Based Routing Algorithm

In the community based routing protocol, every node has a global unique ID and a community ID. When a message is generated, its destination u_d is attached in this message together with the community ID CID_{u_d}.

Algorithm 2 Community Based Routing Algorithm

```
Let $m_1, m_2, \ldots, m_M$ be the messages in $u_i$'s local buffer.
if $u_i$ meets $u_j$ at $t$ then
    $u_i$ and $u_j$ update their contact histories and calculate the up-to-date average meeting interval.
    for $k = 1, 2, \ldots, M$ do
        $u_d \leftarrow m_k.destination$
        if $CID_{u_i} \neq CID_{u_d}$ then
            Trigger Inter-Community Routing Algorithm.
        else
            Trigger Intra-Community Routing Algorithm.
        end if
    end for
end if
```

Algorithm 3 Inter-Community Routing Algorithm

```
if $CID_{u_j} = CID_{u_d}$ then
    $u_i$ sends all replicas of $m_k$ to $u_j$.
else
    if $u_j$ does not hold $m_k$ then
        $M_k \leftarrow m_k.numOfReplicas$
        if $M_k > 1$ then
            $u_i$ sends $\lfloor M_k \cdot \frac{ENEC_j(t,\alpha \cdot TTL_k)}{ENEC_i(t,\alpha \cdot TTL_k)+ENEC_j(t,\alpha \cdot TTL_k)} \rfloor$ replicas of $m_k$ to $u_j$.
        else
            $c \leftarrow CID_{u_d}$
            if $P_{ic} < P_{jc}$ then
                $u_i$ forwards $m_k$ to $u_j$.
            end if
        end if
    end if
end if
```

The community based routing algorithm is shown in Algorithm 2. When nodes u_i and u_j meet, they update

---

[2] In the implementation of the CR, the communities in the network are predefined for simplicity.

[3] We only consider that one node belongs to one community in this paper for simplicity of description. It is noted that our proposed protocol can work well even when one node belongs to multiple communities.

Algorithm 4 Intra-Community Routing Algorithm

```
if $CID_{u_i} = CID_{u_j}$ then
    $u_i$ and $u_j$ exchange their intra-community $MIs$ with each other to form an identical intra-community $MI$.
    $u_i$ and $u_j$ build their intra-community $MDs$.
    if $u_j$ does not hold $m_k$ then
        $M_k \leftarrow m_k.numOfReplicas$
        if $M_k > 1$ then
            $u_i$ sends $\lfloor M_k \cdot \frac{EEV_j(t,\alpha \cdot TTL_k)'}{EEV_i(t,\alpha \cdot TTL_k)' + EEV_j(t,\alpha \cdot TTL_k)'} \rfloor$ replicas of $m_k$ to $u_j$.
        else
            if $MEMD(u_i, u_d)' > MEMD(u_j, u_d)'$ then
                $u_i$ forwards $m_k$ to $u_j$.
            end if
        end if
    end if
end if
```

their contact histories and calculate the up-to-date average meeting interval. For each message m_k which is held by u_i, if the destination of m_k is not within the same community of u_i, the inter-community routing algorithm will be triggered. Otherwise, the intra-community routing algorithm will be triggered.

The inter-community and intra-community routing algorithms are described in Algorithm 3 and Algorithm 4 respectively. In the inter-community routing algorithm, u_i tries to deliver m_k to the destination community. If u_j belongs to the destination community, u_i will send all the replicas of m_k to u_j. Otherwise, u_i will continue the process: If m_k is held by u_i but not u_j, u_i will make the routing decision according to M_k (the number of replicas of m_k). If M_k is larger than 1, u_i will distribute the replicas of m_k between itself and u_j according to the proportion of their expected numbers of encountering communities. u_i will send $\lfloor M_k \cdot \frac{ENEC_j(t,\alpha \cdot TTL_k)}{ENEC_i(t,\alpha \cdot TTL_k) + ENEC_j(t,\alpha \cdot TTL_k)} \rfloor$ replicas of m_k to u_j. Otherwise, if u_i has only one replica of m_k, and if the probability that u_i will encounter the destination community in $(t, t + \alpha \cdot TTL_k]$ is less than that of u_j, u_i will forward m_k to u_j.

In the intra-community routing algorithm, if u_i and u_j belong to different communities, u_i will not send m_k to u_j as u_j is outside the destination community. Otherwise, nodes u_i and u_j will exchange their intra-community MIs to form an identical intra-community MI and build their new intra-community MDs. If m_k is held by node u_i but not u_j, u_i will make the routing decision according to the number of replicas of m_k. If u_i has more than one replica of m_k, it will send $\lfloor M_k \cdot \frac{EEV_j(t,\alpha \cdot TTL_k)'}{EEV_i(t,\alpha \cdot TTL_k)' + EEV_j(t,\alpha \cdot TTL_k)'} \rfloor$ replicas of m_k to u_j, where $EEV_i(t, \alpha \cdot TTL_k)'$ represents the intra-community expected EV of u_i in $(t, t + \alpha \cdot TTL_k]$. If u_i has only one replica of m_k and the $MEMD'$ (intra-community $MEMD$) from u_i to u_d is larger than that from u_j to u_d, u_i will forward m_k to u_j.

V. PERFORMANCE EVALUATION

In this section, we will evaluate our proposed routing protocols with three performance metrics, delivery ratio, latency and goodput, in the Opportunistic Network Environment simulator (ONE) [24]. We also compare our proposed routing protocols with other existing popular DTN routing protocols.

A. Performance Metrics and Simulation Settings

Three metrics will be employed in the performance evaluation, including delivery ratio, latency and goodput. The major goal of the DTN routing is to achieve a high delivery ratio and goodput with a low latency. The definitions of these three metrics are shown as follows:

- *Delivery ratio:* The ratio of the number of delivered messages to the number of all the generated messages.
- *Latency:* The average end-to-end delivery delay between each pair of source and destination in the network.
- *Goodput:* The ratio of the number of delivered messages to the total number of relayed messages in the network.

To evaluate the performance of our proposed DTN routing protocols, we use the vehicular-based map-driven model in our simulations, which is part of the ONE simulator. Some bus lines based on the map of downtown Helsink in Finland are employed into the simulations, and the buses which travel along the bus lines represent the nodes in the network.

We use the following settings in the simulations: the moving speed of the nodes varies from 2.7 to 13.9 m/s, the simulation update interval is $0.1s$, the transmission speed is 2 Mbps and the transmission range is 10 m. The buffer space of each node is 1 MB, and the size of each packet is 25 KB. The network parameter α is set to 0.28, which is indicated to be a reasonable value from the preliminary simulations. Each simulation lasts for $10000s$, and the TTL of each message is 20 minutes. The number of nodes in the network varies from 40 to 240 with an increment of 40. The value of each point in the curves is the average of 10 simulation runs.

B. Simulation Results

To evaluate the effectiveness of our proposed DTN routing protocols, we compare the EER and CR with other four popular protocols: EBR [1], MaxProp [7], Spray-and-Wait [13] and Spray-and-Focus [14]. After that we analyze the effects of λ, which is the initial value of replicas of a message, on the performance of the EER and CR respectively. Due to the space limitation, we do not analyze the effects of other parameters such as α, TTL and buffer size on the performance.

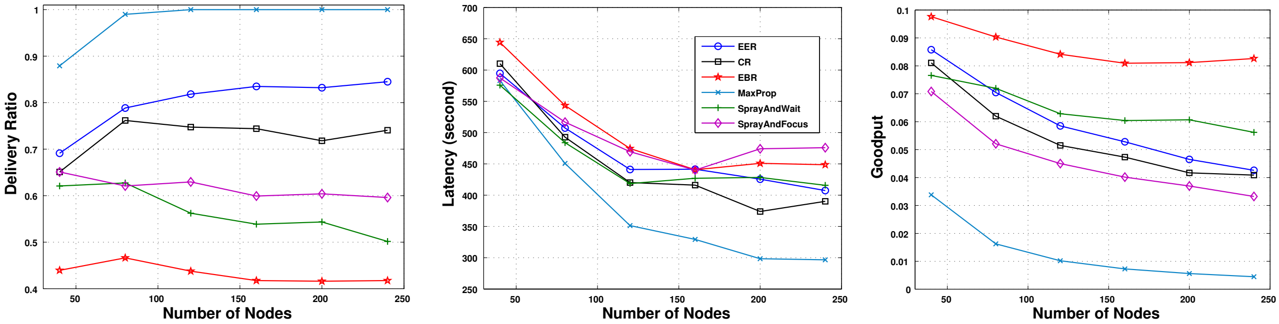


Figure 2. Performance comparison between our proposed routing protocols and other existing protocols: (a) Delivery ratio; (b) Latency; (c) Goodput.

Figure 2 shows the performance comparison between the EER, CR and other four protocols. We set λ as 10 for the EER, CR, Spray-and-Wait and Spray-and-Focus protocols. The figure shows that MaxProp achieves the highest delivery ratio, the shortest latency since it is an epidemic-based protocol. However, the goodput of the MaxProp is also the lowest, which is only about 20% of those of the EER and CR protocols. Thus, the MaxProp fails in the comparison due to its poor goodput. The EBR obtains the best goodput, but its delivery ratio is the lowest and its latency is almost the highest. The goodput of Spray-and-Wait exceeds those of EER and CR when the number of nodes is larger than 80. However, its delivery ratio is much lower than EER and CR, and its latency is comparative to EER and CR. The Spray-and-Focus acquires a lower delivery ratio than EER and CR, with a higher latency and a lower goodput. Consequently, our proposed protocols EER and CR perform effectively compared with other four protocols.

Figure 3 and Figure 4 illustrate the effects of λ on the performance of EER and CR respectively. The value of λ varies from 6 to 12 with an increment of 2. The delivery ratio of both protocols rises when the value of λ increases. The increase of λ can slightly reduce the latency (obvious in Figure 3(b)). However, the increase of λ can heighten the overhead because a larger number of forwards will be employed in the network, and the EER and CR will achieve a lower goodput. Therefore, it is a tradeoff to determine an appropriate value of λ.

VI. CONCLUSION AND FUTURE WORK

In this paper, we first propose an expected encounter based routing protocol (EER) which distributes multiple replicas of a message proportionally between two encounters according to their expected EVs. In case of single replica of a message, EER makes the routing decision by comparing the minimum expected meeting delay to the destination. To take advantages of the community property, we further propose a community based routing protocol (CR), which is divided into inter-community routing and intra-community routing. We evaluate our proposed routing protocols in the ONE simulator under different parameters to demonstrate their effectiveness.

One of our future directions will focus on extending the proposed routing protocols to be applicable to resource-constrained wireless networks by employing the buffer management. Secondly, we will design the distributed community construction method in the CR, which is more suitable for the online routing procedure. Finally, we intend to design adaptive routing protocols in which the network parameters such as α and λ can be tuned automatically to improve the performance.

ACKNOWLEDGMENT

This work is supported in part by grants PolyU 5253/09E, PolyU 5221/11E, 1-ZV5N and ZJU-SKL ICT0903.

REFERENCES

[1] S. C. Nelson, M. Bakht, and R. Kravets, “Encounter-Based Routing in DTNs,” in *Proc. of IEEE INFOCOM*, 2009.

[2] A. Lindgren, A. Doria, and O. Schelen, “Probabilistic Routing in Intermittently Connected Networks,” in *Proc. of ACM MobiHoc*, 2003.

[3] A. Balasubramanian, B. N. Levine, and A. Venkataramani, “DTN routing as a resource allocation problem,” in *Proc. of ACM SIGCOMM*, 2007.

[4] T. Spyropoulos, K. Psounis, and C. S. Raghavendra, “Single-copy routing in intermittently connected mobile networks,” in *Proc. of IEEE SECON*, 2004.

[5] X. Zhang, J. F. Kurose, B. Levine, D. Towsley, and H. Zhang, “Study of a Bus-Based Disruption Tolerant Network: Mobility Modeling and Impact on Routing,” in *Proc. of ACM MobiCom*, 2007.

[6] A. Vahdat and D. Becker, “Epidemic routing for partially connected ad hoc networks,” in *Technical Report CS-200006*, Duke University, April 2000.

[7] J. Burgess, B. Gallagher, D. Jensen, and B. N. Levine, “MaxProp: Routing for Vehicle-Based Disruption-Tolerant Networking,” in *Proc. of IEEE INFOCOM*, 2006.

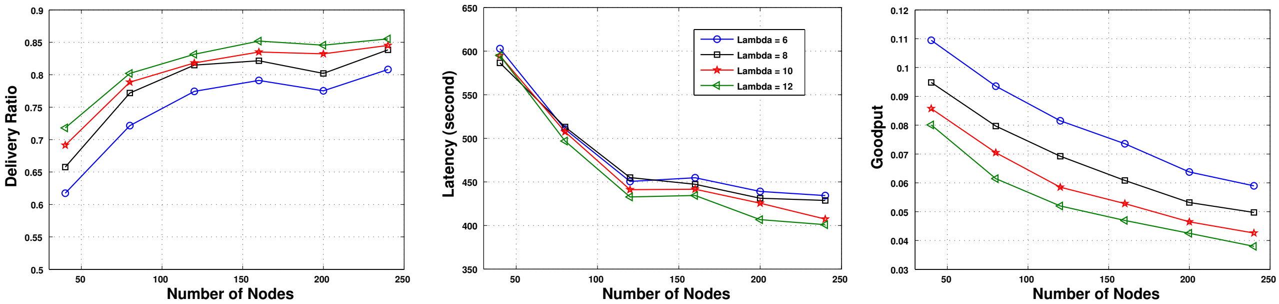


Figure 3. Effects of the value of λ on the performance of the EER: (a) Delivery ratio; (b) Latency; (c) Goodput.

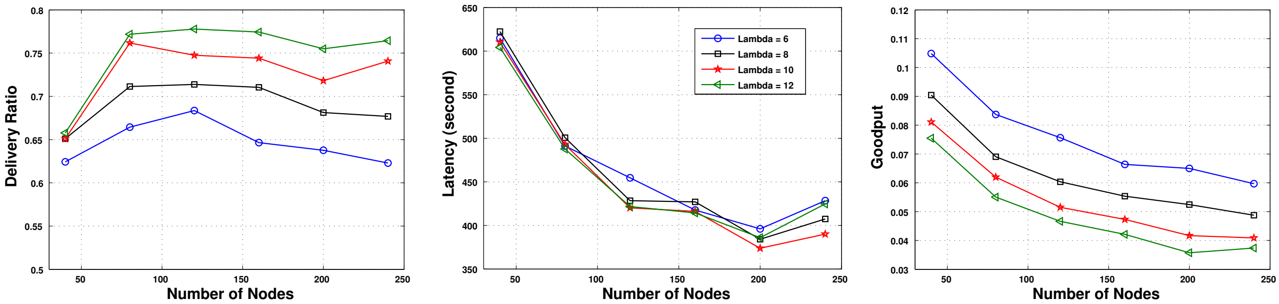


Figure 4. Effects of the value of λ on the performance of the CR: (a) Delivery ratio; (b) Latency; (c) Goodput.

[8] V. Erramilli, M. Crovella, A. Chaintreau, and C. Diot, “Delegation Forwarding,” in *Proc. of ACM MobiHoc*, 2008.

[9] S. Jain, K. Fall, and R. Patra, “Routing in a Delay Tolerant Network,” in *Proc. of ACM SIGCOMM*, 2004.

[10] E. P. C. Jones, L. Li, and P. A. S. Ward, “Practical Routing in Delay-Tolerant Networks,” in *Proc. of ACM WTDN*, 2005.

[11] Q. Yuan and J. Wu, “Predict and Relay: An Efficient Routing in Disruption-Tolerant Networks,” in *Proc. of ACM MobiHoc*, 2009.

[12] W. Gao and G. Cao, “Exploiting Transient Contact Patterns for Data Forwarding in Delay Tolerant Network,” in *Proc. of IEEE ICNP*, 2010.

[13] T. Spyropoulos, K. Psounis, and C. Raghavendra, “Spray and Wait: An Efficient Routing Scheme for Intermitently Connected Mobile Networks,” in *Proc. of ACM WDTN*, 2005.

[14] ——, “Spray and Focus: Efficient Mobility-Assisted Routing for Heterogeneous and Correlated Mobility,” in *Proc. of PerCom Workshops*, 2007.

[15] C. Liu and J. Wu, “An Optimal Probabilistic Forwarding Protocol in Delay Tolerant Networks,” in *Proc. of ACM MobiHoc*, 2009.

[16] F. Li, J. Wu, and A. Srinivasan, “Thwarting Blackhole Attacks in Disruption-Tolerant Networks using Encounter Tickets,” in *Proc. of IEEE INFOCOM*, 2009.

[17] E. Daly and M. Haahr, “Social Network Analysis for Routing in Disconnected Delay-Tolerant MANETs,” in *Proc. of ACM MobiHoc*, 2007.

[18] P. Hui, J. Crowcroft, and E. Yoneki, “BUBBLE Rap: Social-based Forwarding in Delay Tolerant Networks,” in *Proc. of ACM MobiHoc*, 2008.

[19] W. Gao, Q. Li, B. Zhao, and G. Cao, “Multicasting in Delay Tolerant Networks: A Social Network Perspective,” in *Proc. of ACM MobiHoc*, 2009.

[20] L. Song, D. Kotz, R. Jain, and X. He, “Evaluation location predictors with extensive Wi-Fi mobility data,” in *Proc. of IEEE INFOCOM*, 2004.

[21] G. Palla, I. Derenyi, I. Farkas, and T. Vicsek, “Uncovering the overlapping community structure of complex networks in nature and society,” *Nature*, 2005.

[22] M. E. J. Newman, “Analysis of weighted networks,” *Physical Review E*, 2004.

[23] A. Clauset, “Finding local community structure in networks,” *Physical Review E*, 2005.

[24] A. Keranen, J. Ott, and T. Karkkainen, “The ONE Simulator for DTN Protocol Evaluation,” in *Proc. of the 2nd International Conference on Simulation Tools and Techniques (SIMUTools)*, 2009.

APPENDIX

A. Proof of Theorem 1

Proof: Assume that the next meeting interval between u_i and u_j is Δt^{ij}, then the probability that u_i will meet u_j in $(t, t+\tau]$ is $P(\Delta t^{ij} \le t+\tau-t_0^{ij} | \Delta t^{ij} > t-t_0^{ij})$. Thus,

$$EEV_i(t,\tau) = \sum_{1\le j\le n,\ j\ne i} P(\Delta t^{ij} \le t+\tau-t_0^{ij} | \Delta t^{ij} > t-t_0^{ij}).$$

Here,

$$P(\Delta t^{ij} \le t+\tau-t_0^{ij} | \Delta t^{ij} > t-t_0^{ij}) = \frac{P(t-t_0^{ij} < \Delta t^{ij} \le t+\tau-t_0^{ij})}{P(\Delta t^{ij} > t-t_0^{ij})}.$$

Considering $m_{ij} = |M_{ij}|$ where $M_{ij} = \{\Delta t_k^{ij} | \Delta t_k^{ij} \in R_{ij}, \Delta t_k^{ij} > t-t_0^{ij}\}$, and $m_{ij}^\tau = |M_{ij}^\tau|$ where $M_{ij}^\tau = \{\Delta t_k^{ij} | \Delta t_k^{ij} \in M_{ij}, \Delta t_k^{ij} \le t+\tau-t_0^{ij}\}$, we can get

$$P(\Delta t^{ij} > t-t_0^{ij}) = \sum_{\Delta t_k^{ij} \in M_{ij}} \frac{1}{r_{ij}} = \frac{m_{ij}}{r_{ij}},$$

and

$$P(t-t_0^{ij} < \Delta t^{ij} \le t+\tau-t_0^{ij}) = \sum_{\Delta t_k^{ij} \in M_{ij}^\tau} \frac{1}{r_{ij}} = \frac{m_{ij}^\tau}{r_{ij}}.$$

So,

$$P(\Delta t^{ij} \le t+\tau-t_0^{ij} | \Delta t^{ij} > t-t_0^{ij}) = \frac{m_{ij}^\tau / r_{ij}}{m_{ij} / r_{ij}} = \frac{m_{ij}^\tau}{m_{ij}}. \tag{4}$$

Therefore, we can obtain

$$EEV_i(t,\tau) = \sum_{1\le j\le n, j\ne i} \frac{m_{ij}^\tau}{m_{ij}}.$$

■

B. Proof of Theorem 2

Proof: Assume that the next meeting interval between nodes u_i and u_j is Δt^{ij}, thus,

$$\begin{aligned} EMD_{ij}(t) &= E(\Delta t^{ij} - (t-t_0^{ij}) | \Delta t^{ij} > t-t_0^{ij}) \\ &= E(\Delta t^{ij} | \Delta t^{ij} > t-t_0^{ij}) - E(t-t_0^{ij} | \Delta t^{ij} > t-t_0^{ij}) \\ &= E(\Delta t^{ij} | \Delta t^{ij} > t-t_0^{ij}) - (t-t_0^{ij}). \end{aligned}$$

$E(\Delta t^{ij} | \Delta t^{ij} > t-t_0^{ij})$ can be calculated as:

$$\begin{aligned} E(\Delta t^{ij} | \Delta t^{ij} > t-t_0^{ij}) &= \sum_{k=1}^{r_{ij}} P(\Delta t^{ij} = \Delta t_k^{ij} | \Delta t^{ij} > t-t_0^{ij}) \cdot \Delta t_k^{ij} \\ &= \sum_{k=1}^{r_{ij}} \frac{P(\Delta t^{ij} = \Delta t_k^{ij}, \Delta t^{ij} > t-t_0^{ij})}{P(\Delta t^{ij} > t-t_0^{ij})} \cdot \Delta t_k^{ij}. \end{aligned}$$

Here,

$$P(\Delta t^{ij} = \Delta t_k^{ij}, \Delta t^{ij} > t-t_0^{ij}) = \begin{cases} \frac{1}{r_{ij}} & \text{if } \Delta t_k^{ij} > t-t_0^{ij}, \\ 0 & \text{else.} \end{cases}$$

and

$$P(\Delta t^{ij} > t-t_0^{ij}) = \sum_{\Delta t_k^{ij} \in M_{ij}} \frac{1}{r_{ij}} = \frac{m_{ij}}{r_{ij}}.$$

We can get:

$$E(\Delta t^{ij} | \Delta t^{ij} > t-t_0^{ij}) = \frac{r_{ij}}{m_{ij}} \sum_{\Delta t_k^{ij} \in M_{ij}} \frac{1}{r_{ij}} \cdot \Delta t_k^{ij} = \frac{1}{m_{ij}} \sum_{\Delta t_k^{ij} \in M_{ij}} \Delta t_k^{ij}.$$

Therefore,

$$\begin{aligned} EMD_{ij}(t) &= E(\Delta t^{ij} | \Delta t^{ij} > t-t_0^{ij}) - (t-t_0^{ij}) \\ &= \frac{1}{m_{ij}} \sum_{\Delta t_k^{ij} \in M_{ij}} \Delta t_k^{ij} - (t-t_0^{ij}). \end{aligned}$$

■

C. Proof of Theorem 3

Proof: As each element in MD indicates the EMD between a pair of nodes, it is easy to see that the calculated multi-hop meeting delay from the node to a particular destination using the Dijkstra's algorithm is the MEMD between the node and the destination. ■

D. Proof of Theorem 4

Proof: Assume that the next meeting interval between nodes u_i and u_j is Δt^{ij}, and the probability that u_i will encounter community C_k in $(t, t+\tau]$ is P_{ik}. The expected number of encountering communities for u_i in $(t, t+\tau]$ is:

$$ENEC_i(t,\tau) = \sum_{1\le k\le l, k\ne CID_{u_i}} P_{ik}.$$

P_{ik} can be calculated as:

$$P_{ik} = 1 - \prod_{u_j \in C_k} (1 - P(\Delta t^{ij} \le t+\tau-t_0^{ij} | \Delta t^{ij} > t-t_0^{ij})).$$

In Eq. 4 of Theorem 1, we have already got that

$$P(\Delta t^{ij} \le t+\tau-t_0^{ij} | \Delta t^{ij} > t-t_0^{ij}) = \frac{m_{ij}^\tau}{m_{ij}},$$

where $M_{ij} = \{\Delta t_k^{ij} | \Delta t_k^{ij} \in R_{ij}, \Delta t_k^{ij} > t-t_0^{ij}\}$ and $m_{ij} = |M_{ij}|$, $M_{ij}^\tau = \{\Delta t_k^{ij} | \Delta t_k^{ij} \in M_{ij}, \Delta t_k^{ij} \le t+\tau-t_0^{ij}\}$ and $m_{ij}^\tau = |M_{ij}^\tau|$.

Thus,

$$ENEC_i(t,\tau) = \sum_{1\le k\le l, k\ne CID_{u_i}} P_{ik} = \sum_{1\le k\le l, k\ne CID_{u_i}} (1 - \prod_{u_j \in C_k} (1 - \frac{m_{ij}^\tau}{m_{ij}})).$$

■