A General Data and Acknowledgement Dissemination Scheme in Mobile Social Networks

Ning Wang and Jie Wu
Department of Computer and Information Sciences, Temple University, USA
Email: {ning.wang, jiewu}@temple.edu

***Abstract*—In this paper, a general data and acknowledgement dissemination mechanism is proposed in mobile social networks (MSNs). Most existing dissemination schemes in MSNs only consider data transmission. However, receiving acknowledgement has many potential applications in MSNs (e.g., mobile trade and incentive mechanism). Challenging problems thus arise due to this type of mixed messages (i.e., data and acknowledgement) dissemination problem. The buffer constraint and time constraint for data and acknowledgement make this problem even harder to handle in a practical scenario. In order to maximize the research objective (e.g., low delay and high delivery ratio), we have to identify the priority of each message in the network. We propose a general *priority-based compare-split* routing scheme to solve the above buffer exchange problem. During each contact opportunity, first, nodes compare their abilities to send data and acknowledgement based on two types of criteria. They are the contact probability and the social status, which estimate the nodes' direct and indirect relationship with destinations respectively. Nodes then decide which message to exchange, and thus maximize the combined probability. Second, an adaptive priority-based exchange scheme is proposed within each type of message, and so is the relative priority between two types of messages, as to decide the order of exchange. The message with a high priority will transmit first, and thus maximize the research objectives. The effectiveness of our proposed scheme is verified through the extensive simulation in synthetic and real traces.**

***Keywords*—Buffer exchange, priority setting, acknowledgement.**

I. INTRODUCTION

The wide usage of mobile devices (e.g., smartphones and tablets) and the evolution of high-speed short-distance wireless communication (e.g., Bluetooth 4.0 and WIFI Direct) in the recent years has stimulated lots of research in mobile social networks. Currently, we simply use centralized cellular networks (e.g., GSM and 3G) to transmit two types of data transmission, the intermediate data (e.g., voice and video chat), and delay-tolerant data, (e.g., email and software update). We expect that the MSNs are complementary network communication technologies to cellular networks and MSNs are suitable for delay-tolerant data for a local community in which the participants have frequent interactions, (e.g., people working in the same building, students studying in the same school). In MSNs, data are buffered for extended intervals of time until an appropriate forwarding opportunity is recognized in hopes that it will eventually reach its destination (i.e., store-carry-forward) [1]. As a result, MSNs extend communications between mobile devices from the restrictions of cellular infrastructure, improve the capacity of the network and mitigate the congestion for traditional centralized communication methods, and reduce the communication cost simultaneously.

If we consider mobile users in the real world, they can be either cooperative or non-cooperative. In this type of autonomous network, a proper incentive scheme is imperative to stimulate nodal cooperation, and to attract more participants. As a result, the system performance has a big difference. A famous example is the DARPA Network Challenge [2], the aim of which is to find 10 red, weather balloons at 10 previously undisclosed fixed locations in the continental United States. It was thought to be a hard problem. However, participants were incentivized recursively by the MIT team. Although the team began with only 15 people, it eventually grew to 5,000 people, who correctly identified the location of all of the balloons in 8 hours, 52 minutes, and 41 seconds. The idea behind the recursive incentive strategy, called Multi-Level Marketing (MLM), is that not only the last person who finds the balloons will get the reward, but also those connecting the finder. The amount of credit that they can get reduces by half each time. An illusion of the incentive scheme is shown in Fig. 1.

However, without centralized communication methods, to apply this type of incentive mechanism in MSNs, a common method is to collect the acknowledgement from the destination. That is, the relays can inject information into the message. After the data reaches the destination node, the destination node will send the forwarding list, as an acknowledgement, back to the source. This is one of the motivations for why we study this type of mixed data and acknowledgement dissemination problem in MSNs. To make the research objective more practical, the buffer constraint for nodes and time constraints for data and acknowledgement are further considered. Ideally, we have two research objectives: the data should be disseminated to the destination quickly, and acknowledgement should be sent back quickly. However, due to limited contact opportunities, we might not be able to achieve both objectives. If not, which objective should have a higher priority?

Before we answer the above questions, two more questions arise naturally: (1) How can a node compare the benefits of keeping the message and the benefits of exchanging it with others? (2) What's the possible benefit gain for each buffer exchange? The benefit can be regarded as the smaller delivery delay, or higher delivery ratio, and so on.

For the first question, most existing methods try many different methods to estimate the contact probability between the encounter node and the destination of the message, and forward the message to the node with a higher contact probability. It is called the *strongly connected relationship with destination* in this paper. However, if we only consider this type of strongly connected relationship with the destination, we will miss lots of useful information. The node's relationship

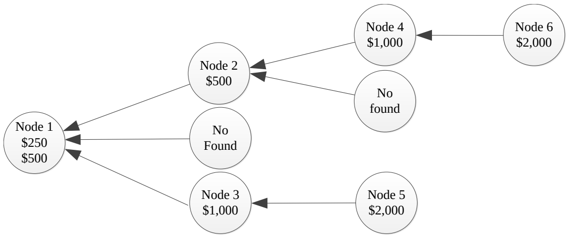
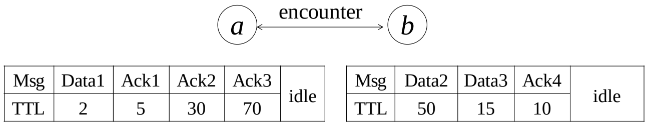


Fig. 1. An illustration of the MIT team's incentive strategy where nodes 5 and 6 finally found the balloon, with each of them getting $2,000 rewards, and the referees each getting $1,000 each; we keep doing this recursively.

TABLE I. SUMMARY OF SYMBOLS

Symbol	Interpretation
$p_a(b)$	Contact probability between node a and node b
$P(a)$	Priority of message a
Δ_a	Probability difference of two nodes to destination a
$S(a)$	Social status of node a
$F(a)$	Nodes encounter with node a in a given time interval
D	Combined destination set of two encounter nodes
$E_a(b)$	Expected delay for relay node a to send data to node b
τ_a	Time of life of message a
α	Relative priority of data and acknowledgement



Msg	Data1	Ack1	Ack2	Ack3	idle
TTL	2	5	30	70	

Node a's buffer

Msg	Data2	Data3	Ack4	idle
TTL	50	15	10	

Node b's buffer

Fig. 2. An illustration of the network model where the value under the data and acknowledgement represents the TTL, and the blank region in the figure means the idle buffer.

with other nodes' (excluding the destination) is called *weakly connected relationship with destination* in this paper, and it does matter for relay selection. In this paper, two types of criteria, social status, the centrality of a node in the network, and contact probability, the probability of a node to meet destination, are used together to estimate the ability that a node can act as a relay to send a message to the destination. By introducing the concept of social status, we combine the direct probability, two-hop indirect probability, and the influence of the other weakly-connect relationship with the destination, and estimate the probability more accurately.

As for the second question, we should set the priority of the message according to the possible beneficial gain. We give the following two criteria in this paper: (1) *Usefulness*: If we forward it, it is highly likely to reach the destination before the deadline so that the work can pay off. (2) *Urgency*: The data that is close to the deadline should have a higher priority. A proposed priority setting leverages the above two criteria. The relative priority between data and acknowledgement is further proposed in different scenarios. In this paper, we proposed two types of scenarios (i.e., the data-first and acknowledgement-first). The idea for the data-first scenario is that nodes want to send data to destinations as soon as possible, such as weather forecast updates and news feeds; Otherwise, the data will expire. However, in the acknowledgement-first scenario, such as mobile trade or some source-incentive mechanisms, the acknowledgement is more important, because the relays would like to get the credit as soon as possible, so that they try to send the acknowledgements back quickly. Our proposed method is a general routing scheme, which can be used in the above-mentioned message dissemination or incentive mechanism scenarios.

The contribution of this paper is organized as follows: (1) We propose a general routing scheme to accelerate the data and acknowledgement transmission simultaneously, with time and buffer constraints. (2) We combine two types of criteria together to estimate a node's ability to act as relay for the strongly connected relationship and the weakly connected relationship with the destination. Thus, the estimated contact probability is more accurate. (3) We propose an adaptive priority scheme for each type of message, so that message which contributes to performance most will be sent first.

The remainder of the paper is organized as follows. The overview of the network model is introduced in Section II. The proposed *priority-based compare-split* algorithm is presented in Section III. After that, an analysis is provided in section IV. The evaluation setting and the simulation results are shown in Section V. Then we review the related work in Section VI, and conclude the paper in Section VII.

II. OVERVIEW OF THE NETWORK MODEL

A. System Model

There might not exist a contemporaneous end-to-end path for content transmission in MSNs, and thus the messages suffer a relatively big delay. However, it does not mean that delay can be any longer in MSNs. Let's take an example of news; we do not need to receive all the latest news in a real time; a 2-hour delay is acceptable, but a 10-hour delay should not be tolerated. So, a deadline or time-to-life (TTL) is needed, or the messages might be out-of-date and meaningless.

In our model, we consider the single-copy multiple messages scenario. If the source wants to send a message to a node, it can choose to send the message itself, or ask other nodes for help. For the latter case, after the source forwards the message to relay, the relay will inject its ID into the message, and later exchange the buffered messages with other encounter nodes to accelerate the message transmission. If the message does not reach the corresponding destination before the deadline, the message is discarded. Otherwise, the message is forwarded to the destination in time, and the destination node will generate an acknowledgement to notify the corresponding source of the relays involving the data forwarding. This type of acknowledgement mechanism is used in many applications. For example, in the incentive model, if the source receives the acknowledgement, it will provide the promised credit to the listed relays through a centralized virtual bank. We further assume all the nodes are honest. They would not forge information to other nodes, or hide the messages that they carried. The security-related problems are out of the scope of this paper, and there exist some strong authentication schemes [3, 4] that provide the verification of information.

However, the buffer size of the node is limited in practical application. We have to assign a priority to each message and let the most beneficial messages exchange first. The two types of messages, data and acknowledgement, have different relative priorities in different application scenarios. We assume that nodes encounter each other in a pairwise manner, or we can use nodes' IDs to decide the communication order. An illusion of the network model is shown in Fig. 2.

B. Probability Estimation

In this paper, we not only estimate the strongly connected relationship with destination nodes, as most existing schemes do, but also define the social status to distinguish a node's weakly connected relationship with the destination. Then, we make a routing decision based on these two criteria. By combining the relationship of nodes with the destination and the other nodes together, we can estimate the ability of a node to act as relay more accurately.

Let's take Fig. 3 as an illustration. Consider two nodes a and b, whose probabilities of reaching destination d is 0.4 and 0.6 respectively. Should node a split its message to b when it meets node b? In this example, though the contact probability between node b and destination node d, is larger than that of node a, node a should not forward the message to node b for the following two reasons: (1) Node a has a high indirect contact probability with node d. The probability that node a sends a message to d from node c is $p_a(c) \cdot p_c(d) \geq p_b(d)$. (2) Node a has a high probability of meeting other nodes, and has high probabilities of reaching the destination. The probability of node a meeting at least one of node e and f is $1 - (1 - p_a(e)) \cdot (1 - p_e(f))$. The probability of sending message from node e or f is also 0.6. From this example, it is clear that the weakly connected relationship with the destination has an influence on the routing decision. That is, if a node is very popular, even this node does not have a high contact probability with the destination, this node can still meet other nodes, which have quite a good relationship with the destination. Just forwarding the message to the encounter node based on the contact probability, without considering the weakly connected relationship with the destination, might not be a good choice.

(1) *Contact probability:* a priori estimation of the contact with a destination in the network in a given period, which can be derived from the contact frequency to estimate the relationship between the relay and destination. The contact probability decays with time. An exponentially weighted moving average (EWMA) method is usually used to update the contact probability. The contact probability of node a with node d can be written in the following format:

$$p_a(d) = \begin{cases} (1-\beta) \cdot p_a(d)_{old} + \beta & \text{encounted} \\ (1-\beta) \cdot p_a(d)_{old} & \text{time out} \end{cases} \tag{1}$$

where β is an empirical value that we can get a proper value from extensive experiments.

(2) *Social status:* a priori estimation of the node's centrality in the network in a given period. It can be written in the following function:

$$S(a) = S(a)_{old} + \sum_{b \in F(a)} p_a(b) \cdot S(b) \tag{2}$$

where $F(a)$ represents the nodes which encounter node a in a time interval. Initially, all the nodes in the network are assigned the same value. When a pair of nodes encounter each other, they will upload their social status. Along with time, nodes frequently contacting other nodes will have high value, and other nodes will keep a small value for social status.

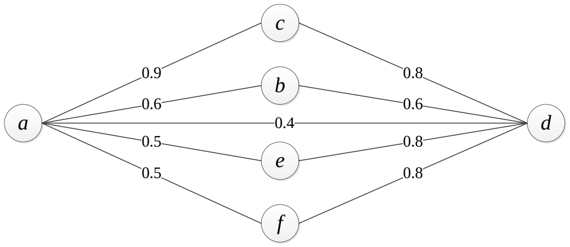


Fig. 3. A contact graph of network where the value in the edge is the contact probability between the two endpoints.

III. PRIORITY-BASED COMPARE-SPLIT SCHEME

In this paper, we propose a *priority-based compare-split* scheme to maximize the achievable benefit during each exchange. We will present the two steps of this scheme, and illustrate them by an example. The first step is *compare*, which collects the necessary information for routing. The second step is *priority-based split*, which decides how to split the messages in order to maximize the achievable beneficial gain.

A. Compare

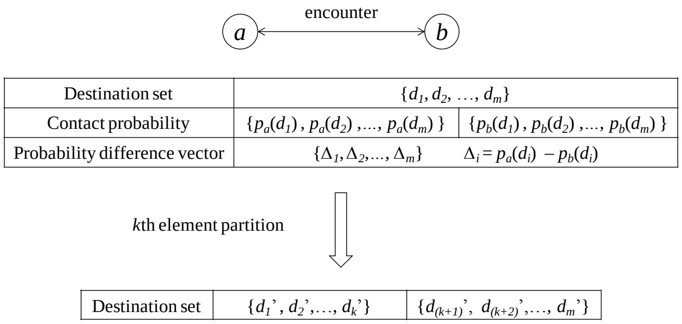
Upon the contact between node a and node b, they exchange their probability vectors to corresponding destinations of messages they carry. Then, each node knows the combined destination set for messages buffered in them and the corresponding probability vector. Let's denote the combined destination set D, $D = \{d_1, d_2, \ldots, d_m\}$, and the probability vectors of node a and node b to destination set D as $\{p_a(d_1), p_a(d_2), \ldots, p_a(d_m)\}$, and $\{p_b(d_1), p_b(d_2), \ldots, p_b(d_m)\}$ respectively. The social statuses of nodes are also changed during this step. Note that there exist two rounds of exchanges. One round is to exchange the destination of the messages in their buffer, and another round is to exchange their probability vectors.

Definition 1. *The probability difference vector of node a and node b is the probability difference to the destination set of the messages they carried. Suppose the destination set is* $\{d_1, d_2, \ldots, d_m\}$, *and the probability difference vector is* $\{\Delta_1, \Delta_2, \ldots, \Delta_m\}$, *where* $\Delta_i = p_a(d_i) - p_b(d_i)$.

The destination set splitting is based on the ratio of two encounter nodes' social statuses. The remaining number of destination set k can be denoted as:

$$k = \lceil \frac{S(a)}{S(a) + S(b)} \times m \rceil$$

The process is described as follows, and is shown in Fig. 4.



Destination set	$\{d_1, d_2, \ldots, d_m\}$	
Contact probability	$\{p_a(d_1), p_a(d_2), \ldots, p_a(d_m)\}$	$\{p_b(d_1), p_b(d_2), \ldots, p_b(d_m)\}$
Probability difference vector	$\{\Delta_1, \Delta_2, \ldots, \Delta_m\}$	$\Delta_i = p_a(d_i) - p_b(d_i)$

Destination set	$\{d_1', d_2', \ldots, d_k'\}$	$\{d_{(k+1)}', d_{(k+2)}', \ldots, d_m'\}$

Fig. 4. An illustration of ratio-based-split where each node will carry the messages that they are more likely to encounter through a buffer exchange.

- Both a and b generate the probability difference vector $(\Delta_1, \Delta_2, \ldots, \Delta_m)$. Find the kth largest element using a sorting algorithm.
- Node a keeps messages for destinations that have higher values than, or equal values to, the kth largest element. When two probability differences are equal, the node's ID is used to break the tie.
- Node b keeps messages for the remaining $m - k$ destinations that have lower values than, or values equal to, the kth largest element.

Let's take Fig. 5 as an example to illustrate the compare scheme. From the top of the figure, you can get the contact probability and social status of nodes a and b. During the contact, node a and node b first exchange the corresponding probability vectors. Node a sends probability vectors $\{p_a(d_1),\ p_a(d_2),\ p_a(d_3),\ p_a(d_4),\ p_a(d_5)\}$ to node b, and node b sends probability vectors, $\{p_b(d_1),\ p_b(d_2),\ p_b(d_3),\ p_b(d_4),\ p_b(d_5)\}$ to node a. Then both nodes form a destination set D, $\{d_1, d_2, d_3, d_4, d_5\}$. In this example, we choose node a's view to calculate the probability difference vector, $(\Delta_1, \Delta_2, \Delta_3, \Delta_4, \Delta_5)$ is $\{0.2, , -0.8, 0.3, -0.1, 0.1\}$, and the remaining number of destinations are $k = \lceil \frac{S(a)}{S(a)+S(b)} \times m \rceil = 2$. Then, we will split the messages of node a and node b. That is, node a should keep the messages for destination $\{1, 3\}$. and node b should keep the messages for destination $\{2, 4, 5\}$ to maximize the combined probability.

B. Priority-based Split

After the split process, we can exchange the messages between a pair of nodes. Due to the limited buffer constraint and contact opportunity, we should decide which message to transmit first. Intuitively, we hope that the message can be delivered to the destination in time, and the message which is close to its TTL should be transmitted first. Also, the expected delivery delay should be considered the priority. We define

$$E_a(d) = \frac{1}{S(a) \times p_a(d)} \tag{3}$$

as the expected delay.

As for the priority setting, we should not only consider how much time remains before the message expires, but also consider how much time is expected to arrive at the destination. So we define the priority as follows:

$$P(a) = \begin{cases} \frac{E_a(d)}{(\tau_a - t)} & (\tau_a - t) > E_a(d) \\ 0 & (\tau_a - t) < E_a(d) \end{cases} \tag{4}$$

Where the τ_a is the TTL of message a. The idea is that $\tau_a - t$ is the remaining time for message a, and thus we should set a lower priority for messages which have a long remaining time. However, the expected delay $E_a(b)$ has an influence on the priority. For example, though the remaining time of message a is smaller than message b, the expected delay of message a is much smaller than that of message b. In this case, message a should have a lower priority since it can reach the destination before the deadline in a high probability. So, $\frac{E_a(d)}{(\tau_a - t)}$ leverages the expected delay and the remaining time to represent the priority of message. If the remaining time is smaller than the expected delay, we should set the priority of the message as 0, since it is highly possible that this message cannot reach its destination before the deadline, so that we do not waste the precious contact opportunity.

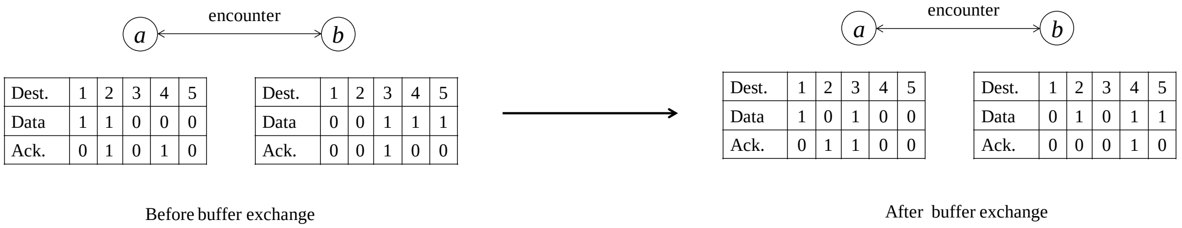
According to the different application scenarios in message dissemination, such as mobile advertising or public information dissemination, we should assign different priorities to data and acknowledgements. We define $P(a)$ as the priority of a data, and $P(b)$ as the priority of an acknowledgement. Further, a relatively important factor α is defined in the priority setting, and the size of different types of messages are also embedded into this factor. Basically, we have the following two application scenarios:

- Data-first: $\alpha P(a) > P(b)$. For mobile advertising, we want to disseminate as much as data as possible in the network, in hopes that data can reach more interested nodes. As for mobile trade, when the destination pays for the credit to relays, relays care more about whether the generated data can be delivered to the corresponding destinations as soon as possible so that they can earn credit.
- Acknowledgement-first: $\alpha P(a) < P(b)$. For the incentive mechanism and mobile trade, the source might be more interested in checking whether the disseminated data reaches the corresponding destinations. Relays care more about whether they can cash the credit, acknowledgement, as soon as possible.

In the above two scenarios, both the sender and relays hope that data and acknowledgements can be transmitted soon. This is a win-win exchange strategy. By using our proposed scheme, after each exchange, the buffered message with high priority in each node is exchanged, and has a higher probability of reaching the corresponding destinations.

We can illustrate the priority-based split scheme in Fig. 5. The left-bottom table in the figure shows the buffer information of node a and b before the split, where the value represents how much data or how many acknowledgements are destined to reach the corresponding destination. To get the maximum feasible combined probability by considering the priority of messages and buffer constraint, first, we calculate the priority of the data and acknowledgement by Equation 4. Then we will get a total transmission order of the data and acknowledgement

Destination set	$\{d_1, d_2, d_3, d_4, d_5\}$	
Social status	6	9
Contact probability	{0.6, 0.2, 0.7, 0.4, 0.7}	{0.4, 1, 0.4, 0.5, 0.6}
Probability difference vector	{0.2, -0.8, 0.3, -0.1, 0.1}	
Partition	{1, 3}	{2, 4, 5}
Priority of the data	{5, 4, 3, 2, 1}	
Priority of the acknowledgement	{0, 0.5, 1, 1.5, 0}	



Dest.	1	2	3	4	5
Data	1	1	0	0	0
Ack.	0	1	0	1	0

Dest.	1	2	3	4	5
Data	0	0	1	1	1
Ack.	0	0	1	0	0

Before buffer exchange

Dest.	1	2	3	4	5
Data	1	0	1	0	0
Ack.	0	1	1	0	0

Dest.	1	2	3	4	5
Data	0	1	0	1	1
Ack.	0	0	0	1	0

After buffer exchange

Fig. 5. An example of priority-based compare-split routing where the table in the top provides the information for buffer exchange, and the lower figure shows the buffer condition before and after the exchange

for node a and b, as shown in the figure. We further assume the size of the data is the same with the size of acknowledgement, and each node can carry 4 messages, at most, in this example. According to the compare step, node a has 3 messages to exchange, and node b has 2 messages to exchange. Due to the buffer constraint, only the 2 messages with the highest priority of each node have an opportunity to exchange in this encounter. For node a, that means the data to destination 2 and acknowledgement to destination 4. As for node b, that means the data to destination 3 and the acknowledgement to destination 3.

There exist two types of TTL: one is the deadline for data, and one is the TTL for acknowledgements. We should distinguish these two types of TTL, since they might have different values in different scenarios. For example, in the data-first scenario, we might set the TTL of data longer to increase the opportunity for successful delivery, and vice versa in the acknowledgement-first scenario.

C. Extension

Our proposed scheme is used in the single-copy unicast scenario. However, it can be extended into multiple-copy scenarios. In this case, one node can encounter different nodes buffering the same data more than one time. For example, source node s generates data i for node d. Then node s encounters node a and node b, and relays data i to them (in order). Another node c later encounters node a and b buffered with data i in sequence and did buffer exchange. Node c should not assign the same priority of data i during these two buffer exchanges. A naive idea is that the priority of the data should decrease as the encountering times increase. That is, the priority of data i is determined by a tuple $\langle times, P(i)\rangle$. As for the acknowledgement generation, the destination will only send back an acknowledgement when it receives data for the first time. In an incentive scenario, this situation is hard to handle. Should the source pay some credit for the relays which send the data to the destination late? If so, how can the source know their work, and how much credit should the source assign for them? Clearly, from the perspective of the source, this late delivery is meaningless. However, if it does not pay for the later relays, this type of incentive mechanism might not work. This is because the relays will very possibly get nothing to help the source. In addition, in multicast scenarios, there exist multiple acknowledgements for one piece of data from different destinations. So there exist four types of relative priorities, different data, the same data, acknowledgements for different data, and acknowledgement for the same data. We should assign the above four relative priorities carefully, or the network can jam with limited buffer.

IV. ANALYSIS

A. Optimal Split Algorithm

The motivation for our proposed scheme is to ensure that each message can be buffered in the node, which has a relatively high probability of reaching the corresponding destination, and thus minimizes the expected delivery delay.

Suppose D_a and D_b is the destination set of messages in nodes a and b's buffers, respectively. We would like to maximize the combined contact probability of the messages in a and b as follows:

$$max\{\sum_{i\in D_a} p_a(d_i) + \sum_{j\in D_b} p_b(d_j)\}$$

Lemma 1. *Suppose D_a and D_b are two subsets, as results of kth element partition. $\Delta_i = p_a(d_i) - p_b(d_i)$ is called the probability difference between nodes a and b for destination i. Maximum combined probability occurs when for each $i \in D_a$ and $j \in D_b$, $\Delta_i \geq \Delta_j$.*

Proof: It is clear that any other partition (including the optimal one) can be generated through a sequence of swaps of messages between two nodes, a and b. We show that each such swap will deteriorate the combined probability. Suppose a message buffered from D_a is assigned into D_b. Based on the split process, we will always have the condition $\Delta_i \geq \Delta_j$, that is, $p_a(d_i) - p_b(d_i) \geq p_a(d_j) - p_b(d_j)$, or

$$p_a(d_i) + p_b(d_j) \geq p_b(d_i) + p_a(d_j)$$

Algorithm 1 Priority-based compare-split routing

```
Input: Destination set $D = \{d_1, d_2, \ldots, d_m\}$,
    $\{p_a(d_1), \ldots, p_a(d_m)\}$, $\{p_b(d_1), \ldots, p_b(d_m)\}$.
    social status of $S(a)$ and $S(b)$ and relative priority $\alpha$
Output: The data exchange result of two nodes.
Calculate the the probability difference vector between two nodes, $(\Delta_1, \Delta_2, \ldots, \Delta_m)$.
Find $k = \lceil \frac{S(a)}{S(a)+S(b)} \times m \rceil$ and split at the $k$th largest element.
Calculate the priority of each message.
Exchange messages based on the order of the priority.
```

Note that $p_a(d_i)+p_b(d_j)$ is the combined probability involving destinations i and j, whereas $p_b(d_i)+p_a(d_j)$ is the combined probability after the swap of i and j. ■

Theorem 1. *If we consider the priority of messages and buffer constraint, the proposed priority-based compare-split scheme achieves a maximum feasible combined probability.*

Proof: When the buffer sizes are enough, nodes a and b will exchange all the messages as the proposed partition, and thus achieve the maximum combined probability, according to lemma 1. When the buffer sizes are not enough for nodes a and b to exchange all the messages, the changeable subset of messages are exchanged as the proposed partition, and thus achieve the partial maximal combined probability. This is because the remaining messages do not have an opportunity to exchange. Thus, our priority-based compare-split scheme achieves a maximum feasible combined probability. ■

This optimal split algorithm can partition the destinations to nodes with a higher probability; hence, the latency of the message close to the TTL is reduced.

B. Case Study

Assume that nodes' contact probabilities follow an exponential distribution with a contact rate of λ. It means that nodes have a probability $1 - e^{-\lambda T}$ of meeting each other within time T. In a realistic application scenario, we only care about whether the messages can be sent to the destination before the deadline or not. Based on the assumption above, the probability of the message being delivered to the destination at time t after it enters the buffer can be given by

$$f(t) = \lambda e^{-\lambda t}$$

As we are only interested in delivered messages, the probability function given above becomes a conditional probability for the messages that are delivered:

$$f_d(t) = \frac{f(t)}{P(t<\tau)} = \frac{\lambda e^{\lambda t}}{1-e^{\lambda \tau}} \tag{5}$$

Where $P(t<\tau)$ denotes the probability that the destination is reached before τ, which is given by the cumulative probability distribution of $f(t)$. Therefore, the expected waiting time of a delivered message can be written as

$$\begin{aligned} E &= \int_0^{\tau} t f_d(t) dt \\ &= \frac{\lambda}{1-e^{-\lambda\tau}} \int_0^{\tau} t e^{-\lambda t} dt \\ &= \frac{\lambda}{1-e^{-\lambda\tau}} \left[\frac{-te^{-\lambda}}{\lambda} \bigg|_0^{\tau} + \frac{1}{\lambda} \int_0^{\tau} e^{\lambda t} dt \right] \\ &= \frac{1}{\lambda} - \frac{e^{-\lambda\tau}}{1-e^{-\lambda\tau}} \tau \end{aligned} \tag{6}$$

From the above equation, we can get the conclusion that the expected delay of delivered messages will be less then $1/\lambda$. Combined with Equation 5, we can calculate the priority of messages by using the proposed method.

V. SIMULATION

The objective of this paper is to develop a general efficient mechanism to accelerate message dissemination in MSNs. Two performance metrics are used: (1) Delivery ratio: the number of messages which arrive at corresponding destinations before TTL out of all the generated messages in a certain interval. (2) Latency: the average duration between a message's generation and the arrival time at the destination. Efficient means that the message with a high priority can be transmitted to the destination in a low delay, and high delivery ratio.

A. Simulation Methods and Setting

During the simulations, we use not only synthetic mobility models, but also real traces, to verify the efficiency of the proposed scheme. We will compare the delivery ratio and latency in each trace.

1) Uniform mobility models: In synthetic mobility models, we set up a network with 20 nodes. Among them, 5 nodes are set as source nodes and 5 modes are set as the destination nodes. The social status of nodes follows the uniform distribution model. The contact probability is generated based on the social status to satisfy Equation 2. We set a 10,000 seconds contact history in our simulation, and every 1 second a new data generates in the network randomly within the source nodes. The contact event of two nodes is randomly generated, and the contact number is proportional to the contact probability of two nodes.

2) The real trace: We use the real trace *Infocom* 2006 trace [5] in our simulation, which has been widely used in MSN routing simulations. This dataset consists of contact traces between short-range Bluetooth wearable devices (iMotes) carried by individuals. Groups of participants are asked to carry small devices (iMotes) for four days during the INFOCOM 2006 conference. The contact information of the 78 participants are recorded in the iMotes. Besides, 20 stationary (long range) iMotes are placed in the experiment. There are 223,657 contacts between these nodes during the 342,915 seconds. Every 50 seconds, new data is randomly generated in our simulation. Also, among them, 10 nodes are set as the source nodes and 10 nodes are set as the destination nodes.

By using these synthetic mobility models and the real trace, we further set the buffer constraint. We do not consider the

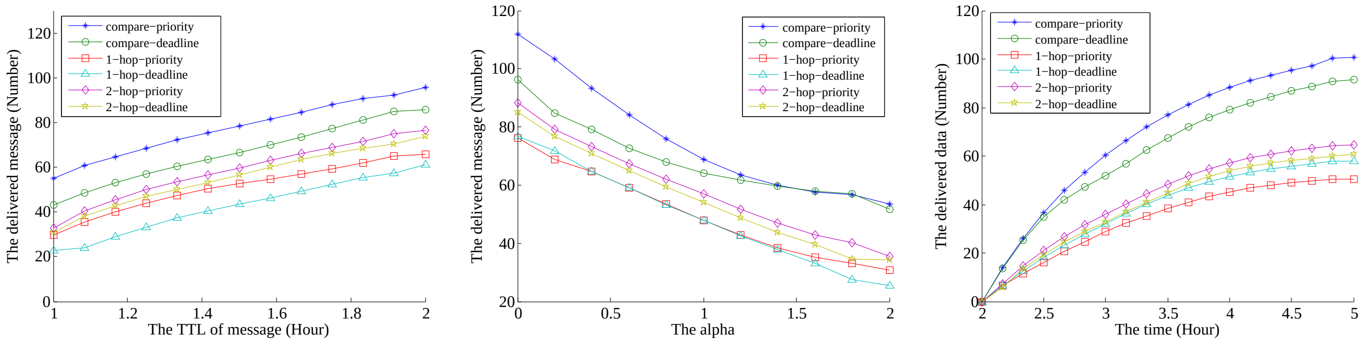


Fig. 6. Number of delivered data in the *Infocom2006* trace

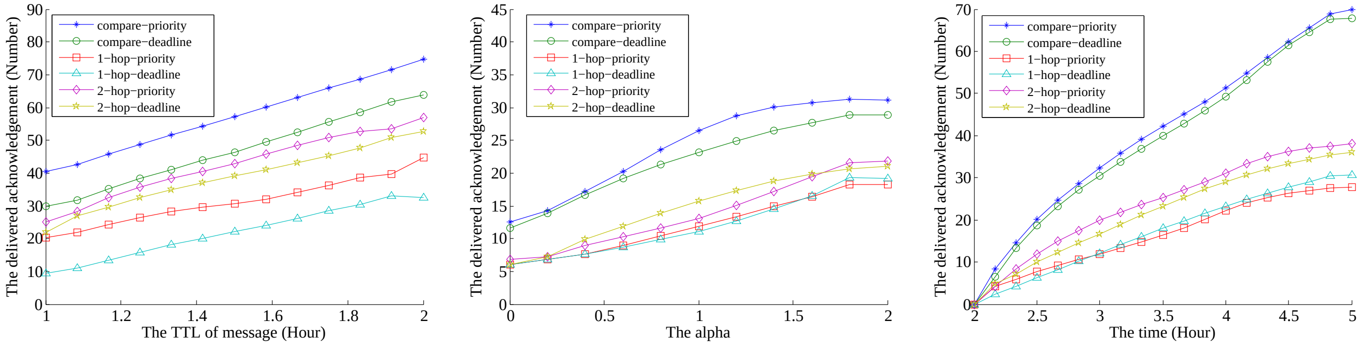


Fig. 7. Number of delivered acknowledgements in the *Infocom2006* trace

bandwidth constraint in this paper, which is reasonable since the current wireless communication speed is fast, compared with the size of the messages. To simplify the simulation, we assume that the size of data and acknowledgement is the same.

B. Compared Algorithms

For the contact probability estimation, we will compare our algorithm with two more probability estimations: one-hop routing, which forwards messages based on strongly contact relationship with the destination node, and two-hop contact probability, which has a transitive two-hop contact probability estimation to the destination node. As for the priority, we can set the remaining time of messages as a criterion and our proposed priority setting. The combination of the above probability estimation and priority setting is 6 algorithms. In the remainder of this paper, we will use compare-priority, compare-deadline, 1-hop-priority, 1-hop-deadline, 2-hop-priority, and 2-hop-deadline to represent these 6 algorithms.

C. Simulation Results

The results can be seen from the Figs. 5-8; Since we set the same data generation ratio, the message delivery number can be used to represent the delivery ratio in our simulation.

1) We try to find the influence of the TTL for the delivery ratio, so we increase the TTL of the messages. Both the delivery number of data and the acknowledgement increase with an increase in data's TTL. It is because that, along with the increasing of data's TTL, more data can be sent to the destinations and more acknowledgements are also generated at the same time. As a result, more acknowledgements are sent back to their sources. It is a near-linear increase in the *Infocom2006* trace before convergence. For the synthetic trace, the delivery number increases very quickly before convergence. Then all the messages are sent to the destination when TTL is large. Any further increase is meaningless, but consumes more buffer resources. Among the six algorithms, the compare-priority algorithm always has the best performance. This is followed by the compare-deadline, 2-hop-priority, 2-hop-deadline, 1-hop-priority and 1-hop-deadline algorithms. The proposed method delivers 53% data and 62% more acknowledgements than do the 1-hop-deadline algorithms.

2) We try to find the impact of the relative priority in the delivery ratio of messages. The results show that along with the increasing of α, the amount of delivered data decreases, and the number of delivered acknowledgements increase at the same time. However, the generation ratio decreases. This type of margin decreasing phenomenon appears clearly in the synthetic trace. For the *Infocom2006* trace, this phenomenon just shows in the proposed compare-priority and compare-deadline schemes. The reason is that with the increasing priority of acknowledgements, the messages cannot be exchanged to the better relays in the limit contact opportunity. So more and more messages are buffered into the intermediate nodes until they time out, and thus the generation ratio of the data decreases.

The latency: we slice the time into small slots, and want to find out how much data has been successfully delivered in each slot. The simulation shows that the transmission speed also follows the same order as above, that is, compare-priority, compare-deadline, 1-hop-priority, 1-hop-deadline, 2-hop-priority, and 2-hop-deadline. This means that our proposed algorithm not only delivers more data, but also does

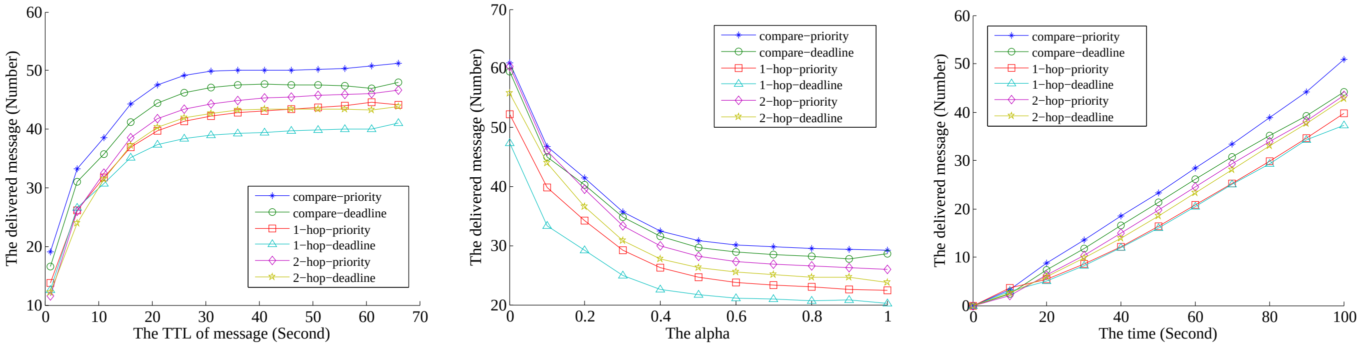


Fig. 8. Number of delivered data in the synthetic trace

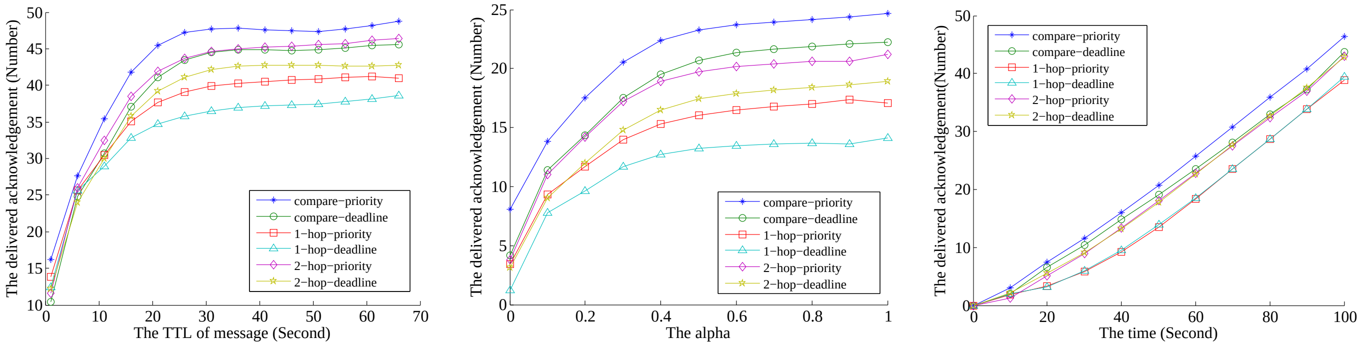


Fig. 9. Number of delivered acknowledgements in the synthetic trace

so at a high speed. In the *Infocom2006* trace, we notice that different contact estimations have a big influence on the number of messages that can be transmitted via the priority setting. However, the priority setting does not have such a big influence on the message delivery number. There also exists a big performance gap between the compare-priority, and the compare-deadline with the remaining four algorithms, which also show the importance of the weakly connected relationship with the destination. The compare-priority algorithm delivers more than 2 times the data and 3 times the acknowledgement of the 1-hop-deadline algorithms. In the synthetic trace, the difference between the 6 algorithms is not so clear; it is caused by the uniform setting of the encounter, so that the influence of the buffer exchange is not so large.

From the above simulation, we can clearly find that our proposed algorithm speeds up the data transmission in the network. At the same time, since data is transmitted faster, more data is transmitted into their destinations respectively, which on the other hand, increases the delivery ratio of acknowledgement in the network. The results show the importance of the indirect relationship with the destination.

VI. RELATED WORK

Routing in MSNs has attracted the attention of many people in the last few years; how to achieve good performance with little system consumption is a major concern. In [6], the author proposes the two-hop transitivity property. In [7], a comprehensive utility is proposed, which can reflect the encounter frequency, average contact period, total contact period, and shortest separation period, simultaneously. In [8, 9], the weighted degree of node is also considered as a criterion for buffer exchange, so the routing decision is based on several criteria. In [10] the author first points out the intrinsic characteristics of MSNs, and uses two utilities called centrality and community locally and globally to make routing decisions more precisely. Later, [11] proposed the centrality like PageRank, which is widely used in Internet searching. The idea is that your importance is decided by the importance of your neighbor. In [12], the author proposed a routing algorithm which considers the selfish characters of MSNs. It adjusts the utility by a factor called willingness. In this way, even though two nodes contact each other frequently, it might not be a good relay if its willingness is low. In [13], the author argues that most existing algorithms try to assign a majority of the workload on a few popular nodes, which is not fair and the resource of these nodes will soon be drawn up. A utility called assortativity is proposed to limit the system resource usage.

The acknowledgement scheme has also been extended into MSNs recently. However, most existing work is still focused on how to estimate the current network topology in a more accurate way. Such algorithms include [14–16]. A more challenging problem is whether we should trust the information from the acknowledgement in such an unsupervised environment. There might exist some selfish and malicious nodes in the network. In [17], the authors study the robustness of MSNs routing in the absence of authentication. The author identifies conditions for an attack to be effective, and present an attack based on a combination of targeted flooding and acknowledgement counterfeiting that is highly effective, even with only a small number of attackers. So a mechanism used to detect the attack is meaningful in MSNs. The results in [18] show that each node should forward the message which is most similar to its common interest, given an encounter between friends, or it should forward the message which is furthest to its common interest, given an encounter between the strangers. In [19], they propose a 2ACK scheme. The basic idea is that, when a node forwards a data packet successfully over the next hop, the destination node of the next-hop link will send back a special two-hop acknowledgment called 2ACK to indicate that the data packet has been received successfully.

The acknowledgement scheme is also used in an incentive mechanism; most existing methods introduce a credit-based scheme. Nodes get paid for providing services to other nodes. When they request other nodes to help them for packet forwarding, they use the same payment system to pay for such services. In Sprite [20], nodes keep receipts (acknowledgement) of the received/forwarded messages. When they have a fast connection to a Credit Clearance Service (CCS), they report all of these receipts. The CCS then decides the charge and credit for the reporting nodes. However, it is not practical to build such a CCS in MSNs, and the node's mobility pattern is heterogenous, so that some nodes might never meet with the CCS. In [21], the authors introduce the trading mechanism to the receipt, that is, nodes would like to exchange their messages and receipts on the condition that both of them can increase their expected probability to successfully cash the receipt after exchanging the receipt.

To the best of our knowledge, none of above algorithms consider the priority of buffer exchange. However, according to the research of [22], lots of data is generated in the MSNs, and the size of data increase at a high speed. So the contact opportunity is limited compared with the size of the messages.

VII. CONCLUSION

In this paper, we propose a general scheme for the data and acknowledgement transmission problem in mobile social networks. The contact opportunity is precious in MSNs, so the buffer constraint and the time constraint of data and acknowledgement is considered in our model. A general routing algorithm, *priority-based compare-split*, is proposed. This algorithm evaluates relays' abilities and the benefits of messages assigning the priority for buffer exchange, which thus maximizes achievable benefit. First, a new probability estimation scheme based on social status and contact probability is combined to evaluate the ability of relay nodes. Second, an adaptive priority-based exchange scheme is proposed within each type of message and the relative priority between different types of messages. Two major application scenarios are further studied and explained in this paper. Extensive simulations show that our algorithm achieves a high delivery ratio in a low latency, simultaneously. Our future work will focus on studying the situation where the size of messages is heterogenous. Besides, contact durations will be considered in the network mode. Multiple copies of data and acknowledgements are another objective for the future work.

REFERENCES

[1] K. Fall, "A delay-tolerant network architecture for challenged internets," in *Proceedings of the ACM SIGCOMM*, 2003, pp. 27–34.

[2] J. C. Tang, M. Cebrian, N. A. Giacobe, H.-W. Kim, T. Kim, and D. B. Wickert, "Reflecting on the darpa red balloon challenge," *Communications of the ACM*, vol. 54, no. 4, pp. 78–85, 2011.

[3] A. Kate, G. M. Zaverucha, and U. Hengartner, "Anonymity and security in delay tolerant networks," in *Proceedings of the IEEE SecureComm*, 2007, pp. 504–513.

[4] R. Chen, F. Bao, M. Chang, and J.-H. Cho, "Trust management for encounter-based routing in delay tolerant networks," in *IEEE GLOBECOM*, 2010, pp. 1–6.

[5] J. Scott, R. Gass, J. Crowcroft, P. Hui, C. Diot, and A. Chaintreau, "CRAWDAD data set cambridge/haggle (v. 2006-01-31)," Downloaded from http://crawdad.org/cambridge/haggle/, Jan. 2006.

[6] A. Lindgren, A. Doria, and O. Schelén, "Probabilistic routing in intermittently connected networks," *ACM SIGMOBILE mobile computing and communications review*, vol. 7, no. 3, pp. 19–20, 2003.

[7] F. Li and J. Wu, "Localcom: a community-based epidemic forwarding scheme in disruption-tolerant networks," in *IEEE SECON*, 2009, pp. 1–9.

[8] J. Wu and Y. Wang, "A non-replication multicasting scheme in delay tolerant networks," in *Proceedings of the IEEE MASS*, 2010, pp. 89–98.

[9] X. F. Guo and M. C. Chan, "Plankton: An efficient dtn routing algorithm," in *IEEE SECON*, 2013, pp. 550–558.

[10] P. Hui, J. Crowcroft, and E. Yoneki, "Bubble rap: Social-based forwarding in delay-tolerant networks," *IEEE Mobile Computing*, vol. 10, no. 11, pp. 1576–1589, 2011.

[11] A. Mtibaa, M. May, C. Diot, and M. Ammar, "Peoplerank: Social opportunistic forwarding," in *proceedingd of the IEEE INFOCOM*, 2010, pp. 1–5.

[12] Q. Li, S. Zhu, and G. Cao, "Routing in socially selfish delay tolerant networks," in *Proceedings of the IEEE INFOCOM*, 2010, pp. 1–9.

[13] J. M. Pujol, A. L. Toledo, and P. Rodriguez, "Fair routing in delay tolerant networks," in *Proceedings of the IEEE INFOCOM*, 2009, pp. 837–845.

[14] J. Burgess, B. Gallagher, D. Jensen, and B. N. Levine, "Maxprop: Routing for vehicle-based disruption-tolerant networks." in *Proceedings of the IEEE INFOCOM*, 2006, pp. 1–11.

[15] P. Mundur and M. Seligman, "Delay tolerant network routing: Beyond epidemic routing," in *IEEE ISWPC 2008*, 2008, pp. 550–553.

[16] R. Ramanathan, R. Hansen, P. Basu, R. Rosales-Hain, and R. Krishnan, "Prioritized epidemic routing for opportunistic networks," in *Proceedings of the ACM MobiSys*, 2007, pp. 62–66.

[17] F. C. Choo, M. C. Chan, and E.-C. Chang, "Robustness of dtn against routing attacks," in *Proceedings of the IEEE COMSNETS*, 2010, pp. 1–10.

[18] Y. Zhang and J. Zhao, "Social network analysis on data diffusion in delay tolerant networks," in *Proceedings of the ACM SIGMOBILE*, 2009, pp. 345–346.

[19] K. Liu, J. Deng, P. K. Varshney, and K. Balakrishnan, "An acknowledgment-based approach for the detection of routing misbehavior in manets," *IEEE Mobile Computing*, vol. 6, no. 5, pp. 536–550, 2007.

[20] S. Zhong, J. Chen, and Y. R. Yang, "Sprite: A simple, cheat-proof, credit-based system for mobile ad-hoc networks," in *Proceedings of the IEEE INFOCOM*, 2003, pp. 1987–1997.

[21] T. Ning, Z. Yang, H. Wu, and Z. Han, "Self-interest-driven incentives for ad dissemination in autonomous mobile social networks," in *Proceedings of the IEEE INFOCOM*, 2013.

[22] M. Cha, H. Kwak, P. Rodriguez, Y.-Y. Ahn, and S. Moon, "I tube, you tube, everybody tubes: analyzing the world's largest user generated content video system," in *Proceedings of the ACM SIGCOMM*, 2007, pp. 1–14.