

Rethink Data Forwarding in Mobile Social Networks using Movement History Information

NING WANG^{1*}, JIE WU¹, AND LI SHENG²

¹ *Department of Computer and Information Sciences, Temple University, USA*

² *Department of Mathematics, Drexel University, USA*

This paper studies data forwarding by using the node's spatial information in mobile social networks (MSNs). Specifically, we partition the 2D space into several grids, and periodically record the nodes' staying within each grid to extract their movement history summaries. Then, nodes' movement history summaries are used to compare their forwarding abilities in the single-copy scenario. In the multiple-copy scenario, we first address the dependent data forwarding path problem, i.e., different copies will reach the same relay with good forwarding ability, and thus the advantage of multiple-copy cannot be fully utilized. To avoid this, we jointly consider the nodes' forwarding abilities and their movement trajectories to perform copy distribution. Therefore, the potential overlap of multiple copies is minimized. In addition, we propose an extended scheme, which periodically records the nodes' transaction in grids. It improves the performance at the cost of more computation and storage consumption. Through extensive trace-driven experiments, proposed algorithms achieve good performance in different scenarios.

Key words: Mobile social network, delay tolerant network, opportunistic network, routing metric, data forwarding, algorithm.

* email: ning.wang@temple.edu

1 INTRODUCTION

Recently, the growing availability of personal mobile devices and new technologies, such as Wi-Fi Aware [1] and Wi-Fi Direct [27] has generated new communication techniques, such as mobile social networks (MSNs). A MSN is a special type of delay tolerant network (DTN) [23], in which mobile users walk around and communicate with each other via Bluetooth or WiFi in their carried short-distance wireless communication devices. The Cisco 2014-2019 White Paper [2] points out that the number of mobile-connected devices has exceeded the world's population in 2014, which has led to many potential contact opportunities in MSNs. White paper also forecasts that the global mobile data traffic will increase nearly tenfold between 2014 and 2019, and monthly global mobile data traffic will surpass 24.3 exabytes by 2019. With an increased amount of data traffic, the cellular networks will not satisfy the increasing traffic demand, especially during peak times in the densely populated area. As the result, users will face extreme performance degradation, such as low network bandwidth, missed voice calls, and unreliable coverage. The MSNs seem to be a promising technique of offloading the extensive traffic from the cellular network. The above-mentioned reasons make the MSNs very attractive in both academia and industry.

Currently, data forwarding in MSNs is still a challenging problem. The reason is that devices (nodes) are carried by people, and people's movements are opportunistic. Early routing algorithms [7, 34], evaluated the forwarding ability based on the encounter frequencies between nodes. The problem is that nodes have to update the forwarding table each time they meet, so that nodes must continue recording and computing the forward metric; this causes considerable storage computation and consumption. Therefore, social-based data forwarding algorithms [6, 11, 16, 32] have become a new trend. The common critical assumption of the social-based algorithms is that there are many social communities, e.g., classmates, interest group, in MSNs, and nodes within the same community have high opportunities for contact with one another. However, how to extract the social community from the MSNs itself is non-trivial. These algorithms also suffer from the following problem. Nodes might belong to multiple (possibly overlapping) communities, which causes difficulty in evaluating the nodes' forwarding abilities.

To address the challenge routing problem in MSNs, in this paper, we re-think the data forwarding in MSNs by using the nodes' movement spatial information. Specifically, the 2D space is partitioned into several grids. Consider a 2D space in Fig. 1(a): this 2D space is divided into 9 grids,

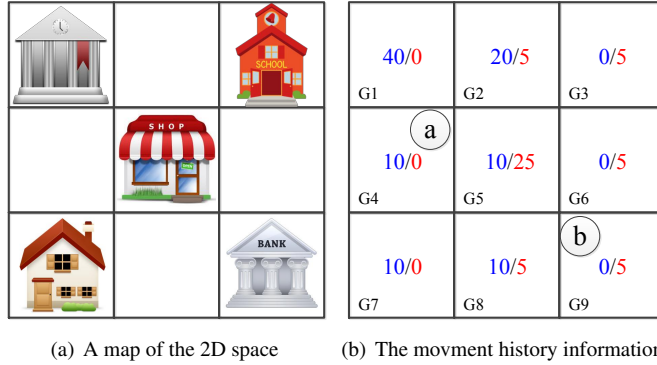


FIGURE 1

A 2D space is partitioned into 9 grids. A library, a school, a retail store, a house, and a bank are located in G_1 , G_3 , G_5 , G_7 , and G_9 , respectively. In Fig. 1(b), Node a and b 's movement history summaries are recorded in red and blue numbers, respectively. In particular, the number in each grid indicates the amount of time that they stayed.

G_1, G_2, \dots, G_9 . Each node in the network records the grid in which it currently stays, based on units of time, such as one hour. As time goes by, each node maintains a table, called the movement history summary, which extracts the node's movement history information, as shown in Fig. 1(b).

By using the movement history summary, each node can calculate the following two factors distributively: the node's *mobility pattern*, the fraction of time that a node stayed in each grid, and the node's *active level*, the total time that a node stayed in this space, to evaluate a node's forwarding ability. In the example shown in Fig. 1(b), the movement history summaries of nodes a and b are shown in red and blue numbers, respectively. Based on them, we can calculate the nodes' mobility patterns and active levels. Nodes a and b 's mobility patterns are $(0.4, 0.2, 0, 0.1, 0.1, 0, 0.1, 0.1, 0)$, and $(0, 0.1, 0.1, 0, 0.5, 0.1, 0, 0.1, 0.1)$, respectively. The active levels of nodes a and b are 100 and 50, respectively.

These two factors represent the node's forwarding ability in two dimensions. The mobility pattern can be used to evaluate the relationship of two nodes. If two nodes have similar mobility patterns, it means that they are highly likely to move into the same grid. On the other hand, the active level represents the inter-meeting delay of encountering good relays for the desti-

nation. It can be regarded as the potential of the node. For example, if the destination node always stays at G_5 , nodes a and b stay for 60% and 80% fraction of their time in G_5 , respectively. However, node a spends two more hours in G_5 than node b . Considering the fact that node a has more opportunity than node b of meeting other good relays or the destination itself, there exists a trade-off between the mobility pattern and the active level in data forwarding; we should consider these two factors jointly.

In the multi-copy uni-cast scenario, how to distribute the copies in the network so that the delay to the destination is minimized is another non-trivial problem. From the view of the destination, it only cares about the first arrived copy. A good strategy is to distribute all the copies to as many relays as possible; in this case, the probability that one of the copies arrives the destination is maximized. However, if nodes just forward copies based on the forwarding ability, multiple copies might reach the same relay in the end, and thus the copies will not be distributed to other nodes, As a result, the advantage of multiple copies decreases. This challenge is called the *dependent forwarding path* problem in this paper. To address it, we propose an independent path routing scheme, which jointly considers the encounter node's forwarding ability and movement direction. The idea is that we would like to control copies' forwarding directions, in the hope that their forwarding paths are independent.

The major contribution of this paper is three-fold:

- We re-think the data forwarding problem by using movement history information, which is easily implemented. It avoids the drawback of the social-based algorithms and consumes few resources.
- We extract the mobility pattern and the active level from the movement history information to evaluate the node's forwarding ability. Later, we extend it by recording the grid transaction history. These two approaches can be applied into different application scenarios, where the former approach can achieve relatively good performance with little buffer consumption, and the latter approach can achieve even better performance with a little more resource consumption.
- We address the dependent forwarding path problem in the multi-copy uni-cast scenario by jointly considering the node's forwarding path. Therefore, the data forwarding performance improves.

The remainder of the paper is organized as follows. The related works are in Section II. The model and problem formulation are introduced in Section

III. Then, the routing algorithm based on nodes' staying history is provided in Section IV. The extended algorithm, which uses nodes' grid transaction information, is provided in Section V. The performance evaluation setting and results are shown in Section VI. The acknowledgment and the conclusion of the paper are in Section VII and Section VIII, respectively.

2 RELATED WORKS

In this section, we capture some important issues arising from the data forwarding scheme in MSNs [17, 22, 29, 30] and motivate the work in this paper.

2.1 Social-based Routing in MSNs

The idea in social-based routing [6, 11, 18, 32, 35, 38] is to design routing algorithms which are based on the invariable metric in MSNs. The common critical assumption of the social-based algorithms is that there are many social communities, e.g., classmates, interest groups, in MSNs, and nodes within the same community have a high contact opportunity with each other. SimBet [6] used the betweenness centrality and social similarity to increase the probability of a successful data forwarding. Authors in [6] show that it performs well, especially when the connectivity is low. Bubble rap [11] made use of node centrality and the weighted k-clique community structure to enhance delivery performance. However, it has the same problem as [6], i.e., using betweenness to calculate the global and local centrality, not considering the node encounter probability. In [24], the authors propose a distributed community detection algorithm. They suffer the drawback of the social community partition, since they do not consider the situation where a node belongs to multiple communities. In [18, 32], they mapped the nodes into a hypercube, based on their social features, and then they did routing at the hypercube. However, how to extract the social community from the MSNs is non-trivial. Besides, how to assign a proper weight for different communities during the routing decision is challenging.

2.2 Geo-routings in MSNs

The main idea of the geo-routing in MSNs [4, 9, 13, 15, 28, 31, 37] is that the relay will forward the message to the encountered node, which is geographically the closest to the destination. Initially, they only consider the geometric distance from the current nodes to the destinations, followed by the Mobility Prediction-based Adaptive Data (MPAD) [36], Greedy Perimeter Stateless Routing (GPSR) [14], and Geographic Source Routing (GSR) [21], which

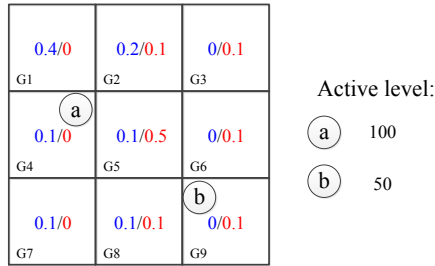


FIGURE 2

An illustration of the network model, where the space is divided into 9 grids. There are two nodes in this example; the blue number and the red number in each grid represents the fraction of time that they stayed in each grid for nodes a and b , respectively.

use more information. The problem with these algorithms is that they are all under the assumption that they know the destination node's location. However, in MSNs, this assumption might not be true. In [5, 33], the entire area is also split into sub-areas, like the grid in this paper, and each node summarizes its mobility information as transient sub-area visiting record. Upon arriving at a sub-area, a node generates a new visiting record for the sub-area and distributes it to nodes that are likely to stay in the previous sub-area, so that visiting records form a chain for the locators to trace the node. In our paper, we do not have the concept of locators. Therefore, the main solution is different. The drawback of using locators is that the situation might arise in which the current node cannot find a node which stays in the previous sub-area. Besides, a lot of information is exchanged, and each node might buffer many records.

In this paper, instead of using the social-based concept, we use movement history information to estimate the relay's forwarding ability. Therefore, the drawback of social-based routing is avoided. On the other hand, the proposed scheme consumes little computation and buffer resource.

3 NETWORK MODEL AND PROBLEM

3.1 Network Model

We consider a 2D space (e.g., center city), which is partitioned into several grids, (e.g., school, home, store, and office). A number of people (nodes)

move in this 2D space. Different people visit different grids with different frequencies. Also, different people might stay different lengths of time in these grids. The movement history information reveals lots of useful information to guide the routing. Consider students in a university environment: PhD students usually stay in their dormitories and laboratories, while the undergraduate students spend more time within their classrooms, the student centers and gyms. If you want to send word to a PhD student, a good option is to ask someone who is frequently in the laboratories, and is frequently at the university for help. The real applications include mobile advertisement dissemination [19, 25], mobile publish/subscribe system [8, 10].

More specifically, k mobile nodes, $V = \{1, 2, \dots, k\}$, move independently on a 2D space, which is partitioned into m grids, denoted by G , $G = \{G_1, G_2, \dots, G_m\}$. An example is shown in Fig. 1(a), a 2D space is partitioned into 9 grids, G_1, G_2, \dots, G_9 . Among these 9 grids, a library, a school, a retail store, a house, and a bank are located in G_1, G_3, G_5, G_7 , and G_9 , respectively. Note that the grid can be any shape and the area of grids can differ in reality. In addition, there might be the case in which more than one places of interests. For example, a school and a library are located in one grid. It is equivalent as there are several overlapping grids, and there is only one place of interest in each grid. We assume that nodes know their locations, which is reasonable, since all the newly-made smartphones have Global Positioning System (GPS) function. Thus, every node knows the grid in which it currently stays, and it continuously records its location every unit of time. Along with the time, each node maintains a vector, called a movement history summary. Specially, for node a , it is $\{t_1^a, t_2^a, \dots, t_m^a\}$, $\sum_{i=1}^m t_i^a = T_a$, where t_i^a indicates the time that the node a stayed at G_i in the past period of time. In Fig. 1(b), nodes a and b have movement history summaries recorded by the red and blue numbers, respectively. By using the movement history summary information, we can extract the following two types of information.

Definition 1. Mobility pattern: We define a mobility pattern of a node $M = \{p_1, p_2, \dots, p_m\}$ for each mobile node, where p_i denotes the probability of the node at the grid G_i , $p_i = \frac{t_i}{T}$.

Definition 2. Active level: We define the total time, T , that a node stays in the 2D place in a period of time as the active level of this node.

We assume the node will exchange its movement history summary with every encountered node, and buffer this meta information. Note that, along with the time, the movement history summary of a node will be stable. This

phenomenon is called the cyclic MobiSpace [20]. Along with the time, each node will have mobility patterns of all the nodes in this area. As for the grid partition, if each grid is too large, we cannot partition the nodes well, so that we cannot judge whether a node has a good relationship with the destination or not. In the extreme case, the whole 2D space is a grid; we cannot judge whether a node has a close relationship with the destination at all. If the grid is too small, two nodes with a close relationship cannot be efficiently selected. Besides, in a dense partition environment, each node has to keep much more information. In this paper, we assume that nodes in the same grid can contact one another. A proper area of the grid should be the transmission range of nodes or the WiFi coverage area. In this case, once two nodes come into the same grid, they can communicate with each other.

3.2 Problem Formulation and Challenges

In this paper, we focus on the data forwarding problem in MSNs, with the movement history information. We plan to address the following two problems gradually:

- What is the efficient method of completing data forwarding in a single-copy uni-cast scenario?
- What is the efficient method of completing data forwarding in the multiple-copy uni-cast scenario?

The above two scenarios present major challenges. The first challenge lies in balancing the mobility pattern and active level when two nodes meet. A similar mobility pattern indicates a close relationship with the destination. However, a high active level indicates low delay in meeting the destination, or the high chance to meet the other good relays. The second challenge is copy distribution in the multiple-copy scenario. Former works, which only consider the forwarding ability to distribute copies, are always under an assumption that different copies will be sent to the destination through different relays. In reality, several copies might gradually end in one good relay node. Therefore, multiple copies cannot be fully distributed in the network, and therefore, the advantage of multiple-copy decreases.

4 GRID-HISTORY-BASED SCHEME

In this section, we first solve the single-copy uni-cast problem by using the node's grid staying history information. Then, we address the multiple-copy uni-cast problem. The dependent path problem in the latter problem is solved.

Algorithm 1 MA

Input: The mobility pattern, active level of the current node, c , other nodes in this grid v_i , and the destination, d , respectively.

Output: The forwarding strategy of the current node.

- 1: **for** each encountered node v_i **do**
 - 2: Calculate S_{cd} , $S_{v_i d}$, T_a , and T_{v_i}
 - 3: **if** $S_{v_i d} \cdot T_{v_i} > S_{cd} \cdot T_a$ **then**
 - 4: add v_i to the set $Temp$
 - 5: **if** $Temp \neq \phi$ **then**
 - 6: Sort the v_i in set $Temp$ in descending order, according to the value of the $S_{v_i d} \cdot T_{v_i}$.
 - 7: Choose the largest v_i from $Temp$ to forward data.
 - 8: **else**
 - 9: Node c maintains the copies.
-

4.1 Single-copy Uni-cast

In this subsection, we would like to jointly consider the mobility pattern and the active level of the nodes to do data forwarding in the single-copy scenario.

To understand the influence of the mobility pattern and the active level to the data forwarding, we do the probability analysis on a general case here. Assume nodes a and b move freely in a 2D space, which is divided into m grids. Nodes a and b can contact each other once they move into the same grid. During each move, the probability that nodes a and b move into each grid is $\{p_{a1}, p_{a2}, \dots, p_{am}\}$, and $\{p_{b1}, p_{b2}, \dots, p_{bm}\}$, respectively. The probability that they meet during a movement of a is $\sum_{i=1}^m p_{ai}p_{bi}$. Besides, node a moves every $\frac{T}{T_a}$ seconds on average, where T is the total time. Then, we can get the following estimation: the expected time that node a moves into node b 's grid is

$$\sum_{i=1}^{\infty} \frac{iT}{T_a} (1 - \sum_{j=1}^m p_{aj}p_{bj})^{i-1} \sum_{j=1}^m p_{aj}p_{bj} = \frac{T}{T_a \sum_{j=1}^m p_{aj}p_{bj}}, \quad (1)$$

where the $(1 - \sum_{j=1}^m p_{aj}p_{bj})^{i-1} \sum_{j=1}^m p_{aj}p_{bj}$ represents the probability that node a moves into node b 's grid, until the i^{th} movement. It is clear that the $\sum_{i=1}^m p_{ai}p_{bi}$ is the inner production of the mobility pattern of two nodes. From Eq. 1, we can get the following observations: a similar mobility pattern leads to a small delivery delay; a high active level leads to a small delivery delay; the overall influence of these two factors are their product result.

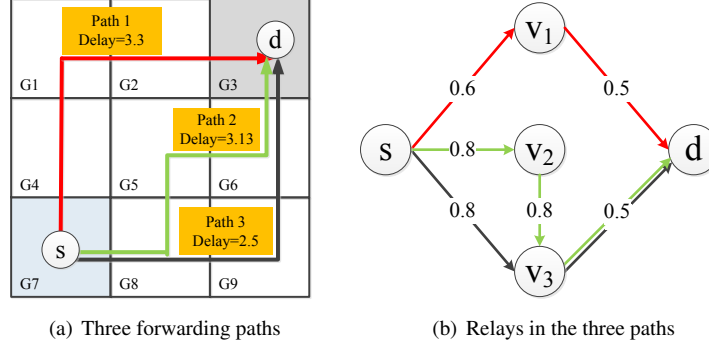


FIGURE 3

An example of the dependent forwarding path in multi-copy scenario, where the source is at the left-bottom grid, and the destination is at the top-right grid, respectively. There exist three forwarding paths from the source to the destination. Among paths 2 and 3, they actually use the same relay.

Definition 3. Movement similarity: Given two mobility patterns of nodes a and b $M_a = \{p_{a1}, p_{a2}, \dots, p_{am}\}$, $M_b = \{p_{b1}, p_{b2}, \dots, p_{bm}\}$, we define the movement similarity between node a and b as S_{ab} , having $S_{ab} = M_a \cdot M_b$, where the symbol \cdot denotes the inner product of vectors.

Based on the observations from Eq. 1, the nodes can complete routing decisions locally. Each time, two nodes compare the product of the movement similarity with the destination and their own active level. Then, the copy will always be kept to the node which has a larger result. In Fig. 2, the mobility pattern of node a , M_a is $\{0.4, 0.2, 0, 0.1, 0.1, 0, 0.1, 0.1, 0\}$, and the mobility pattern of node b , M_b is $\{0, 0.1, 0.1, 0, 0.5, 0.1, 0, 0.1, 0, 1\}$, respectively. The active levels of nodes a and b are 100 and 50, respectively. Assume the mobility pattern of destination, M_d , is $\{0.2, 0.1, 0, 0.1, 0, 0.4, 0.1, 0, 0.1\}$. Then, the movement similarity between nodes a and d , S_{ad} , is 0.16. The movement similarity between nodes b and d , S_{bd} , is 0.22. In this case, if we jointly consider the different active level of nodes a and b , we think node a is better. This is because $T_a \times S_{ad}$ is 16, which is larger than $T_b \times S_{bd}$, 11.

4.2 Multiple-copy Uni-cast

In this subsection, we extend our data forwarding scheme into the multiple-copy scenario, in which the dependent forwarding path problem exists.

TABLE 1
Expected delay in different cases

Strategy	1	2	3	4	5	6
Path 1	√			√	√	
Path 2		√		√		√
Path 3			√		√	√
Expected delay	3.3	3.13	2.5	2.17	1.79	1.91

A motivational example is provided in Fig. 3. In this example, the current node, s , which has two data copies, is connected with three other nodes, v_1 , v_2 , and v_3 . These three nodes all have better forwarding abilities than the current node, and the three shortest delay forwarding paths from the current node to the destination are shown in red, green and black, respectively. Fig. 3(a) shows the spatial location of the three paths, and Fig. 3(b) shows the actual relays in the three paths. In Fig. 3(b), the number in the link is the contact probability of two nodes. The three colors represent the corresponding three paths in Fig. 3(a). Same as most existing works [23, 25, 30], the path delivery probability in this paper is assumed to be the product of the contact probability of the nodes in this path. So, the path delivery probabilities of paths 1, 2 and 3 are 0.3, 0.32 and 0.4, respectively. Therefore, we get the expected delay of these three paths, 3.33, 3.13 and 2.5, respectively. If the copy distribution decision is made purely based on the expected delay of the path, the two forwarding paths with the lowest delay, paths 2 and 3, will be selected. However, this is far from the optimal. The following is a simple calculation. If the current node chooses the fastest two paths, paths 2 and 3, the probability that at least one copy will end in node d is $(1 - (1 - 0.8 \times 0.8)(1 - 0.8)) \times 0.8 = 0.46$. However, if we choose paths 1 and 3, the probability that at least one copy will end in node d is $1 - (1 - 0.3)(1 - 0.4) = 0.56$. Then, we can calculate the corresponding expected delivery delay. The detailed result is shown in Table. 1. From this table, we get the observation that if we just choose the relay according to its forwarding abilities, its performance might be really poor. The reason is that the forwarding paths are not independent of each other, and thus the new forwarding path might bring less contribution than expected.

The following is a theoretical analysis about performance degradation caused by the the dependent path. Assume that there exist x paths from source to des-

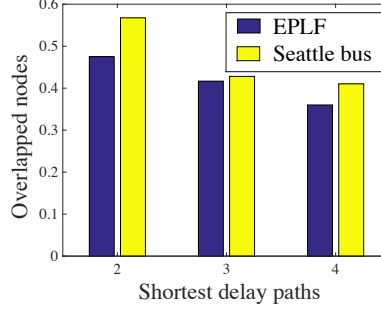


FIGURE 4

An illustration of dependent path problem in two real data sets, where the 2, 3, 4 in x axis means the 2nd, 3rd, and 4th shortest paths. This figure shows the percentage of overlapping nodes with the former shortest paths.

mination, and each path's length is y . The contact probability between nodes is identical, p . If these x paths are independent of one another, the expected path delivery probability by using these x paths is $1 - (1 - p^y)^x$. If the z out of y lengths of continuous paths are dependent upon other paths, the remaining paths are separated, by the continuous path, into paths with lengths, i and j . The expected path delivery probability is

$$(1 - (1 - (1 - p^i)^x)(1 - (1 - p^j)^x))p^z, \quad (2)$$

where $i + j$ equals $y - z$. Eq. 2 is smaller than $(1 - (1 - p^{(y-z)})^x)p^z$. Then the performance decay caused by the dependency is

$$\frac{(1 - (1 - p^{(y-z)})^x)p^z}{1 - (1 - p^y)^x} \geq \frac{(1 - (1 - p^y)^x)p^z}{1 - (1 - p^y)^x} = p^z, \quad (3)$$

From Eq. 3, we conclude that the performance degradation increases exponentially along with the increasing of the fraction of the path dependency. We also test the shortest forwarding path routing under two real datasets, EPFL [26] and Seattle bus [12]. The results show that the second shortest forwarding path has about 40% dependency with the shortest forwarding path. The 4th shortest forwarding path still has about 30% dependency on average.

To avoid the performance degradation caused by the path dependent problem, the idea in this paper is to control the forwarding directions of the copies,

Algorithm 2 MMA

Input: The mobility pattern, active level of current node, c , other nodes in this grid v_i , and the destination, d , respectively. The initial copy number, n , the current copy number, n_c

Output: The forwarding strategy of the current node.

- 1: Find the top l frequently visited grids, $\{p_{d1}, p_{d2}, \dots, p_{dl}\}$ of the destination node. Assign $n \times \lceil \frac{p_{dj}}{\sum_{i=1}^k p_{di}} \rceil$ copies to each grid from the most frequently-visit grids until done.
 - 2: **for** each encountered node v_i **do**
 - 3: Calculate the candidate grids C of each copy, and the movement probability of v_i to C , p_i^c , to each C_i .
 - 4: **if** $S_{v_i d} \cdot T_{v_i} > S_{cd} \cdot T_c$ && $\exists p_i^c > \theta$ **then**
 - 5: add v_i to the set $Temp$
 - 6: **if** $Temp \neq \phi$ **then**
 - 7: Sort the v_i in set $Temp$ in descending order, according to the value of the $S_{v_i d} \cdot T_{v_i}$.
 - 8: Choose the largest v_i from $Temp$ to forward data.
 - 9: Forward $n_c \times \lceil \frac{S_{v_i d} \cdot T_{v_i}}{S_{cd} \cdot T_c + S_{v_i d} \cdot T_{v_i}} \rceil$ copies.
 - 10: **else**
 - 11: Node c maintains the copies.
-

in hopes that copies are distributed to multiple independent paths. As a result, the advantage of multiple copies can be fully used. However, due to the opportunistic movement, estimations of the forwarding path in MSNs are unrealistic. Therefore, an approximately independent path routing control scheme, which controls the next hop movement of multiple copies is proposed here. The algorithm is called as MMA algorithm, and is briefly shown as follows.

- The source selects the l grids, which the destination node visits most, and assigns the proportional number of copies to these l grids, according to the corresponding mobility pattern. For each data copy, it maintains a candidate grid set, in which the grids are closer to the assigned grid than the current grid.
- If the current node encounters another node with a better forwarding ability than the current nodes, and it has a high probability (larger than a threshold, θ) of moving into any candidate grid, the copies are split

according to their forwarding abilities. After that, this candidate grid is deleted from the candidate grid set.

In this scheme, the relay jointly considers the other nodes' forwarding abilities and their movement directions. So, the copies are distributed evenly, leading to a smaller probability of ending with dependent paths.

5 GRID-TRANSACTION-BASED SCHEME

In this section, we extend the mobility pattern based scheme by asking the node to record more information. That is, the node's grid transaction history is recorded. This approach improves the performance by requiring the node to record more information and to do more computation. Therefore, it can be applied to the scenario in which forwarding delay is much more important than the resource consumption.

The idea of this approach is that each node detects the grid in which it stays, and the grid in which the node currently stays is compared with the grid in which it stayed in the last detection. Then, the node records the transition between these two detections. As a result, each node will record a matrix, called the *grid transition matrix*, as shown in Table 2, where the number in each entry is the time at which the node transitioned from one grid to another. In this example, a node's movement history is 1 – 3 – 3 – 1 – 3 – 3 – 3 – 3 – 1 – 1 – 2 – 1 – 2 – 2 – 3, where 1 – 3 indicates that the node stayed at grid 1 during the last detection, and it stayed at grid 3 at the next detection. Then, this mobility model can be thought of as a Markov model, where a possible state transition is made during each time period. The different grids can be regarded as different states. A movement can be regarded as the state transition, respectively.

We also extract two types of information from the grid transition matrix. That is, the *grid transition probability matrix*, P , and the active level of the node. If every entry in the grid transition matrix is divided by the summation of the row in which the entry is, we get the grid transition probability matrix. An example is shown in Table 3. The entry p_{ii} is the probability that the node will stay at the current grid, and p_{ij} indicates the probability that the node moves from grid i to another grid j . In this example, once this node was at G_1 , it has 20%, 40%, and 40% probability of moving into G_1 , G_2 , and G_3 . The grid transition probability matrix can be used to estimate the node's movement from the current grid. The definition of the active level of the node is the same as we have discussed in Section 4.1.

TABLE 2
Grid transition matrix

Grid	1	2	3
1	1	2	2
2	1	1	1
3	2	0	4

We can use the above two types of information to estimate the forwarding ability of nodes. Let us denote the expected delivery delay D_{ij} of a node that arrives at G_j from G_i , which can be regarded as the state transition from i to j at the first time in the grid transition probability matrix. Therefore, we make state j an absorbing state and other states transition states. Then, if a node transits from a transition state to another transition state, this transition will cause delay. If a node transits from a transition state to the absorbing state, the forwarding is done. Now, we can extract the *transition to transition* matrix, P_T , that specifies only the transition probabilities from transition states. In the table, we just delete the row and the column of the absorbing state to get the matrix, P_T . An example is shown in Tables 3 and 4. In the example, we use G_3 as the destination grid. Based on the Markov transaction, the probability of transitioning from G_i to G_j in exactly h steps is the a (i, j) -entry of P_T^h . Summing this for all h (from 0 to ∞) yields the expected delay matrix, E .

$$E = \sum_{h=1}^{\infty} P_T^h = (I - P_T)^{-1},$$

where the matrix I is the identity matrix. In our example, matrix E , is shown in Table 5. By using matrix E , we can estimate the delay from the current grid i to the destination grid, which is the summation of the row i of E . For example, if the current node stays at G_1 , this node is expected to stay at G_1 1.66 units of time, and stay at G_2 0.99 units of time. That is, the expected delay from G_1 to G_3 is 2.65 units of time. Similarly, if the current node stays at the G_2 , the expected delay from G_2 to G_3 is 2.80 units of time.

The forwarding ability is evaluated based on the product of the average expected delay to the top l grids of the destination and the active level of the

TABLE 3
Grid transition probability matrix, P

Grid	1	2	3
1	0.2	0.4	0.4
2	0.33	0.33	0.33
3	0.33	0	0.67

TABLE 4
Transition to transition matrix, P_T

Grid	1	2
1	0.2	0.4
2	0.33	0.33

TABLE 5
Expected delay matrix, E , for G_3

Grid	1	2
1	1.66	0.99
2	0.82	1.98

current node. Based on the matrix, E , we propose another two algorithms in single-copy and multiple-copy scenarios called MA2 algorithm and MMA2 algorithm, respectively. In the single-copy scenario, the relay node make a forwarding decision based on expected forwarding delays. For the multiple-copy scenario, we continue to use the idea in Section 4.1. The idea is that

Algorithm 3 MA2

Input: The grid transition matrix of the current node, other nodes in this grid, and the destination node.

Output: The forwarding strategy of the current node.

- 1: Same as Algorithm 1, except the changes of S_{cd} and $S_{v_i d}$: S_{cd} and $S_{v_i d}$ are changed into the expected average D_{cd} and average $D_{v_i d}$, attained from the corresponding expected delay matrix, E .
-

Algorithm 4 MMA2

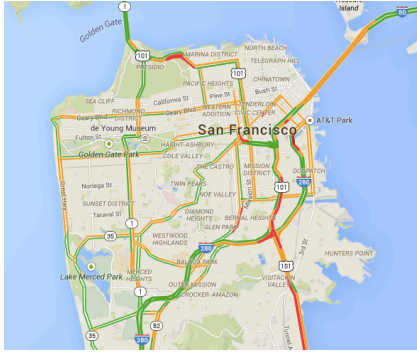
Input: The grid transition matrix of the current node, other nodes in this grid, and the destination node.

Output: The forwarding strategy of the current node.

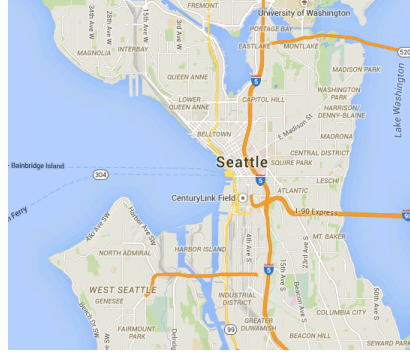
- 1: Same as Algorithm 2, except the changes of S_{cd} and $S_{v_i d}$: S_{cd} and $S_{v_i d}$ are changed into the expected average D_{cd} and average $D_{v_i d}$, attained from the corresponding expected delay matrix, E .
-

we would like to distribute the copy through different forwarding paths with low dependency. The current relay first compares the forwarding ability of the encounter node. Then, it checks the one-hop movement direction of the encounter node and its possible next movement. The current node will distribute copies to the encounter node only when the encounter node has a good forwarding ability, and has a high probability of moving into the grid, where no other relays exist.

A brief complexity analysis of the MA algorithm and MA2 algorithm is as follows. As for the buffer consumption, in the MA algorithm, each node consumes $O(m)$ to store the movement history information, but the MA2 algorithm consumes $O(m^2)$. Note that the storage consumption is only related to the number of grids, but not to the node number. The advantage of the two algorithms will be significant when the number of nodes is large. For the computation cost, the MA algorithm needs $O(m)$ to calculate the forwarding ability, while the MA2 algorithm needs $O(lm^2)$. Note that after the warm-up period, each node might regard the mobility pattern, active level, and the grid transition matrix as a constant value. In this case, each node can choose the store $O(k)$ forwarding ability values to get $O(1)$ computation. For the MA2 algorithm, it needs to store $O(km^3)$ forwarding ability values to get $O(1)$ computation. From the above analysis, there exists a performance-cost trade-off of these two algorithms. Therefore, the MA algorithm is proper for the scenario, where the system resource is limited. While the MA2 algorithm can

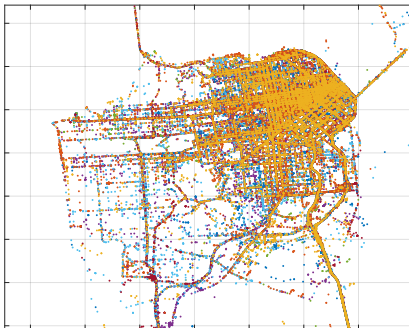


(a) Map of San Francisco



(b) Map of Seattle

FIGURE 5
Map of cities in EPFL and Seattle bus traces.



(a) Movement history of taxis



(b) Movement history of buses

FIGURE 6
Vehicles' movement history in EPFL and Seattle bus traces.

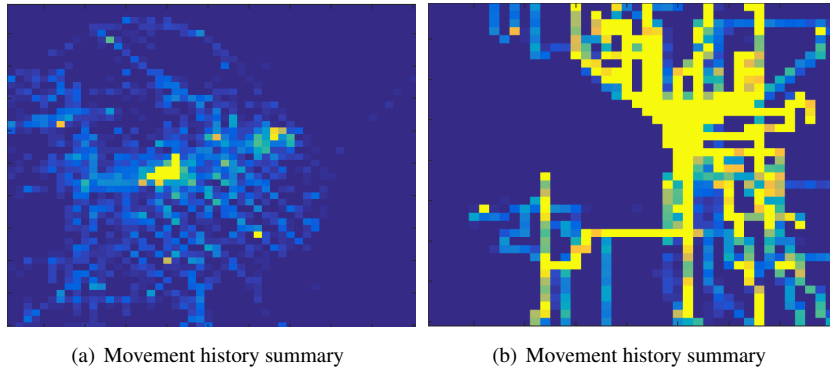


FIGURE 7
Vehicles' movement history summary in EPFL and Seattle bus traces.

be applied into the scenario which forwarding delay is much more important than the resource consumption.

6 PERFORMANCE EVALUATION

In this section, we compare the proposed algorithms mentioned in this paper through extensive experiments. We first introduce the experimental settings and their parameters. Then, we will discuss the performance evaluation results.

6.1 Trace Introduction

Here, we use EPFL [26] trace, which is the taxi trace collected from San Francisco, USA. It contains GPS coordinates of approximately 500 taxis collected over 30 days in the San Francisco Bay Area. Another trace that we use is the Seattle bus [12] trace. The traces collected the bus data while on different routes in Seattle, USA for several weeks. This trace provides a topographically challenging routing environment, created by a 35 square mile section in the middle of the city. The reason that we choose these two traces is that they represent two types of movement; the taxis' movements are very unpredictable, and the buses' movement are relatively predictable.

Some detailed experiment parameters are as follows: we choose center city within these two cities, $10,000 \text{ (ft)} \times 10,000 \text{ (ft)}$, as the experiment area

in these two traces. The grid size is $200 \text{ (ft)} \times 200 \text{ (ft)}$, which is the typical WiFi range under 2.4 GHz in 802.11 protocol for an outdoor environment [3]. Since we only know the location of each vehicle through GPS information in these two traces, we assume that if two vehicles move into the same grid within a time duration in these two traces, it is a contact. In this experiments, the time duration are 60s and 6s, respectively in the two trace. In the experiment, we choose the first 40 taxis in EPFL trace, and we choose the 60 buses in the Seattle bus trace. Every 10 minutes, a data is randomly generated from these vehicles to a randomly-selected destination node.

6.2 Algorithm Comparison

Our comparison consists of three parts. First, we compare four algorithms to the proposed algorithm in the single-copy scenario. 1) The flooding algorithm: the number of copies are unlimited, once a node encounters another node with the copy, both nodes will get one copy. This algorithm, called BS in the experiments, is used as the benchmark, regarding the delay and the cost. 2) The active-only algorithm, called AO is that in which the forwarding ability is only based on the active levels of two nodes. 3) The mobility-pattern-only solution algorithm, called MO, is that in which the forwarding ability is only based on movement similarity with the destination of two nodes 4) the Proposed algorithm, called MA, is that where the forwarding ability is based on the product of their active levels and the movement similarities with the destination. The aim of this comparison is to demonstrate the effectiveness of the jointly-considered movement similarity and the active level. After that, we compare the MA with the MA2 algorithm. Third, we compare the MMA and MMA2, along with the algorithm MMSW which is the same as the MMA, except that MMSW only uses the forwarding ability to distribute the copies. We compare the above-mentioned algorithms in two aspects, the data forwarding delay and the number of relays used in the forwarding.

6.3 The Performance Results of the Single-copy Uni-cast

The result is shown in Figs. 8, 9 and 10. As for the forwarding delay, the proposed algorithm achieves the lowest delay, except for the flooding algorithm, followed by the AO algorithm. The MO algorithm's performance is the worst, which demonstrates the importance of the active level. In the EPFL trace, since the taxis move opportunistically, the active level is more important than the mobility pattern. In the Seattle bus trace, the performance differences of the AO algorithm and MO algorithm are small, since buses move with high regulation. The MA algorithm's performance is much better than that of AO

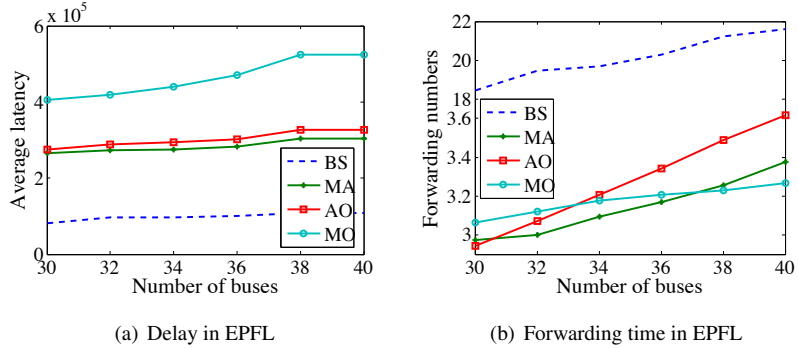
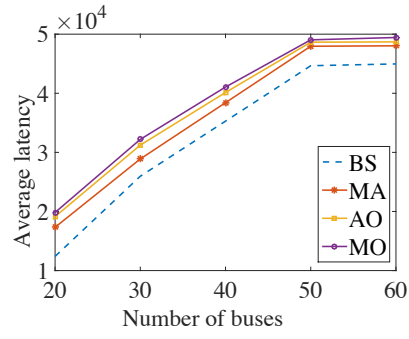


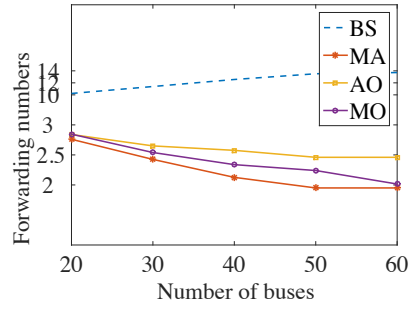
FIGURE 8
Performance evaluation of single-copy uni-cast in EPFL traces.

and MO, as shown in Figs. 8(a) and 9(a). In regard to the cost, the number of relays used in the forwarding, the flooding algorithm's cost is much larger than that of the other three algorithms. We have to adjust the Y axis to show it in Figs. 8(b) and 9(b). The MA algorithm achieves the lowest cost in a majority of the time, followed by the MO algorithm and AO algorithm.

Besides, we compare the two versions of our algorithms. The results are shown in Figs. 11 and 12. Among them, Figs 11(a) and 12(a) are the results of the EPFL trace, and Figs 11(b) and 12(b) are the results of the Seattle bus trace. From Fig. 11(a), we know that the MA2 algorithm achieves better performance than that of the MA algorithm, regarding the delivery delay. On the other hand, the number of relays used in the MA2 algorithm is much larger than that of the MA algorithm. The reason might be that the taxi has some mobility pattern, and the additional information in the MA2 helps significantly in estimating the delay. However, due to the movement of nodes, MA2 keeps calculating the current node's forwarding ability and the encounter nodes' forwarding abilities. The copies are forwarded much more frequently than in the MA algorithm. In the Seattle bus trace, the MA2 algorithm does not improve the performance much, the reason potentially being that the buses only move in a small range of the area, and they have to move along with the route. This shows that the mobility pattern has already extracted most of the information. As the result, their performances are similar. However, the MA2 algorithm avoids forwarding some data in the wrong direction, so that

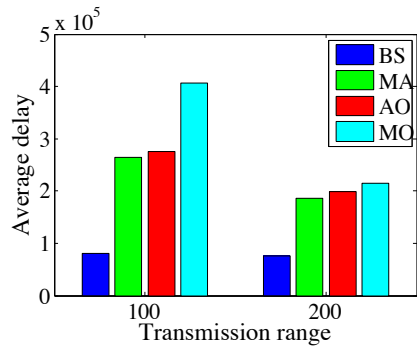


(a) Delay in Seattle bus

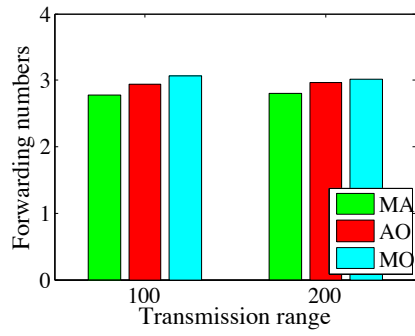


(b) Forwarding time in Seattle bus

FIGURE 9 Performance evaluation of single-copy uni-cast in Seattle bus traces.

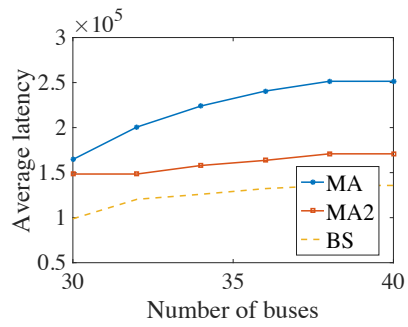


(a) Delay

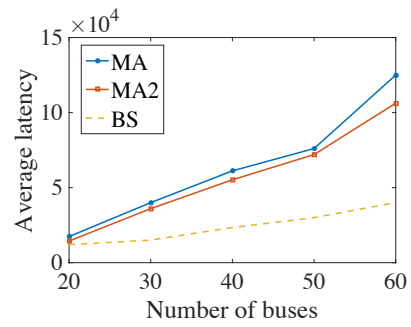


(b) Cost

FIGURE 10 Performance evaluation of single-copy uni-cast in different transmission ranges in EPFL trace.

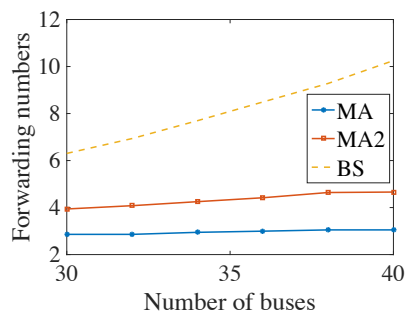


(a) Delay in EPFL

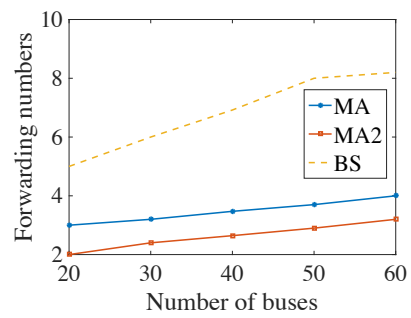


(b) Delay in Seattle bus trace

FIGURE 11
Performance comparison in single-copy uni-cast.

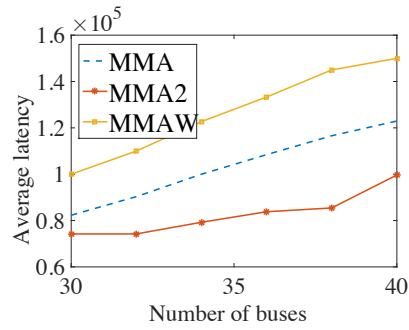


(a) Cost in EPFL

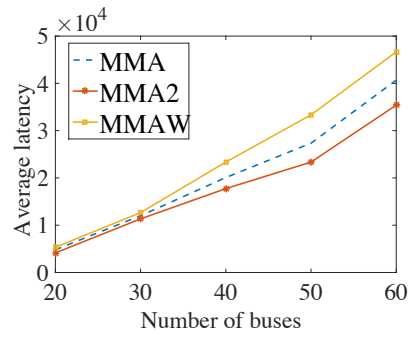


(b) Cost in Seattle bus trace

FIGURE 12
Performance comparison in single-copy uni-cast.

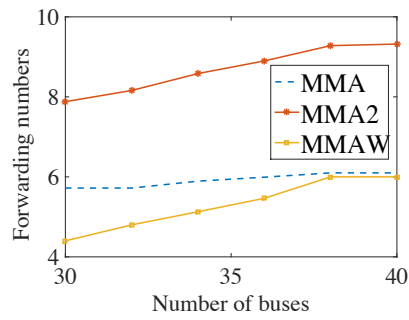


(a) Delay in EPFL

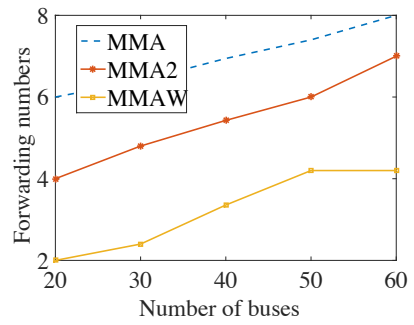


(b) Delay in Seattle bus trace

FIGURE 13
Performance comparison in multiple-copy uni-cast.



(a) Cost in EPFL



(b) Cost in Seattle bus trace

FIGURE 14
Performance comparison in multiple-copy uni-cast.

it forwards less frequently, as shown in Fig. 12(b).

6.4 The Performance Results of the Multiple-copy Uni-cast

We also demonstrate the performance degradation caused by the dependency paths of multiple copies. The results of the proposed algorithm are shown in Figs. 13 and 14. Among them, Figs. 13(a) and 14(a) are the results of the EPFL trace, and Figs. 13(b) and 14(b) are the results of the Seattle bus trace. As for the data delivery data, the MMA2 algorithm achieves the lowest delay, followed by the MMA algorithm and the MMAW algorithm. In the EPFL trace, the MMA2 algorithm's performance is much better than that of the MMAW algorithm. The reason might be found in Fig. 14(a). The copies are kept by few relays in the MMAW algorithm. However, in the MMA algorithm and MMA2 algorithm, the copies are distributed to more relays. As a result, there is an increased probability of one of the relays encountering the destination node. For the Seattle bus trace, the performance differences of these three are not as significant as those in the EPFL trace. The reason might be that each bus only contacts 2 to 3 buses in their routes. Therefore, if you distribute copies to other buses, it helps a little.

7 ACKNOWLEDGMENT

This research was supported in part by NSF grants CNS 1449860, CNS 1461932, CNS 1460971, CNS 1439672, CNS 1301774, and ECCS 1231461.

8 CONCLUSION

In this paper, we design an efficient data forwarding scheme in MSNs. The node's movement history information is used to guide data forwarding. Specifically, we partition the 2D space of the nodes into several non-overlapping grids, and record the nodes' stay in each grid to extract their movement history summary. Then, we extract the movement pattern and the active level of the node. Based on these two factors, we design a distributed data forwarding scheme in a single-copy uni-cast scenario. Then, we further extend it into the multiple-copy uni-cast scenario. We first address the performance degradation caused by the path dependency problem, and propose a distributed algorithm, which jointly considers the relay's forwarding ability and the other copies' distribution. Besides, we propose an extension, which achieves a better performance at the cost of more storage and computation. Through experiments, our algorithms achieve a low delay with few relays' help.

REFERENCES

- [1] <http://www.wi-fi.org/>.html.
- [2] http://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/white_paper_c11-520862.html.
- [3] http://en.wikipedia.org/wiki/IEEE_802.11.html.
- [4] Giuseppe Araniti, Nikolaos Bezirgiannidis, Edward Birrane, Igor Bisio, Scott Burleigh, Carlo Caini, Marius Feldmann, Mario Marchese, John Segui, and Kiyohisa Suzuki. (2015). Contact graph routing in dtn space networks: overview, enhancements and performance. *IEEE Communications Magazine*, 53(3):38–46.
- [5] Kang Chen and Haiying Shen. (2014). Dsearching: Distributed searching of mobile nodes in dtns with floating mobility information. In *Proceedings of the IEEE INFOCOM*.
- [6] Elizabeth M Daly and Mads Haahr. (2007). Social network analysis for routing in disconnected delay-tolerant manets. In *Proceedings of the ACM MobiHoc*.
- [7] Vijay Erramilli, Mark Crovella, Augustin Chaintreau, and Christophe Diot. (2008). Delegation forwarding. In *Proceedings of the ACM MobiCom*.
- [8] Younghwan Go, YoungGyou Moon, and KyoungSoo Park. (2012). Enabling dtn-based data offloading in urban mobile network environments. In *Proceedings of the ACM CFI*.
- [9] Ping He and Hong Shen. (2015). Efficient data retrieval algorithms for avoiding 2-slot conflicts in wireless data broadcast. *Adhoc & Sensor Wireless Networks*, 29.
- [10] Ólafur R Helgason, Emre A Yavuz, Sylvia T Kouyoumdjieva, Ljubica Pajevic, and Gunnar Karlsson. (2010). A mobile peer-to-peer system for opportunistic content-centric networking. In *Proceedings of the ACM SIGCOMM MobiHeld*.
- [11] Pan Hui, Jon Crowcroft, and Eiko Yoneki. (2011). Bubble rap: Social-based forwarding in delay-tolerant networks. *IEEE Transactions on Mobile Computing*, 10(11):1576–1589.
- [12] Jorjeta G Jetcheva, Y-C Hu, Santashil PalChaudhuri, Amit Kumar Saha, and David B Johnson. (2003). Design and evaluation of a metropolitan area multitier wireless ad hoc network architecture. In *Proceedings of the IEEE HotMobile*.

- [13] WANG JUN and GU WEI. (2016). A grid-based qos-aware routing protocol for wireless sensor networks. *Adhoc & Sensor Wireless Networks*, 32.
- [14] Brad Karp and Hsiang-Tsung Kung. (2000). Gpsr: Greedy perimeter stateless routing for wireless networks. In *Proceedings of the ACM MobiCom*.
- [15] Jérémie Leguay, Timur Friedman, and Vania Conan. (2005). Dtn routing in a mobility pattern space. In *Proceedings of the ACM SIGCOMM*.
- [16] Wen-Zao Li, Feng Lin, Ji-Liu Zhou, and Yan Wang. (2015). Dtn routing with fixed stations based on the geographic grid approach in an urban environment. *Wireless Personal Communications*, 82(4):2033–2049.
- [17] Yong Li, Li Qiu, Depeng Jin, Li Su, Pan Hui, and Lieguang Zeng. (2015). Contact duration aware evaluation for content dissemination delay in mobile social network. *Wireless Communications and Mobile Computing*, 15(3):527–537.
- [18] Zhong Li, Cheng Wang, Siqian Yang, Changjun Jiang, and Xiangyang Li. (2015). Lass: Local-activity and social-similarity based data forwarding in mobile social networks. *IEEE Transactions on Parallel and Distributed Systems*, 26(1):174–184.
- [19] Chia-Yu Lin, Zhi-Feng Jiang, Li-Chun Wang, and Bao-Shuh Paul Lin. (2014). An effective algorithm for interest aware opportunistic advertising by mining social and consuming information. In *Proceedings of the IEEE VTC*.
- [20] Cong Liu and Jie Wu. (2008). Routing in a cyclic mobispace. In *Proceedings of the ACM MobiCom*.
- [21] Christian Lochert, Hannes Hartenstein, Jing Tian, Holger Fussler, Dagmar Hermann, and Martin Mauve. (2003). A routing strategy for vehicular ad hoc networks in city environments. In *Proceedings of the IEEE IV*.
- [22] Namita Mehta and Mehul Shah. (2014). Performance of efficient routing protocol in delay tolerant network: A comparative survey. *International Journal of Future Generation Communication and Networking*, 7(1):151–158.
- [23] Vinícius FS Mota, Felipe D Cunha, Daniel F Macedo, José MS Nogueira, and Antonio AF Loureiro. (2014). Protocols, mobility models and tools in opportunistic networks: A survey. *Computer Communications*, 48:5–19.
- [24] Nam P Nguyen, Thang N Dinh, Sindhura Tokala, and My T Thai. (2011). Overlapping communities in dynamic networks: their detection and mobile applications. In *Proceedings of the ACM MobiCom*.

- [25] Ting Ning, Zhipeng Yang, Hongyi Wu, and Zhu Han. (2013). Self-interest-driven incentives for ad dissemination in autonomous mobile social networks. In *Proceedings of the IEEE INFOCOM*.
- [26] Michał Piórkowski, Natasa Sarafijanovic-Djukic, and Matthias Grossglauser. (2009). A parsimonious model of mobile partitioned networks with clustering. In *Proceedings of the IEEE COMSNETS*.
- [27] Alexander Pyattaev, Olga Galinina, Kerstin Johnsson, Adam Surak, Roman Florea, Sergey Andreev, and Yevgeni Koucheryavy. (2014). Network-assisted d2d over wifi direct. In *Smart Device to Smart Device Communication*, pages 165–218. Springer.
- [28] Minsu Shin, Seongik Hong, and Injong Rhee. (2008). Dtn routing strategies using optimal search patterns. In *Proceedings of the ACM CHANTS*.
- [29] Ning Wang and Jie Wu. (2014). Interests spread: an efficient method for content transmission in mobile social networks. In *Proceedings of the ACM MobiHoc MSCC*.
- [30] Kaimin Wei, Xiao Liang, and Ke Xu. (2014). A survey of social-aware routing protocols in delay tolerant networks: Applications, taxonomy and design-related issues. *IEEE Communications Surveys & Tutorials*, 16(1):556–578.
- [31] Hao Wen, Fengyuan Ren, Jia Liu, Chuang Lin, Pan Li, and Yuguang Fang. (2011). A storage-friendly routing scheme in intermittently connected mobile network. *IEEE Transactions on Vehicular Technology*, 60(3):1138–1149.
- [32] Jie Wu and Yunsheng Wang. (2012). Social feature-based multi-path routing in delay tolerant networks. In *Proceedings of the IEEE INFOCOM*.
- [33] Li Yan, Haiying Shen, and Kang Chen. (2015). Tsearch: Target-oriented low-delay node searching in dtns with social network properties. In *Proceedings of the IEEE INFOCOM*.
- [34] Peng Yang and Mooi Choo Chuah. (2006). Context-aware multicast routing scheme for disruption tolerant networks. In *Proceedings of the ACM MSWiM*.
- [35] Fei Yu and Victor Leung. (2002). Mobility-based predictive call admission control and bandwidth reservation in wireless cellular networks. *Computer Networks*, 38(5):577–589.
- [36] Jinqi Zhu, Jiannong Cao, Ming Liu, Yuan Zheng, Haigang Gong, and Guihai Chen. (2008). A mobility prediction-based adaptive data gathering protocol for delay tolerant mobile sensor network. In *Proceedings of the IEEE GLOBECOM*.

- [37] Yanmin Zhu, Ruobing Jiang, Jiadi Yu, Zhi Li, and Minglu Li. (2014). Geographic routing based on predictive locations in vehicular ad hoc networks. *Journal on Wireless Communications and Networking*, 2014(1):1–9.
- [38] Ying Zhu, Bin Xu, Xinghua Shi, and Yu Wang. (2013). A survey of social-based routing in delay tolerant networks: Positive and negative social effects. *IEEE Communications Surveys & Tutorials*, 15(1):387–401.



Ning Wang received his B.Eng. in Electrical Engineering from the University of Electronic Science and Technology of China, Chengdu, China in 2013. He is currently a 3rd year Ph.D. student in the Department of Computer and Information Sciences, Temple University, Philadelphia, PA, USA. His current research focuses on the delay tolerant networks and vehicular networks.



Jie Wu is the Associate Vice Provost for International Affairs at Temple University. He also serves as the Chair and Laura H. Carnell professor in the Department of Computer and Information Sciences. Prior to joining Temple University, he was a program director at the National Science Foundation and was a distinguished professor at Florida Atlantic University. His current research interests include mobile computing and wireless networks, routing protocols, cloud and green computing, network trust and security, and social network applications. Dr. Wu regularly publishes in scholarly journals, conference proceedings, and books. He serves on several editorial boards, including IEEE Transactions on Service Computing and the Journal of Parallel and Distributed Computing. Dr. Wu was general co-chair/chair for IEEE MASS 2006, IEEE IPDPS 2008, IEEE ICDCS 2013, and ACM MobiHoc 2014, as well as program co-chair for IEEE INFOCOM 2011 and CCF CNCC 2013. He was an IEEE Computer Society Distinguished Visitor, ACM Distinguished Speaker, and chair for the IEEE Technical Committee on Distributed Processing (TCDP). Dr. Wu is a CCF Distinguished Speaker and a Fellow of the IEEE. He is the recipient of the 2011 China Computer Federation (CCF) Overseas Outstanding Achievement Award.



Li Sheng received her MS degree in Statistics from Rutgers University, and received her PhD in Operations Research from Rutgers University in 1998; later that year, she also joined the school's Department of Mathematics. Her current research interests focus on discrete optimization, operations research, graph theory and its applications, and biostatistics.