Relative Interpersonal-Influence-Aware Routing in Buffer Constrained Delay-Tolerant Networks

Ziyun Zhu⋆, Sheng Liu⋆, Suguo Du⋆, Xiaodong Lin†, Haojin Zhu⋆
⋆Shanghai Jiao Tong University, Shanghai, China
{bamboo220, liusheng, sgdu, zhu-hj}@sjtu.edu.cn
†University of Ontario Institute of Technology, Canada; xiaodong.lin@uoit.ca

Abstract—**In Delay-Tolerant Networks, the existence of social selfishness results in different levels of willingness to forward the packet from acquaintances and strangers. The traditional social selfishness is defined solely by the social ties between node pairs to indicate the preference for packet delivery. However, the absolute value of social ties cannot accurately model the competition among friends when the storage resources become scarce. Thus, if the social selfishness exists in the buffer management, the traditional social ties cannot reflect the willingness for saving the packet when buffer overflows. To address this issue, we take the relay's social ties with all other nodes into account and introduce the definition of interpersonal influence as a metric for the willingness to save packets in buffer management. Furthermore, based on the interpersonal influence we estimate the overall delivery probability and further propose a multi-copy routing protocol to reduce the occurrence of packet dropping. Simulation result based on real trace INFOCOM06 demonstrates the efficiency of our scheme.**

Keywords – **Delay-Tolerant Network, Social-Aware Routing, Social Selfishness**

I. INTRODUCTION

Delay-tolerant networks (DTNs) [1]–[4] are partitioned wireless ad hoc networks with intermittent connectivity. Instead of relying on end-to-end link, DTNs deliver and forward packets by utilizing temporary connection in a store-carry-and-forward manner. These networks could be useful in scenarios including sensor networks with scheduled intermittent connectivity, vehicular DTNs disseminating location-dependent information, and pocket-switched networks enabling humans to communicate without network infrastructure.

One of the fundamental obstacles arising when putting DTNs into practice is routing. Unlike traditional routing schemes in wired networks which are modeled as connected graphs where shortest path approaches are widely adopted, routing problems in DTNs must take into account the capacity of intermediate buffering and the mobility of nodes. To efficiently and successfully deliver the messages in DTNs, many routing protocols with different properties are proposed. For example, replication-based routing protocols [5]–[8] can achieve a high delivery probability; with the knowledge about network topology and social structure, forwarding-based routing protocols [3] [4] [9] are efficient in a low consumption in storage and energy.

Most of the existing DTN routing protocols are built on the assumption that intermediate nodes are ready to forward data for others. However, this hypothesis may fail to hold in front of rational nodes which would refuse to forward messages for others in order to save precious resources (e.g., energy or buffer size) [10]. Different from regarding users as rational, some researchers introduce the concept of social selfishness and consider the nodes to be social, which means that nodes are willing to forward packets for friends but refuse to do so for strangers [9] [11] [12]. Motivated by this idea, the data is given various priority according to the social closeness between the sender and relay nodes.

However, the existing work based on social closeness only considers absolute *social willingness* between node pairs or communities, which may be insufficient to describe the competition between friends in resource constrained networks. If the buffer size is unlimited, a node can absolutely save all the packets from the friends. However, in practice, buffer space is finite and valuable, which means that the packets from friends are possibly beyond the buffer constraint and dropping decision must be made among friends. In this case, the traditional social ties cannot reflect the willingness to save the packet when buffer overflows.

Moreover, current social-aware routing [3] [4] makes the buffer congestion more severe and the competition for buffer space more fierce. In traditional approaches, the data is more likely to be delivered to the central nodes which are located in the centrality of the social networks since they have more influences on other nodes and thus achieve better routing performance. However, in DTNs with limited buffer space, if each node forwards its data to the central node, the buffer size of this node will run out quickly. As a result, the central node tends to remove the packets from those unfamiliar friends but remain those from nodes in close relation, which makes the competition for buffer space more fierce.

Therefore, the question we should focus on turns from "Am I your friend that you will help me forward the packet?" to "Are we close enough that you will not drop my packet even when your buffer is full?" In order to solve this major question, we bring a concept *interpersonal influence* to describe the relative position of social relationship. In addition, based on the *interpersonal influence*, we propose a multi-copy routing protocol, which we name it "*RIIA*", to avoid the competition for buffer space.

Our contributions are twofold.

1) We extend the idea of social selfishness in DTN routing

from the current willingness to forward the packet to the future willingness to save the packet when buffer overflows. We define the interpersonal influence to describe the relative position of social relationship which is more practical in buffer constrained DTNs.

2) We design a multi-copy routing protocol in DTNs. In the assumption of social selfishness in buffer management, we model the incoming packets as Poisson process, and estimate the staying time for the packets during the competition for buffer space. Based on staying time, delivery probability is used as a metric for choosing the best relay.

The rest of paper is organized as follows. In section II, we briefly introduce the research related to our work. In section III, we present the system model. In section IV, we show the general routing protocol. Section V gives the detailed achievement of our proposed scheme. Section VI shows the experiment result. Finally, we draw our conclusions in section VII.

II. Related Work

A. Routing Scheme

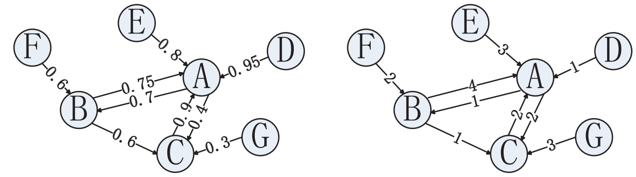
Compared to forwarding-based routing [3] [13], replication-based routing [8] is more effective and has a higher delivery rate as well as a lower delivery delay. However, the disadvantage of replication-based routing is also obvious. Plenty of replicas existing in the network are a great waste in buffer space and energy. Besides, the ideal delivery rate can never reach in practice due to the storage limitation [14]. Thus, mitigating the storage overhead is important in replication-based routing (e.g., limiting the total replicas [6] or the number of hops [7] for one message [6], copying the message with a probability [15], continuously increasing the transmission threshold [8]). After the storage overhead actually occurs, buffer management schemes [16]–[18] are proposed to drop the packet which contributes little delivery rate when the buffer is not enough.

B. Social Selfishness

Li *at al.* [9] consider the social selfishness in data forwarding, and propose a routing protocol allowing the existence of selfishness. Li *at al.* [11] evaluate the effect of social selfishness on epidemic routing. Qiu *at al.* [12] consider both contact period and inter-contact period to evaluate the effect of social selfishness. However, the previous work only consider absolute social closeness which is related to immediate drop, but ignore the relative interpersonal influence from a global view which is related to the future drop. Moreover, current detecting mechanism cannot find whether the nodes follow the buffer management policy when the buffer is full. Thus, designing a routing scheme in the assumption of existence of selfishness in buffer management is of great importance.

III. System Model

In this section, we first describe the interpersonal influence and the network model, and then introduce the architecture of our routing scheme.



(a) Social Closeness (b) Interpersonal Influence

Fig. 1. An example of calculating interpersonal influence from social closeness

A. Interpersonal Influence

We extend social ties to interpersonal influence from a global view. S_{ij} denotes the absolute social ties of node j from the perspective of node i, which can reflect the social relationship between the node pair i and j. Then for node i, we sort all its social ties in descending order from 1 to N (total number of nodes). The ranking of social ties S_{ij} is notated as I_{ij} which we call it "*Interpersonal Influence*". The value for the node itself I_{ii} is set to 0, which means that packet from itself has the highest priority compared to other nodes if it shares its buffer with other nodes.

An example of calculating the interpersonal influence is shown in Figure 1. The edge from node B to node A means the social closeness or interpersonal influence of node B from the perspective of node A. As shown in Figure 1(a), the edge represents the social closeness between node pairs and the value ranges from 0 to 1(closely related). More Specifically, for node C, the social closeness $S_{GC}(0.3)$ is smaller than $S_{BC}(0.6)$, therefore node C is more willing to deliver the packets from node B. However, social closeness cannot reflect the competition for the buffer space. For example, for node B, although the absolute social ties to node A is larger than node C, node B is less competitive in the buffer competition. In order to solve this problem, we convert the social closeness to interpersonal influence as shown in Figure 1(b). We can easily find that node B is actually more important for node C, and therefore node C should be a better relay.

Since the interpersonal influence is derived from the absolute social ties, the next question is how to define the social ties in our work. Although the frequency of collocation cannot be an ideal metric for social ties, it partially reflects the social relationships [19]. The contact frequency between node i and node j is f_{ij}, and then we use this value to represent the social ties S_{ij}. Because we only care about the relative position of the social relationship, the absolute value does not matter. Moreover, this is only used for setting the default value, the users can also change the value of all other users.

B. Network model

Node contacts in the network are described by the contact graph $G(V, E)$. And for the edge $e_{ij} \in E$ between node $N_i, N_j \in V$, the weight is defined by the contact rate between the node pair. In this paper, the distribution of inter-contact time between nodes is exponential, and each contact is Poisson distributed.

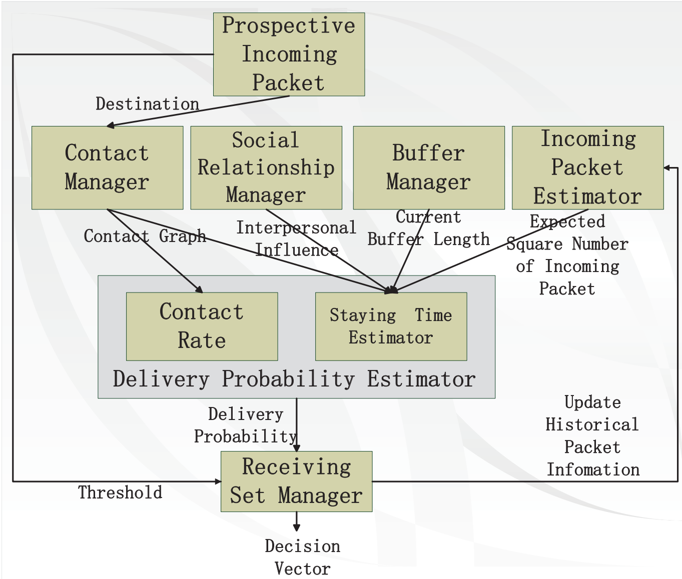


Fig. 2. Architecture

C. Architecture

The architecture of our routing scheme consists of the following components (shown in Figure 2).

1) Contact Manager: Contact manager is responsible for the estimation of contact rate. The input of it is the destination of packet and the current time; the output is the contact rate. This rate may be time-varying [4], and can utilize the transient social contact patterns for a better metric of contact rate. In addition, Similar to [9], the contact rate can be continuously updated in our work.

2) Social Relationship Manager: The default value of interpersonal influence is calculated in this component, and the users can further change this value.

3) Buffer Manager: Due to lack of detection scheme for optimal buffer management schemes, we allow users to drop those packets from users with relative lower social closeness. When the buffer overflows, this part will select those packets from the nodes with relatively lower interpersonal influence.

4) Incoming Packet Estimator: It records the historical behavior of packet transmission from every other nodes, and provides the information of incoming packet for delivery probability estimation.

5) Delivery Probability Estimator: It is responsible for the calculation of delivery probability which is the metric for routing algorithm in our scheme. Two parameters are used here: *contact rate* and *packet staying time*. The former can be directly derived from contact manager, and the latter should be estimated from contact rate, interpersonal influence, current buffer size, and the expected square number of incoming packets. The calculation of the packet staying time is given in (5).

6) Receiving Set Manager: It compares the delivery probability from delivery probability estimator with the threshold of each packet. Then a decision vector is generated from it informing another node which packet should be transmitted. At the same time, it updates the incoming packet information in *incoming packet estimator*.

7) Packet Threshold Manager: After getting which packet should be transmitted, it revises the threshold of packet to be sent, which is beneficial for reducing the total number of replicas in the network. In [8], the authors claim that the total number of packet can be $O(\sqrt{N})$ (N is the number of nodes in the network). We use the same method to update the threshold in our work.

IV. PROPOSED ROUTING PROTOCOL

Since the central nodes are more likely to receive a large volume of packets and must drop some packets when the buffer overflows, they are not a good choice for the nodes with relatively lower social ties. Thus, if we do not take into account whether the packet will be dropped and how long the packet can stay in the buffer, many packets transmitted are actually wasted. In order to solve this problem, we propose a multi-copy routing protocol to reduce the number of packets wasted based on the prediction of future dropping behavior.

The outline of deciding which packet to deliver is as follows.

1) exchange the property of packet;
2) estimate the delivery probability;
3) send the packet with delivery probability higher than a threshold.

For each packet in the network, we use $\vec{P} = (No, src, dst, thrsh)$ to describe its property. No denotes the packet number; src denotes the source node of the packet; dst denotes the destination node of the packet; $thrsh$ denotes the threshold of packet which is increased each time when the packet replicates. After receiving the property vector, the node will set the decision vector $r\vec{e}c$ to 0, which is used to describe whether the packets should be sent.

More specifically, considering two nodes i and j meet at time t, and the total number of packets in their buffer are respectively l and s, the general routing algorithm for node i is shown in Algorithm 1. After exchanging the property of packets with node j, node i first checks whether the packets in node j's buffer have already existed in its own buffer, and then calculates the delivery probability (detailed calculation is shown in (5)(6)). Next, the node j sets the rec_k to 1 for the packet k in node j's buffer if the delivery probability is bigger than the threshold. Then two nodes exchange $r\vec{e}c$. Based on the decision vector sent by node j, node i sends the packet k to node j, if rec_k^j equals 1. After receiving the packets from node j, node i records the total length of incoming packets.

V. PERFORMANCE ANALYSIS

In this section, we first analyze how the queue length changes, and then we further derive the staying time for the packet and the overall delivery probability.

A. Queue Length Estimation

Assume that a pair of node A and B contact at time t, and a packet with destination D saved in node A's buffer.

Algorithm 1: General Routing Algorithm

send the property vector of each packet in the buffer $\vec{P^i} = (\vec{P_1^i}, \vec{P_2^i}, \cdots, \vec{P_l^i})$ and receive the property vector of node j $\vec{P^j} = (\vec{P_1^j}, \vec{P_2^j}, \cdots, \vec{P_s^j})$;
set $\vec{rec^i} = (rec_1^i, rec_2^i, \cdots, rec_s^i) = 0$;
for each message M_k in node j's buffer **do**
  if message k already exists in node i's buffer **then**
    continue;
  else
    estimate the delivery probability pr_k;
    if $thrsh_k < pr_k$ **then**
      set $rec_k^i = 1$;
    end if
  end if
end for
send $\vec{rec^i}$ to node j and receive $\vec{rec^j}$
for $k = 1$; $k \leq l$; $k++$ **do**
  if $rec_k^j == 1$ **then**
    revise the threshold $thrsh_k$ and send packet k;
  end if
end for
receive the packet from node j;
update information of incoming packets;

The contact rate between node B and D is λ_{BD}. Once a packet is replicated, we set a fixed time stamp T_{exp}[1] for each new packet, and the packet, which stays in the buffer for a time longer than T_{exp}, will be deleted from the buffer. L_{Bi}^k denotes the number of packets from node i to node B at the k-th contact with distribution function $f_{L_{Bi}^k}$. $N_{Bi}(t)$ denotes the times that node i meets node B within time t. The total length of packets transmitted from node i to node B is derived by $X_{Bi}(t)=\sum_{k=1}^{N_{Bi}(t)} L_{Bi}^k$.

Lemma 1: If node A is packet sender and node B is packet receiver and $X_B^A(t)$ denotes the total length of incoming packets from nodes with higher priority than node A, the expectation and variance of $X_B^A(t)$ are as follows.

$$E(X_B^A(t)) = \sum_{i=1}^{N} \lambda_{Bi} x_{iA} E(L_{Bi}^k) t \tag{1}$$

$$\mathrm{var}(X_B^A(t)) = \sum_{i=1}^{N} \lambda_{Bi} x_{iA} E({L_{Bi}^k}^2) t \tag{2}$$

$$x_{iA} = \begin{cases} 1 & , \quad I_{Bi} < I_{BA} \\ 0 & , \quad otherwise \end{cases}$$

Proof: Since node A only cares about the incoming packet from the nodes with higher social ties, x_{iA} is used to exclude those packets with lower priority. Thus, the total length of incoming packets can be written as follows.

$$X_B^A(t) = \sum_{i=1}^{N} \sum_{k=1}^{N_{Bi}(t)} L_{Bi}^k x_{iA} = \sum_{k=1}^{N_B(t)} L_B^k$$

$N_B(t)$ denotes the total times that node B meets other nodes with rate $\lambda_B=\sum_{i=1}^{N} \lambda_{Bi} x_{iA}$, L_B^k denotes the length of packets received by node B at the kth contact with distribution function

$$f_{L_B^k}(\cdot) = \sum_{i=1}^{N} \frac{\lambda_{Bi} x_{iA}}{\lambda_B} f_{L_{Bi}^k}(\cdot)$$

By using Wald's equation, we can derive

$$\begin{aligned} E(X_B^A(t)) &= E(N_B(t)) E(L_B^k) \\ &= \lambda_B t \sum_{i=1}^{N} \frac{\lambda_{Bi} x_{iA}}{\lambda_B} E(L_{Bi}^k) \\ &= \sum_{i=1}^{N} \lambda_{Bi} x_{iA} E(L_{Bi}^k) t \end{aligned}$$

According to the law of total variance, we can derive

$$\begin{aligned} \mathrm{var}(X(t)) &= E(\mathrm{var}(X(t)|N(t))) + \mathrm{var}(E(X(t))|N(t)) \\ &= E(N(t)\mathrm{var}(L^k)) + \mathrm{var}(N(t)E(L^k)) \\ &= \lambda t (\mathrm{var}(L^k) + E(L^k)^2) \\ &= \lambda t E({L^k}^2) = \sum_{i=1}^{N} \lambda_{Bi} x_{iA} E({L_{Bi}^k}^2) t \end{aligned}$$

■

Lemma 2: If node A is packet sender and node B is packet receiver and $Z_B^A(t)$ denotes the queue length of node B at time t from the perspective of node A, the expectation and variance are as follows.

$$E(Z_B^A(t)) = Z_B(0) \tag{3}$$

$$\mathrm{var}(Z_B^A(t)) = 2 \sum_{i=1}^{N} \lambda_{Bi} x_{iA} E({L_{Bi}^k}^2) t \tag{4}$$

Proof: We first analyze the outgoing packets. There are two reasons for the packet leaving: 1) the packet expires; 2) The packet is dropped caused by the limited buffer. In our proposed scheme, since we consider the occurrence of packet dropping ahead of time, we believe that the occurrence of packet dropping is greatly reduced and the major reason for packet leaving is expiration. On the other hand, the expiration time is set to all incoming packets, and therefore the distribution for packet leaving is exactly corresponding to that of the incoming of packets.

$Y_B^A(t)$ denotes the total length of leaving packets in node B within time t. And from the above discussion, we believe that $Y_B^A(t)$ and $X_B^A(t)$ should have the same expectation and variance because of the fixed time stamp. The fluctuation of the queue length can be modeled as follows.

$$E(Z_B^A(t)) = E(Z_B(0) + X_B^A(t) - Y_B^A(t)) = Z_B(0)$$

[1]We use expiration time to describe this time stamp in the remainder of our paper

$$\begin{aligned}\mathrm{var}(Z_B^A(t)) &= \mathrm{var}(Z_B(0) + X_B^A(t) - Y_B^A(t)) \\ &= 2\sum_{i=1}^{N} \lambda_{Bi} x_{iA} E({L_{Bi}^k}^2) t\end{aligned}$$

■

B. Delivery Probability Estimation

During a time interval t, the probability that the packet is dropped is equivalent to the probability that the total length of packets with higher priority is larger than the buffer size. Thus for a fixed dropping probability, we can then define the staying time.

Theorem 1: Assume that node A is packet sender and node B is packet receiver and T_s denotes the time that the packets from node B can stay in node A's buffer. For a fixed dropping probability P,

$$T_s = \begin{cases} \frac{Z'_B}{c\sqrt{2\lambda_B E({L_B}^2)}} & , \frac{Z'_B}{c\sqrt{2\lambda_B E({L_B}^2)}} < T_{exp} \\ T_{exp} & , otherwise \end{cases} \tag{5}$$

$$Z'_B = Z_{MAX} - Z_B(0)$$

$$c = \Phi^{-1}(1 - P)$$

Z_{MAX} denotes the maximum buffer length.

Proof:

$$P_{overflow} = 1 - \Phi\left(\frac{Z_{MAX} - Z_B(0)}{\sqrt{2\lambda_B t E({L_B}^2)}}\right)$$

Since we only care about the minimum value between the dropping time and expiration time, t should be bounded to $(0, T_{exp})$. And the upper bound of overflow probability is

$$\sup_{t\in(0,T_{exp})} \{P_{overflow}(t)\} = 1 - \Phi\left(\frac{Z_{MAX} - Z(0)}{\sqrt{2\lambda_B E({L_B}^2) T_{exp}}}\right)$$

Let dropping probability P equals the upper bound of $P_{overflow}$, we can get the equation (5). ■

Then through staying time T_s, we can derive the delivery probability which is the major metric in our algorithm.

$$P_{del} = (1 - \exp(-\lambda_{BD} T_s)) \tag{6}$$

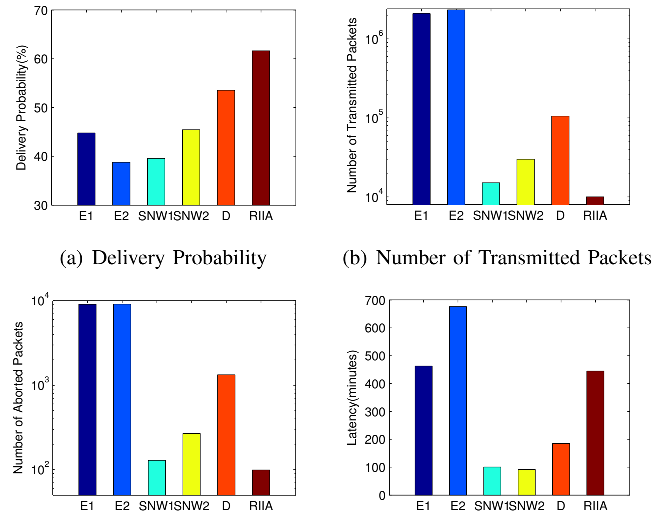
VI. EVALUATION RESULTS

A. Simulation Setup

We use the ONE DTN simulator [20] and the real trace INFOCOM06 to test our routing algorithm. This trace collects the movement of 78 participants who joined INFOCOM in 2006, and records their contacts within 4 days.

In our simulation, all nodes use the Bluetooth interface, which has 2Mbps bandwidth. The time interval for packet creation is from 50 seconds to 70 seconds, and the source and the destination are both randomly selected among all participants. The buffer size for each node is 5MB, and the size of each packet is randomly set from 50KB to 100KB.

Since there are few multi-copy routing algorithms for mitigating the social selfishness, we only choose some basic but benchmarking multi-copy routing algorithms which are Epidemic Routing [5], SprayAndWait [6], and Delegation [8]. The buffer management policy is uniformly set to all routing algorithms including ours, which drops the packet depending on the relative interpersonal influence. Since some other algorithms like MaxProp [21] incorporate the buffer management policy in their routing algorithm, we do not use these algorithms for comparison.



(a) Delivery Probability (b) Number of Transmitted Packets

(c) Number of Aborted Packets (d) Latency

Fig. 3. Effectiveness of Different Routing Algorithms: *Epidemic Routing* with 1200 and 2400 minutes TTL are notated "E1" and "E2" respectively; "SNW1" and "SNW2" represent *SprayAndWait* with 4 and 8 number of copies respectively; "D" stands for *Delegation*, and the contact probability is used as the metric which is calculated from contact rate, and the initial threshold for packet forwarding is 0.5; in our scheme, the expiration time is 1500 minutes, and the constant c in Equation (5) is 1, and the initial threshold is 0.5 which is the same with *Delegation*.

B. Results

We use four metrics to evaluate our routing algorithm: *delivery probability*, *the number of packet transmitted*, *the number of packet aborted*, and *latency*. In our experiment, different parameters are tested for all algorithms in order to find the best setting, but we only show some representative results in this paper, which are already sufficient to illustrate the advantage of our scheme. The simulation results are collected for the whole experiment time including the time for warm-up.

1) Delivery probability: Figure 3(a) shows the result of delivery probability. Since in our experiment we allow all the nodes to drop the packets from others with relatively lower social ties, the overall delivery probability cannot reach very high. But our algorithm outperforms other algorithms since we take both contact probability and buffer constraint into consideration. If the congestion is severe, the competition for buffer space is fierce, and consequently decreases the delivery probability which is the reason why *Epidemic* sends the most packets but achieves the lowest delivery probability. In addition, different from *SprayAndWait* and *Delegation*, *RIIA* avoids fierce buffer competition and chooses the nodes with light load to forward data, thus it achieves a higher overall delivery probability.

2) The number of transmitted packets: As we can see in Figure 3(b), since *Epidemic* and *Delagation* do not take into account the buffer constraint, the total number of packets being transmitted is much higher than *SprayAndWait* and *RIIA*. Because we take into account the buffer constraint, the number of transmitted packets can be greatly reduced. In addition, the congestion can be mitigated at the same time, which subsequently increases the delivery probability.

3) The number of aborted packets: We use the number of packets aborted due to the interruption of connection to test the influence of contact duration which is not considered in our routing algorithm. As we can see in Figure 3(b)(c), the number of aborted packets is proportional to the number of transmitted packets. Since the aborted packets only take 1% in the total number of transmitted packets, we do not consider the contact duration in our work.

4) Latency: As we can see in Figure 3(d), our protocol has a longer latency than *SprayAndWait* and *Delegation*. On one hand, the basic idea of our protocol is sending the packet to the nodes where buffer competition is not fierce in order to increase the probability that the packet can stay in the buffer. However, the path where congestion is not severe may not be a path with short distance, which means we may choose a longer path to forward data in order to increase its delivery probability. On the other hand, we do not take any action to decrease the latency. To solve this problem, we can also add one constraint on staying time when deciding whether to transmit the packet, which however decreases the overall delivery probability.

VII. CONCLUSION

In this paper, we focus on the social selfishness in buffer management and design a multi-copy routing algorithm mitigating the adverse effect of that phenomenon. In order to find the relative position in the competition for the buffer space, we use the interpersonal influence to model the selfishness from a global view rather than an absolute value between node pairs. The contact probability can be further derived from the contact rate, interpersonal influence, as well as historical incoming packet information. Experiment result is given in this paper, which demonstrates that our routing algorithm performs well if users drop the packet due to social ties when storage is scarce.

ACKNOWLEDGEMENT

The correspondence author of this article is S. Du. This research is supported by National Natural Science Foundation of China (Grant No.61003218, 70971086, 61272444, 61161140320, 61033014), Doctoral Fund of Ministry of Education of China (Grant No.20100073120065)

REFERENCES

[1] H. Zhu, X. Lin, R. Lu, Y. Fan, and X. Shen, “Smart: A secure multilayer credit-based incentive scheme for delay-tolerant networks,” *IEEE Transactions on Vehicular Technology*, vol. 58, no. 8, pp. 4628-4639, 2009.

[2] H. Zhu, S. Du, Z. Gao, M. Dong, and Z. Cao, “A Probabilistic Misbehavior Detection Scheme Towards Efficient Trust Establishment in Delay-Tolerant Networks.”, *IEEE Transactions on Parallel and Distributed Systems*, to appear, 2013.

[3] W. Gao, Q. Li, B. Zhao and G. Cao, “Social-Aware Multicast in Disruption-Tolerant Networks,” *IEEE/ACM Transactions on Networking*, vol. 20, Issue 5, pp. 1553-1566, 2012.

[4] W. Gao, G. Cao, T. La Porta, and J. Han, “On exploiting transient social contact patterns for data forwarding in delay-tolerant networks,” *IEEE Transactions on Mobile Computing*, vol. 12, no. 1, pp. 151 C165, 2013.

[5] A. Vahdat and D. Becker, “Epidemic Routing for Partially Connected Ad Hoc Networks,” Duke University, Tech. Rep. CS-200006, 2000.

[6] T. Spyropoulos, K. Psounis, and C. S. Raghavendra, “Spray and Wait: An Efficient Routing Scheme for Intermittently Connected Mobile Networks,” in *Proc. of ACM SIGCOMM workshop on Delay-tolerant networking*, 2005.

[7] G. Sharma, R. Mazumdar, and N. B. Shroff, “Delay and Capacity Trade-offs in Mobile Ad-hoc Networks: A Global Perspective,” *IEEE/ACM Transaction on Networking*, vol. 15, Issue 5, pp. 981-992, 2007.

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

[9] Q. Li, S. Zhu, and G. Cao, “Routing in Socially Selfish Delay Tolerant Networks,” in *Proc. of IEEE INFOCOM*, 2010.

[10] P. Hui, A. Chaintreau, J. Scott, R. Gass, J. Crowcroft, and C. Diot, “Pocket switched networks and human mobility in conference environments,” in *Proc. of ACM SIGCOMM workshop on Delay-tolerant networking*, pp. 244-251, 2005.

[11] Y. Li, P. Hui, D. Jin, L. Su, and L. Zeng, “Evaluating the Impact of Social Selfishness on the Epidemic Routing in Delay Tolerant Networks,” *IEEE Communications Letters*, vol. 14, no. 11, pp. 1026C1028, 2010.

[12] L. Qiu, Y. Li, P. Hui, D. Jin, L. Su, and L. Zeng, “Edge-Markovian Dynamic Graph Based Performance Evaluation for Delay Tolerant Networks,” in *IEEE Wireless Communications and Networking Conference (WCNC)*,pp. 2129-2133, 2012.

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

[14] T. Spyropoulos, K. Psounis, and C. S. Raghavendra, “Efficient Routing in Intermittently Connected Mobile Networks: The Multiple-Copy Case,” *IEEE/ACM Transaction on Networking*, vol. 16, pp. 77-90, 2008.

[15] J. Z. Haas, and T. small, “A New Networking Model for Biological Applications of Ad Hoc Networks,” *IEEE/ACM Transaction on networking*, vol 14, Issue 1, pp 27-40, 2006.

[16] Y. Li, M. Qian, D Jin, L Su, and L Zeng, “Adaptive Optimal Buffer Management Policies for realistic DTN,” in *Proc. of IEEE GlobeCom*, 2009.

[17] A. Krifa, C. Barakat, and T. Spyropoulos, “Message Drop and Schedulingin DTNs: Theory and Practice,” *IEEE Transaction on Mobile Computing*, vol. 11, no. 9, pp. 1470-1483, 2012.

[18] A. Krifa, C. Barakat, and T. Spyropoulos, “Optimal Buffer Management Policies for Delay-Tolerant Networks,” in *Proc. of IEEE SECON*, pp. 260C268, 2008.

[19] J. Wiese, P. G. Kelley, L. F. Cranor, L. Dabbish, J. I. Hong, and J. Zimmerman. “Are you close with me? Are you nearby?: investigating social groups, closeness, and willingness to share,” in *Proc. of the 13th International Conference on Ubiquitous Computing (UbiComp)*, 2011.

[20] A. Keranen, J. Ott and T. Karkkainen, “The ONE Simulator for DTN Protocol Evaluation,” *SIMUTools’09: 2nd International Conference on Simulation Tools and Techniques*, Rome, March 2009.

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