The Journal of Systems and Software 83 (2010) 1301–1309

Contents lists available at ScienceDirect

The Journal of Systems and Software

journal homepage: www.elsevier.com/locate/jss

Improving routing protocol performance in delay tolerant networks using extended information

Xiao Chen [a,*], Jian Shen [b], Jie Wu [c]

[a] Department of Computer Science, Texas State University, San Marcos, TX 78666, United States
[b] Department of Mathematics, Texas State University, San Marcos, TX 78666, United States
[c] Department of Computer and Information Sciences, Temple University, Philadelphia, PA 19122, United States

ARTICLE INFO

Article history:
Received 14 September 2009
Received in revised form 12 January 2010
Accepted 13 January 2010
Available online 25 January 2010

Keywords:
Delay tolerant networks
Regression
Routing
Utility function

ABSTRACT

Delay tolerant networks (DTNs) are wireless mobile networks that do not guarantee the existence of a path between a source and a destination at any time. When two nodes move within each other's transmission range during a period of time, they can contact each other. The contact of nodes can be periodical, predictable and nonpredictable. In this paper, we assume the contact of nodes is nonpredictable so that it can reflect the most flexible way of nodes movement. Due to the uncertainty and time-varying nature of DTNs, routing poses special challenges. Some existing schemes use utility functions to steer the routing in the right direction. We find that these schemes do not capture enough information of the network. Thus, we develop an extended information model that can capture more mobility information and use regression functions for data processing. Experimental results from both our own simulator and real wireless trace data show that our routing algorithms based on the extended information model can increase the delivery ratio and reduce the delivery latency of routing compared with existing ones.

1. Introduction

Delay Tolerant Network (DTN) is a hot research topic these days (DTN Research Group). It is a type of wireless mobile networks that does not guarantee the existence of a path between a source and a destination at any time. When two nodes move within each other's transmission range during a period of time, they can *contact* or *meet* each other. When they move out of their transmission ranges, the connection is lost. A DTN can be described abstractly using a graph. Each edge in this graph represents a contact. If there is no contact with other host, the message to be delivered needs to be stored in the local buffer of the current host until a connection comes again. Depending on the application, the contact between nodes can be *periodic*, or *predictable* or *nonpredictable*. Therefore, the network must tolerate the delay of the message. Representative DTNs include sensor-based networks that use scheduled intermittent connectivity, terrestrial wireless networks that cannot ordinarily maintain end-to-end connectivity, satellite networks that have moderate delays and periodic connectivity, and underwater acoustic networks that display moderate delays and frequent interruptions due to environmental factors. They also include people carrying mobile devices moving in conferences, university campuses and in social settings.

Due to the uncertainty and time-varying nature of DTNs, routing poses unique challenges compared to conventional wireless networks. In the literature, some approaches are based on deterministic mobility (Ghosh et al., 2005; Jain et al., 2004; Leguay et al., 2005; Lindgren et al., 2004; Liu and Wu, 2007, 2008; Merugu et al., 2004; Tariq et al., 2005; Wu et al., 2007; Zhao et al., 2004) while some others are based on general mobility where nodes mobility cannot be predicted (Chen and Murphy, 2001; Dubois-Ferriere et al., 2003; Vahdat and Becker, 2000). In this paper, we use the general mobility model which reflects the most flexible way of nodes movement: nodes can move dynamically in different directions with different speeds.

If the general mobility model is used, one rudimental approach is *flooding* (Vahdat and Becker, 2000). However, this kind of method results in large number of message copies in the network and thus consumes a high amount of bandwidth and energy which is scarce in DTNs. Therefore some people use single-copy schemes where at any time there is one *holder* or *custodian* of a message. The key point now is how to select the next best router in the neighborhood of the current custodian that has the highest potential to deliver the message to the destination.

The solution to this is the design of utility functions used by several papers in the literature (Chen and Murphy, 2001; Dubois-Ferriere et al., 2003; Juang et al., 2002; Spyropoulos et al., 2004). That is, each node maintains a utility value for every other node in the network, calculated by different criterion such as the last number of times two nodes met, the average of nodes' past meet-

* Corresponding author. Tel.: +1 512 245 3873; fax: +1 512 245 8750.
E-mail addresses: xc10@txstate.edu (X. Chen), js48@txstate.edu (J. Shen), jiewu@temple.edu (J. Wu).

0164-1212/$ - see front matter
doi:10.1016/j.jss.2010.01.029

ing times, and the time elapsed since two nodes last met, etc. These utility values essentially carry indirect information about relative node locations, which get diffused through nodes' mobility. Therefore, algorithms can be designed so that the current custodian can select the next best candidate from the nodes it can reach (including itself) hoping that the candidate can forward the message closer to the destination based on the utility function.

After a closer look at these utility functions in the literature, we think they do not capture enough information about network mobility and there is still room to improve by deciding which information to record and how to deal with it. In our preliminary work in Chen et al. (2009), we extended the information recorded in existing papers to capture more mobility features of the network and used regression functions to process the data. The experimental results using self-written simulator showed that the routing schemes developed based on the extended information model can increase the delivery ratio and reduce the delivery latency of routing compared with existing routing algorithms. In this paper, we explain our methods in detail and further confirm the effectiveness of the extended information model by conducting more experiments using real traces.

In summary, our algorithms use general mobility model and single-copy scheme. Our contributions in DTN routing are: (i) we develop an extended information model that can capture more mobility features in the network and adopt regression functions for data processing so that better routing algorithms can be derived; (ii) experimental results using both our own simulator and real traces show that the routing algorithms based on the extended information model have better delivery ratio and latency than the existing ones.

The rest of the paper is organized as follows: Section 2 introduces the related work, Section 3 describes our motivation, Section 4 puts forward our extended information model, Section 5 presents the routing algorithms based on the extended information model, Section 6 shows the experimental results and Section 7 concludes the paper and points out the future work.

2. Related work

Routing in DTNs poses unique challenges compared to conventional wireless networks due to the uncertainty and the time-varying nature of network connectivity. The addition of time dimension significantly complicates the routing decision. In the literature, some routing approaches are based on deterministic mobility while some others are based on general mobility in which nodes mobility is nonpredictable. The approaches based on deterministic or semi-deterministic mobility include: the centralized routing approaches (Jain et al., 2004; Merugu et al., 2004), the ferry-based routing (Tariq et al., 2005; Zhao et al., 2004; Wu et al., 2007), the probability-based routing without a message delivery guarantee (Ghosh et al., 2005; Leguay et al., 2005; Lindgren et al., 2004), and the scalable routing in DTN where nodes have strict repetitive motions (Liu and Wu, 2007; Liu and Wu, 2008).

If the general mobility model is used, when a source host wants to find a route to a destination, since it does not know where the destination lies, one rudimental approach is to perform a flooding-based route discovery as in Vahdat and Becker (2000) where whenever a host receives a message, it will pass it to all those hosts it can reach directly at that time so that the spread of the message is like the epidemic of a disease. The flooding-based routing uses multiple copies of a single message to find independent path to the destination so as to improve efficiency and robustness. However, it has nonneglectable drawbacks (Spyropoulos et al., 2004): it consumes a high amount of bandwidth and energy; may result in poor performance because of high contention for shared resources. As the average node degree increases, it is not scalable in terms of memory size needed and number of transmissions performed. To overcome these drawbacks, single-copy schemes are put forward.

In single-copy schemes, there is only one custodian for each message. Therefore, the key point in designing efficient single-copy schemes is the selection of the next best custodian, that is, the current custodian tries to find, within the cluster of hosts that it can reach directly at that moment, the host that is the next best candidate to relay the message to the destination. Thus a utility function needs to be defined to compare the potential of each node to reach the destination. Fig. 1 shows the routing process of a single-copy scheme using some utility function. There are four continuous snapshots (a)–(d) of the network in the figure, each showing the custodian and the neighbors of the custodian at a particular time. All other nodes in the network are not drawn for cleaner pictures. Between the snapshots, the nodes are moving dynamically in different directions with different speeds. Suppose some source node wants to deliver a message to the destination 10. After some time the message reaches node 1 as shown by subfigure (a). Node 1 becomes the custodian of the message. It has three neighbors: 2, 3 and 4. Using some utility function, it finds out that node 3 is more likely to meet the destination in the future than itself and others. So node 3 is selected as the next custodian. The message is then delivered to 3. In the next snapshot (b), node 3 selects the next best candidate 6 based on the results of the utility function applied to each of its neighbors and itself. In (c), node 6 selects node 8 using the same method. Then finally in (d), node 8 is fortunate enough to meet the destination 10 and hands it the message.

In the literature, many papers design utility functions by recording contact history of nodes. This is motivated by a simple observation: the history of contact between nodes contains valuable, but noisy information about the current network topology (Dubois-Ferriere et al., 2003). Dubois-Ferriere et al. (2003) record the history of last encounters between nodes. This method predicts the future by just looking at one past data, the number of times two nodes met last time. This utility function is simple but it may not reflect the nature of future mobility. Chen and Murphy (2001) consider not only that but also the frequency of nodes contacting destination D in the past and calculate the average. In this

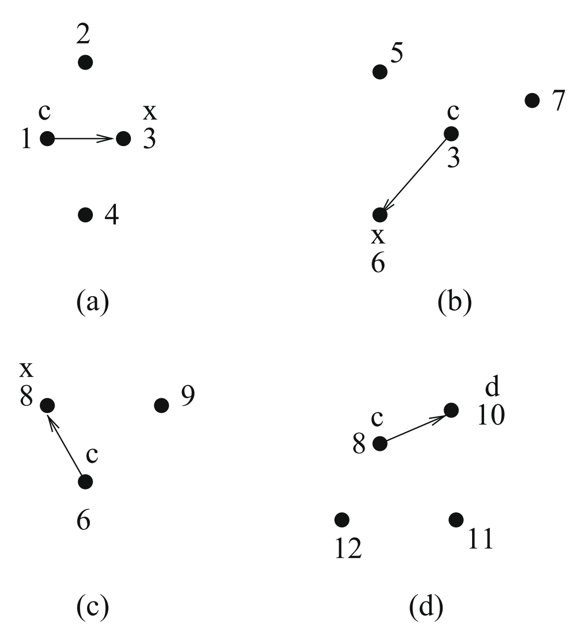


Fig. 1. An example of the routing process.

method more information is included. However, it is also not adequate to reflect mobility as shown in the example in the next section. There are also some variations of these algorithms. For example, Spyropoulos et al. (2004) record the time elapsed since every other node was last encountered because the elapsed time contains the relative location information of the nodes. Juang et al. (2002) assign each node a hierarchy level based on its past success in transferring data to the base station (destination). These two methods are good for applications where the nodes do not move very dynamically. If a network topology changes dynamically, a node that was previously near the destination may no longer be the best communication target. Then, the proposed protocol may mis-direct traffic frequently and get a poor delivery success ratio.

3. Motivation

As stated above, we believe the information the existing papers record is not comprehensive and there is still room to improve by considering extended information in contact history of nodes so that better prediction functions can be designed. The following example motivates our idea.

In Fig. 2, suppose the current time is 6, the custodian meets two hosts *A* and *B* and it wants to choose one of them to relay the message. It will look at the contact history of both hosts with destination *D*. The contact history of host *A* with destination *D* (in Fig. 2a) and *B* with *D* (in Fig. 2b) in the past six units of time are represented by solid dots in the figure. The future contacts of each host with *D* in time units 7 and 8 are shown by not-filled dots. But this information is not currently available to the custodian. Now at time 6, which one to choose? *A* or *B*? That is, the custodian needs to predict at time unit 7, which host, *A* or *B*, will have higher chance to meet *D* in the future using some utility function.

1. If the last meeting times is considered as mentioned in Dubois-Ferriere et al. (2003), during time unit 6, *A* met *D* six times and *B* met *D* seven times, so *B* will be chosen as the candidate to relay the message. But in reality, according to the figure, host *A* has a tendency to contact *D* more and *B*'s tendency becomes quite flat. At time unit 7, *A* will exceed *B* in the number of times to contact *D*. That means, by just looking at the last meeting times is not enough. We need to observe a longer history for the tendency and apply the tendency to predict future number of contacts.
2. If the average meeting times in the last six units of time is calculated as in Chen and Murphy (2001), host *A*'s average is: (2 + 2 + 4 + 4 + 6 + 6)/6 = 4 while *B*'s is: (6 + 6 + 6 + 6 + 7 + 7)/6 = 6.3. So again *B* will be chosen. But *A* is a better candidate because it will meet *D* more and more. If the average method is used, the meetings in the past are treated with equal weights. Actually more recent meetings should be more important than the meetings long time ago.

This example shows both the last meeting times and the average meeting times can mislead the routing sometimes. It motivates us to use extended information in contact history to design better utility functions to steer the routing in the right direction in DTNs.

4. Extended information model

In our extended information model, we require each host to record all the hosts that it has met in the last t ($t \geqslant 1$) units of time. And, unlike those in the literature that take an average of the past meeting times, we weigh the meeting times differently because a host that sees *D* 1 min ago is more likely to see it again in the future than a host that sees *D* 1 day ago. We put different weights on different meeting times. The more recent the meeting time, the larger the weight it will get. The key point to the success of our information model and the later routing protocols is the estimation of the future number of contacts in time unit $t+1$ which is denoted by n_{t+1}. In order to make a good prediction, we put forward the following utility functions.

4.1. Weight and frequency function

In this function, see Fig. 3, we observe t time units before the current time. Let n_i be the number of contacts at time unit *i*. That is, the candidate host met *D* n_t times in time unit t, n_{t-1} times in time unit $t-1, \ldots$, and n_1 times in time unit 1. The formula to predict the future number of contacts is:

$$n_{t+1} = \frac{tn_t + (t-1)n_{t-1} + \cdots + 1 \cdot n_1}{t + (t-1) + (t-2) + \cdots + 1}$$

The host with the highest value of n_{t+1} will be chosen.

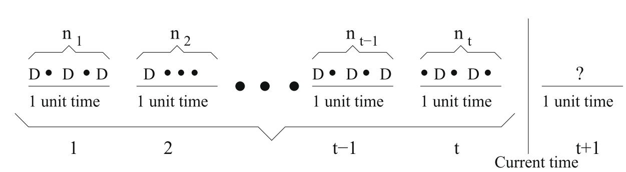


Fig. 3. Illustration of meeting *D* in the *t* units of time.

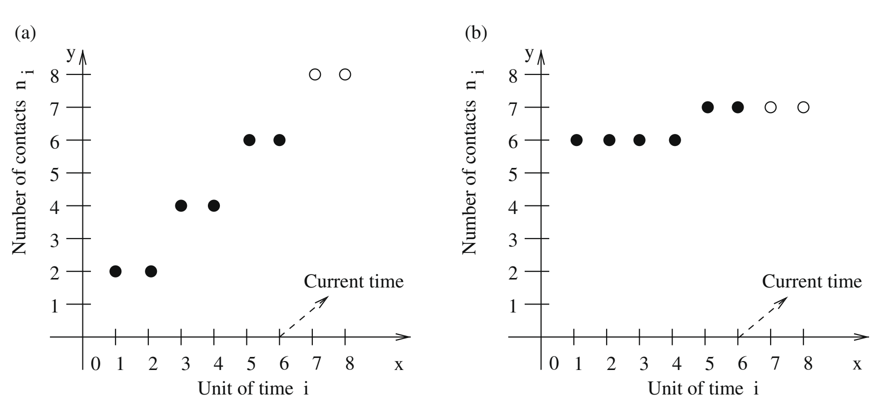


Fig. 2. The contact history of hosts *A* and *B* with destination *D*.

4.2. Regression functions

In this section, we put forward a set of formulas that use the regression models to do the prediction. Let X-axis represent the time unit i and Y-axis represent the number of contacts n_i. We can obtain t points $(i, n_i)(1 \leqslant i \leqslant t)$ in the two dimensional space. Now, the Least-Squares-Method can give a good prediction for n_{t+1}. Depending on the regression models, we can have the following functions.

4.2.1. Linear Regression

Although these t points (i, n_i) may not all lie on a line, we may use a linear model $y = ax + b$ to predict n_{t+1} (see Fig. 4). Since the line $y = ax + b$ may not go through each point (i, n_i), it is reasonable to examine $d_i = n_i - (ai + b) = n_i - ai - b$, which is the difference between the y-coordinates of the point (i, n_i) and the corresponding point on the line $y = ax + b$. The Least-Squares-Method criterion for the "best" linear model approximation is to determine the values of a and b that minimize the sum of squares of all y-differences as follows:

$$F(a,b) = \sum_{i=1}^{t} (n_i - ai - b)^2.$$

To minimize $F(a,b)$, we take the partial derivatives of $F(a,b)$ and set them equal to 0 to find the unique critical point for $F(a,b)$:

$$F_a(a,b) = -\sum_{i=1}^{t} 2i(n_i - ai - b) = 0,$$
$$F_b(a,b) = -\sum_{i=1}^{t} 2(n_i - ai - b) = 0.$$

Then

$$\left(\sum_{i=1}^{t} i^2\right)a + \left(\sum_{i=1}^{t} i\right)b = \sum_{i=1}^{t} in_i,$$
$$\left(\sum_{i=1}^{t} i\right)a + tb = \sum_{i=1}^{t} n_i.$$

Thus

$$\begin{bmatrix} a \\ b \end{bmatrix} = \begin{bmatrix} \sum_{i=1}^{t} i^2 & \sum_{i=1}^{t} i \\ \sum_{i=1}^{t} i & t \end{bmatrix}^{-1} \begin{bmatrix} \sum_{i=1}^{t} in_i \\ \sum_{i=1}^{t} n_i \end{bmatrix}$$

and

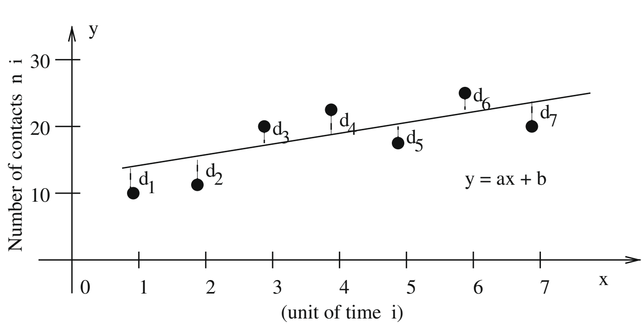


Fig. 4. Linear Regression model.

$$n_{t+1} = a(t+1) + b = [\,t+1 \quad 1\,]\begin{bmatrix} a \\ b \end{bmatrix}$$
$$= [\,t+1 \quad 1\,]\begin{bmatrix} \sum_{i=1}^{t} i^2 & \sum_{i=1}^{t} i \\ \sum_{i=1}^{t} i & t \end{bmatrix}^{-1} \begin{bmatrix} \sum_{i=1}^{t} in_i \\ \sum_{i=1}^{t} n_i \end{bmatrix}$$
$$= \frac{[\,t+1 \quad 1\,]\begin{bmatrix} t & -\sum_{i=1}^{t} i \\ -\sum_{i=1}^{t} i & \sum_{i=1}^{t} i^2 \end{bmatrix}\begin{bmatrix} \sum_{i=1}^{t} in_i \\ \sum_{i=1}^{t} n_i \end{bmatrix}}{t\sum_{i=1}^{t} i^2 - \left(\sum_{i=1}^{t} i\right)^2}.$$

4.2.2. Weighted Linear Regression

It is natural to assume that a recent point (i, n_i) is more closely related to the prediction of n_{t+1} than a less recent point $(i-1, n_{i-1})$. So we add a different weight w_i to each point (i, n_i), where $w_t > w_{t-1} > \cdots > w_1$. (For example, one may use $w_i = i$.) The goal of the Weighted Linear Regression is to minimize the sum of weighted squares of y-differences:

$$WF(a,b) = \sum_{i=1}^{t} w_i(n_i - ai - b)^2.$$

To minimize $WF(a,b)$, we take the partial derivatives of $WF(a,b)$ and find the unique critical point:

$$WF_a(a,b) = -\sum_{i=1}^{t} 2iw_i(n_i - ai - b) = 0,$$
$$WF_b(a,b) = -\sum_{i=1}^{t} 2w_i(n_i - ai - b) = 0.$$

Then

$$\begin{bmatrix} a \\ b \end{bmatrix} = \begin{bmatrix} \sum_{i=1}^{t} i^2 w_i & \sum_{i=1}^{t} w_i \\ \sum_{i=1}^{t} 2iw_i & \sum_{i=1}^{t} w_i \end{bmatrix}^{-1} \begin{bmatrix} \sum_{i=1}^{t} iw_i n_i \\ \sum_{i=1}^{t} w_i n_i \end{bmatrix}$$

and

$$n_{t+1} = \frac{[\,t+1 \quad 1\,]\begin{bmatrix} \sum_{i=1}^{t} w_i & -\sum_{i=1}^{t} iw_i \\ -\sum_{i=1}^{t} iw_i & \sum_{i=1}^{t} i^2 w_i \end{bmatrix}\begin{bmatrix} \sum_{i=1}^{t} iw_i n_i \\ \sum_{i=1}^{t} w_i n_i \end{bmatrix}}{\sum_{i=1}^{t} i^2 w_i \sum_{i=1}^{t} w_i - \left(\sum_{i=1}^{t} iw_i\right)^2}.$$

4.2.3. Quadratic Regression

If we observe that distribution of these points (i, n_i) are not close to any straight line, we may use a quadratic model $y = ax^2 + bx + c$ to predict n_{t+1}. In this quadratic model, the y-difference between each point (i, n_i) and its corresponding point on the graph of $y = ax^2 + bx + c$ is given by $n_i - ai^2 - bi - c$. So the Least-Squares-Method minimizes the following function:

$$Q(a,b,c) = \sum_{i=1}^{t} (n_i - ai^2 - bi - c)^2.$$

Similarly, we can take the partial derivatives of $Q(a,b,c)$ to find the unique critical point for $Q(a,b,c)$. Then we obtain

$$n_{t+1} = \left[\,(t+1)^2 \quad t+1 \quad 1\,\right]\begin{bmatrix} a \\ b \\ c \end{bmatrix},$$

where

$$\begin{bmatrix} a \\ b \\ c \end{bmatrix} = \begin{bmatrix} \sum_{i=1}^{t} i^4 & \sum_{i=1}^{t} i^3 & \sum_{i=1}^{t} i^2 \\ \sum_{i=1}^{t} i^3 & \sum_{i=1}^{t} i^2 & \sum_{i=1}^{t} i \\ \sum_{i=1}^{t} i^2 & \sum_{i=1}^{t} i & t \end{bmatrix}^{-1} \begin{bmatrix} \sum_{i=1}^{t} i^2 n_i \\ \sum_{i=1}^{t} i n_i \\ \sum_{i=1}^{t} n_i \end{bmatrix}.$$

4.2.4. Weighted Quadratic Regression

Similar to the Weighted Linear Regression, if we want to emphasize more on the point (i, n_i) than its previous point $(i-1, n_{i-1})$, we may add a weight w_i to each point (i, n_i) with $w_t > w_{t-1} > \cdots > w_1$. Then the goal of the Weighted Linear Regression is to minimize the following function:

$$WQ(a, b, c) = \sum_{i=1}^{t} w_i (n_i - a i^2 - b i - c)^2.$$

Then $n_{t+1} = \begin{bmatrix} (t+1)^2 & t+1 & 1 \end{bmatrix} M$, where

$$M = \begin{bmatrix} \sum_{i=1}^{t} i^4 w_i & \sum_{i=1}^{t} i^3 w_i & \sum_{i=1}^{t} i^2 w_i \\ \sum_{i=1}^{t} i^3 w_i & \sum_{i=1}^{t} i^2 w_i & \sum_{i=1}^{t} i w_i \\ \sum_{i=1}^{t} i^2 w_i & \sum_{i=1}^{t} i w_i & \sum_{i=1}^{t} w_i \end{bmatrix}^{-1} \begin{bmatrix} \sum_{i=1}^{t} i^2 w_i n_i \\ \sum_{i=1}^{t} i w_i n_i \\ \sum_{i=1}^{t} w_i n_i \end{bmatrix}.$$

4.2.5. Polynomial Regression and Weighted Polynomial Regression

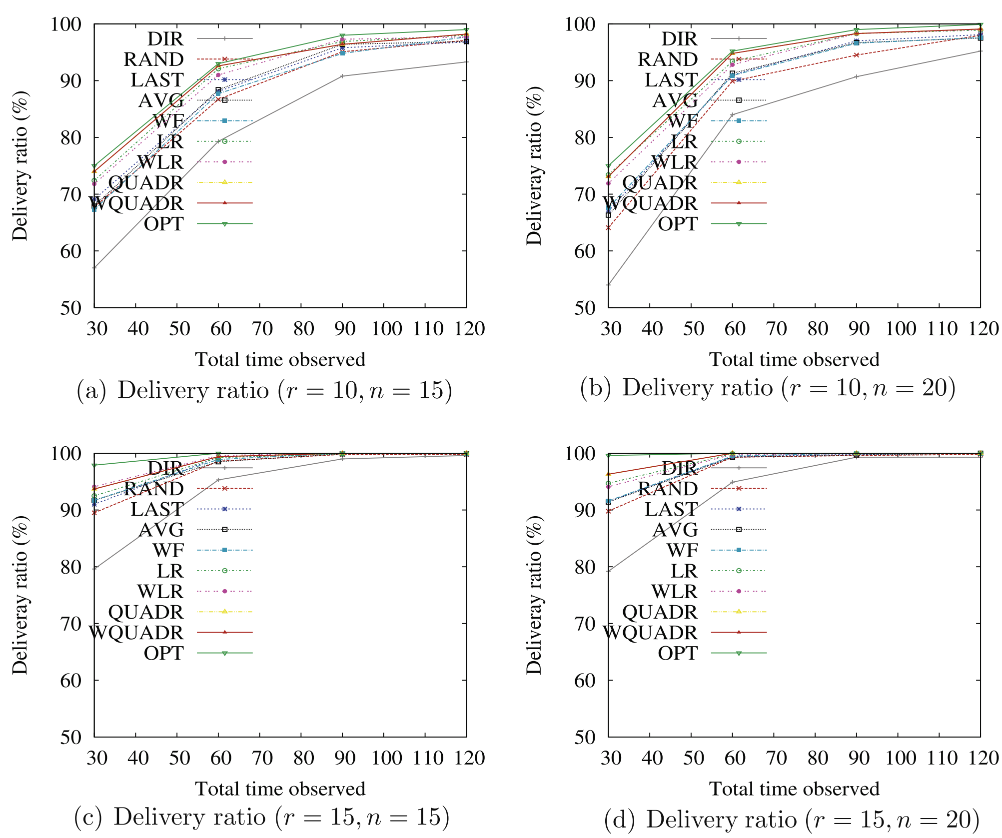
If we observe that the distribution of these points (i, n_i) do not closely follow any linear or quadratic model, we may use a polynomial model of higher order to predict n_{t+1}. Furthermore, if we want to emphasize more on the point (i, n_i) than its previous point $(i-1, n_{i-1})$, we may assign a different weight w_i to each (i, n_t) with $w_t > w_{t-1} > \cdots > w_1$. Results can be obtained similarly. Mathematically speaking, the higher the order of the polynomial, the more accurate the prediction of n_{t+1}. On the other hand, the higher the order of the polynomial, the more computational work is needed for the prediction. So there is a trade-off between choosing the order of the polynomial and the complexity of the calculation.

5. Routing algorithms

In this section, we describe various routing algorithms based on the extended information model. Also, we present some related routing algorithms for later simulation comparisons.

5.1. Extended information-based routing algorithms

Routing algorithms can be derived using the utility functions based on the extended information model to steer the forwarding in the right direction. Specifically, when a host needs to choose a candidate from its cluster of neighbors, an extended information-based utility function is selected from the above and each host in the cluster will use the function to predict the number of contacts n_{t+1} with D in time unit $t+1$ based on its past history. The candidate with the highest number of contacts will be chosen to relay the message. This method is based on the assumption that if a host meets D very often in the past, it is very likely that it will meet D again in the near future. Based on our extended information model, we have the corresponding routing algorithms: *Weight and Frequency* (*WF*), *Linear Regression* (*LR*), *Weighted Linear Regression* (*WLR*), *Quadratic Regression* (*QUADR*) and *Weighted Quadratic Regression* (*WQUADR*) algorithms.



(a) Delivery ratio ($r = 10, n = 15$)

(b) Delivery ratio ($r = 10, n = 20$)

(c) Delivery ratio ($r = 15, n = 15$)

(d) Delivery ratio ($r = 15, n = 20$)

Fig. 5. Delivery ratio with different parameters.

One problem that is associated with these utility-based routing algorithms is that at the beginning it may take a while for the current custodian to find the next best candidate. This is especially true if the current custodian is far from the destination and its neighbors all have poor utility values. A solution to reducing the delivery latency is to initially use random forwarding, which is described below, until the utility value gets higher (Spyropoulos et al., 2004). This hybrid approach initially allows a message to actively explore the network until it finds a good carrier, and then it uses the standard utility routing to efficiently reach the destination. In our simulations, all our schemes have been adapted to include this idea to reduce delivery latency.

5.2. Random algorithm

In the random algorithm, the current custodian selects the next candidate randomly among the neighbors that it can reach.

5.3. Optimal routing algorithm

In the optimal routing algorithm, we assume that we know all the topologies of the network as the nodes move. In that case, an optimal path can be found from a source to a destination. This does not seem practical, but it provides the optimal results for comparison with other algorithms.

5.4. Direct routing algorithm

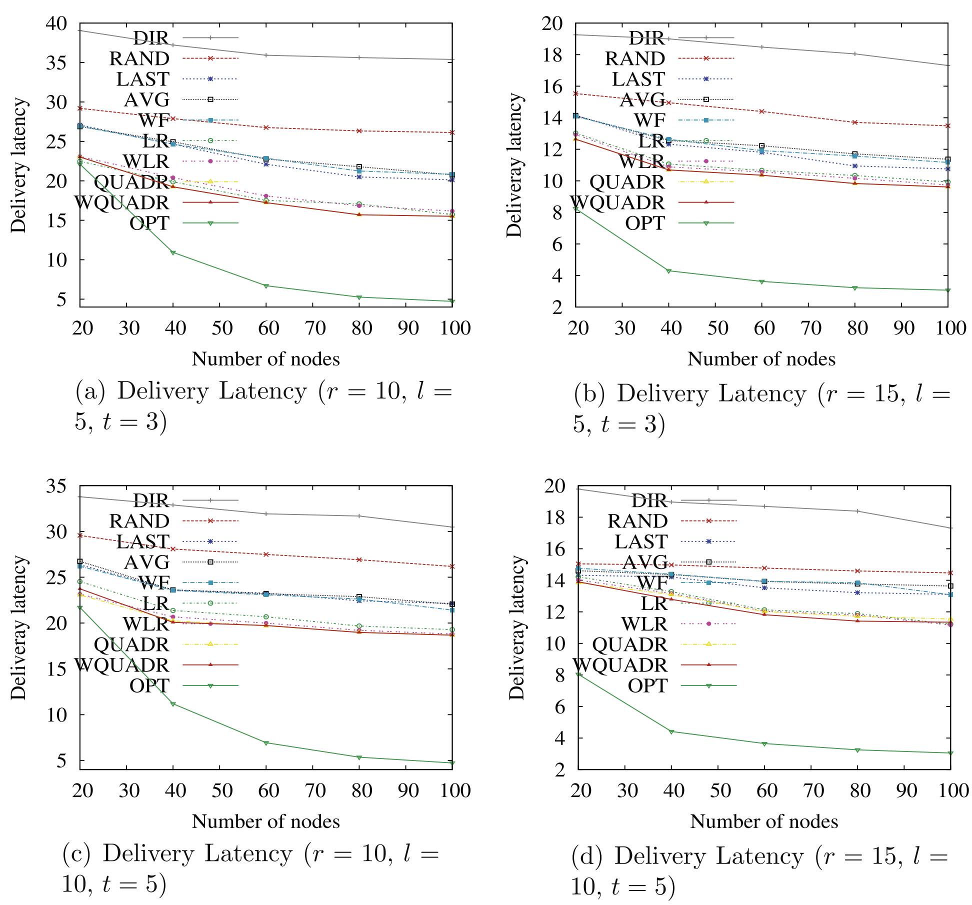
In the direct routing algorithm, a source holds the message until it directly meets the destination and passes it the message. That is, the source does not use any other node as an intermediate router. This algorithm is used as the worst case scenario for comparison with other algorithms.

6. Experimental results

In this section, we show experimental results using a simulator written by ourselves in C language and using real wireless trace data posted on Dartmouth College website (CRAWDAD). We compare our algorithms with existing ones such as Last-time (Dubois-Ferriere et al., 2003) and Average (Chen and Murphy, 2001) Routing Algorithms. For comparison, Random, Optimal and Direct Routing Algorithms are also included. Therefore, here is the list of algorithms that we are going to compare:

1. The Direct Routing Algorithm (DIR).
2. The Random Routing Algorithm (RAND).
3. The Last-time Routing Algorithm (LAST).
4. The Average Routing Algorithm (AVG).
5. The Weighted and Frequency Routing Algorithm (WF).
6. The Linear Regression Routing Algorithm (LR).
7. The Weighted Linear Regression Routing Algorithm (WLR).
8. The Quadratic Routing Algorithm (QUADR).
9. The Weighted Quadratic Routing Algorithm (WQUADR).
10. The Optimal Routing Algorithm (OPT).

In order to compare routing strategies, we define two important metrics to evaluate their performance. One metric is *delivery ratio* and the other is *delivery latency* (Jones and Ward, 2006). The deliv-



(a) Delivery Latency ($r = 10$, $l = 5$, $t = 3$)

(b) Delivery Latency ($r = 15$, $l = 5$, $t = 3$)

(c) Delivery Latency ($r = 10$, $l = 10$, $t = 5$)

(d) Delivery Latency ($r = 15$, $l = 10$, $t = 5$)

Fig. 6. Delivery latency with different parameters.

ery ratio is the fraction of generated messages that are correctly delivered to the final destination within a given time period. This metric shows the ability of a strategy to deliver the message to the destination within a specified period of time. The delivery latency is the time between when a message is generated and when it is received. A short delivery latency can benefit many applications.

6.1. Experiments using our simulator

In the simulator we build, initially all nodes are randomly generated in a 100 m × 100 m area. A source and a destination are randomly picked. Then these nodes move in different directions with different speeds at every time step. We assume in our experiments that if a node met the destination before, there is a tendency that it will meet the destination again in the future. We call the length of the time period we observe the network the *total observation time*.

First we look at the delivery ratio. When the total observation time is short, the transmission range and the number of nodes are small, it is very likely that a message may not reach the destination. We set the number of nodes n to be 15 and 20, transmission range r to be 10 and 15, the length of one time unit l to be 5 time steps and the number of time units t to be 3. The transmission ranges are set small in our experiments because in a sparse graph it is more obvious to see how far away these algorithms are from the OPT algorithm. The total observation time is set as 30, 60, 90, and 100 time steps, respectively. Each sample is run 1000 times. The delivery ratios for each algorithm are recorded and averaged for comparison in Fig. 5a–d.

From the figures, the delivery ratios of OPT and DIR provide the lower- and upper-bounds of all the algorithms. Overall the regression functions are better than RAND, LAST, AVG and WF. Especially when the observation time is short (30 time steps), the difference among them is large. That means, using these regression utility functions can increase the chance to find a path to the destination when time is short. As observation time increases, the difference becomes smaller and smaller. If the observation time is long enough, all algorithms can reach 100% ratio.

Next we compare the delivery latency of the algorithms. The delivery latency is calculated by number of hops in this paper. Different from regular hop counting where one hop is counted when a message is delivered from one node to another, here if the next best candidate is still the current node and thus is equivalent to the custodian delivering the message to itself at that moment, hop count keeps accumulating. Since delivery latency is the parameter that we want to look at, we make the delivery ratio for each algorithm 100%. In our experiments, setting the observation time to 200 time steps is enough to achieve that. The length of one time unit l is set at 5 and 10 time steps and the number of time units t is set at 3 and 5. The transmission range r is set at 10 and 15, respectively. The number of nodes n tried is 20, 40, 60, 80, and 100. Each sample of parameters is run 1000 times. The results are presented in Fig. 6a–d.

In the figures, DIR and OPT algorithms provide the upper- and lower-bounds of delivery latency. If a custodian keeps holding the message until it meets the destination itself as in the DIR algorithm, the delivery latency is the longest. From the results we can see that it is better for the custodian to forward the message to some other node, even randomly. The three algorithms: Last, AVG and WF provide similar results. The performance of Last is not bad in the three considering the simplicity of information it records. The four regression algorithms are better than the previous three. Thus the complexity of the algorithms does have a reward here. However, the differences among the four regression algo-

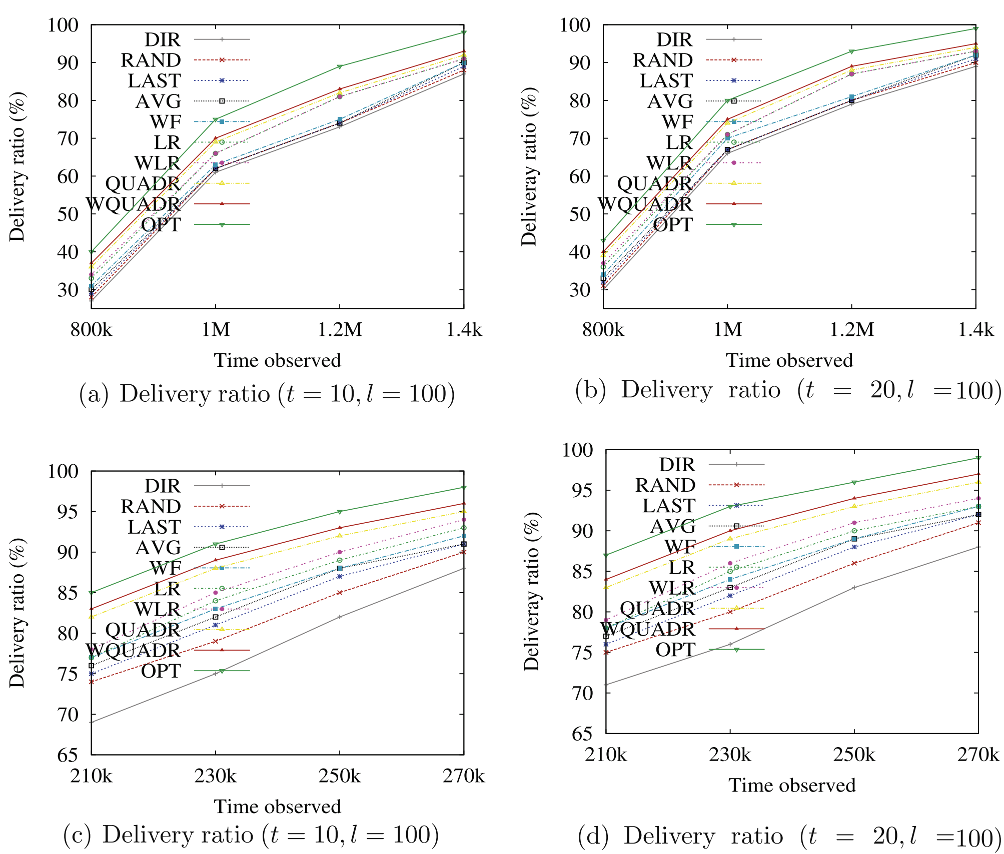


Fig. 7. Delivery ratio with different parameters using Content (a) and (b) and Info traces (c) and (d).

rithms are not obvious. Unless you know that the mobility of nodes follows a certain pattern, these regression algorithms are already good enough. That also explains why we do not further explore higher order regression functions such as Polynomial Regression and Weighted Polynomial Regression here. The OPT algorithm presents the shortest latency because if the changes of the topology are all known, the shortest path from the source to the destination can be found. Also we can see that the gaps in delivery latency between the three algorithms and the four regression ones are larger in Fig. 6a and c than those in Fig. 6b and d. That is because in more sparse graphs where the routing is more difficult, the regression algorithms are more likely to deliver the message to the destination with a lower latency.

6.2. Experiments using real traces

In this section, we compare the above 10 algorithms using real traces posted on CRAWDAD website (CRAWDAD). This is a Dartmouth college website posting wireless network data gathered from real human mobility patterns for the research community. The data resources we use here consist of contact traces between short-range Blue-tooth enabled devices (iMotes Chaintreau et al., 2007) carried by individuals in conference environment, namely Content 2006 and Infocom 2006. In short, we call them Content trace and Info trace.

We still use delivery ratio and delivery latency to compare the 10 algorithms in the last section. The delivery ratios of different algorithms using Content trace are shown in Fig. 7a and b while those of the Info trace are presented in Fig. 7c and d. Based on data ranges in the traces, in both traces, we set $l = 100$ and t to be 10 and 20, respectively. In the Content trace, the total observation time is set from 800K to 1.4M s. Here, $K = 10^3$ and $M = 10^6$. In the Info trace, the total observation time is set from 210K to 270K s. In both traces, the results of the delivery ratio are similar to those in our simulator above. The DIR and OPT algorithms provide the lowest and the highest delivery ratios. Here, the higher order regression algorithms QUADR and WQUADR are better than lower order regression algorithms LR and WLR, and the lower order regression algorithms are better than simple utility algorithms WF, AVG, LAST and RANDOM.

Now we look at delivery latency. For both traces, if we vary the observation time and draw the delivery latency for different algorithms as in the figures of delivery ratio, the results cannot be clearly displayed due to the feature of the dataset. In that case, to show our results better, we change the meaning of the horizontal axis from “Time observed” to “Algorithm”. Now the ticks on the horizontal axis represent the labels of the 10 algorithms. For example, tick 1 represents Direct Routing Algorithm, tick 2 represents Random Routing Algorithm, and so on. We set the observation time to be 1M and 1.4M s for the Content trace, whereas for the Info trace, we set the observation time to be 210K and 270K s. In both traces, $t = 10$ and $l = 100$. In each setting, the delivery latency is calculated and averaged over all the successful cases. The results are shown in Fig. 8a and b for the Content trace and Fig. 8c and d for the Info trace. As we can see, the OPT and the DIR algorithms give the shortest and longest latencies. From the OPT algorithm, the latency increases in the order of WQUADR, QUADR, WLR, LR, WF, AVG, LAST, RAND and DIR. The results also match those in our simulator in the last section.

In summary, from the experimental results using real traces and our own simulator, we can draw the conclusion that gathering more information in the contact history of nodes and using appro-

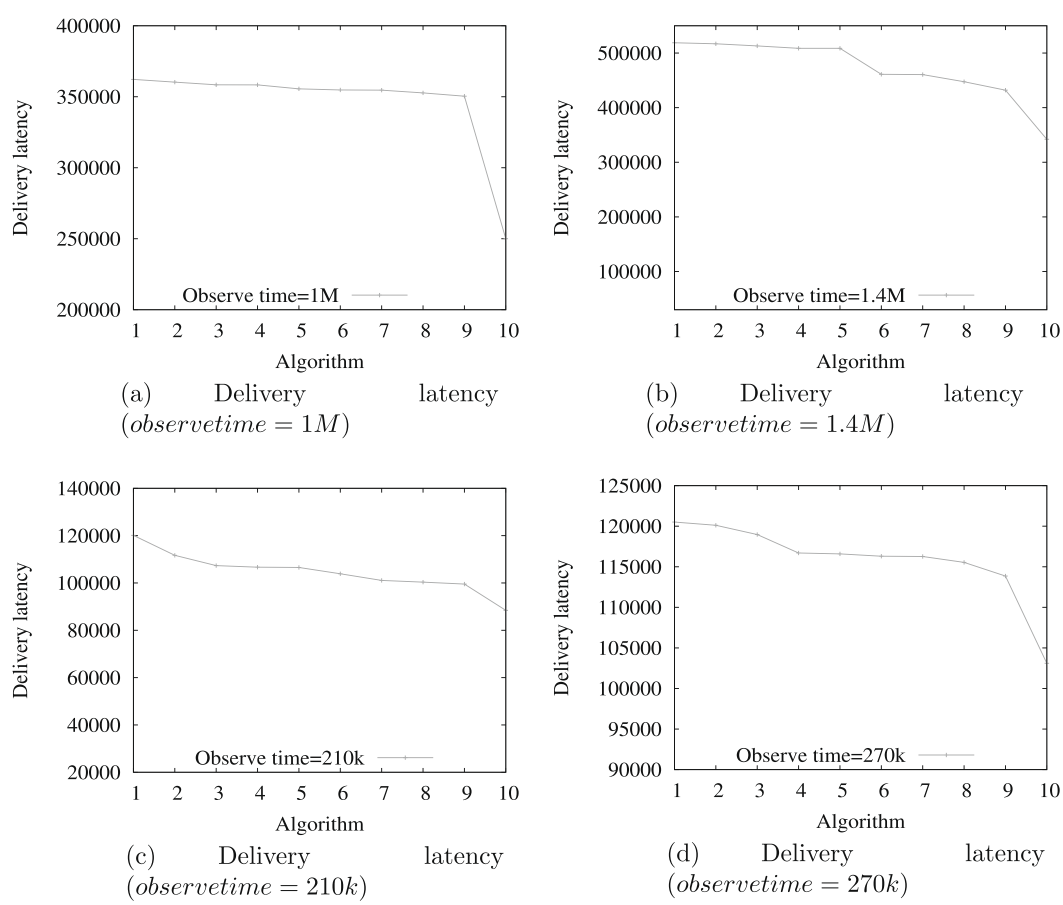


Fig. 8. Delivery latency with different parameters using Content (a) and (b) and Info traces (c) and (d).

priate regression functions to process them can improve routing efficiency in DTNs.

7. Conclusions and future work

In this paper, we developed efficient routing algorithms for DTNs using an extended information model. In a DTN where nodes move dynamically in different directions with different speeds, we believe recording and processing the right information in the right way will help a message to be delivered to the destination faster. Experimental results from our own simulator and real wireless trace data showed that our algorithms can give better performance than those using simpler information and simpler processing methods.

Our extended information model can be further improved. For example, it can incorporate more important parameters to the system such as energy and the future plans of hosts. As we know, one of the scarcest resources of DTNs is energy. In many applications, once the hosts are deployed, it is difficult to recharge them. In the routing algorithms we propose, if a host meets the destination very often, it will be used to relay the message so frequently that its battery will be depleted very soon. In order to balance the energy in all the hosts, the information model should be extended to include the energy parameter. Also, if more information can be known about the hosts, for example, their future meeting schedules with each other, the efficiency of the routing can be further improved. The routing algorithm can be dynamically switched from one to another when the future plan changes to best suit the mobility of nodes. The concerns will be how to incorporate the future plans into the information model and when to switch the routing algorithm for the best results.

In addition, so far we have looked at the single-copy schemes. In the next step, we are going to study multi-copy schemes using our extended information model. How many copies are needed and how the copies are distributed in each hop to find the path to the destination efficiently will be the key points. That will be our future work.

Acknowledgments

This research was supported in part by NSF Grants CNS 0835834, CNS 0531410 and CNS 0626240.

References

Chaintreau, A., Hui, P., Crowcroft, J., Diot, C., Gass, R., Scott, J., 2007. Impact of human mobility on opportunistic forwarding algorithms. IEEE Transaction on Mobile Computing 6 (6), 606–620.

Chen, X.C., Murphy, A.L., 2001. Enabling disconnected transitive communication in mobile ad hoc networks. In: Proceedings of the Workshop on Principles of Mobile Computing (POMC), August, 2001, pp. 21–27.

Chen, X., Shen, J., Wu, J., 2009. A novel information model for efficient routing protocols in delay tolerant networks. In: Proceedings of 8th International Workshop PMEO UCNS'2009 in Conjunction with IEEE IPDPS'2009.

CRAWDAD: A Community Resource for Achieving Wireless Data at Dartmouth. <http://crawdad.cs.dartmouth.edu>.

DTN Research Group. <http://w5.www.dtnrg.org/>.

Dubois-Ferriere, H., Grossglauser, M., Vetterli, M., 2003. Age matters: efficient route discovery in mobile ad hoc networks using encounter ages. In: Proceedings of ACM MobiHoc.

Ghosh, J., Philip, S.J., Qiao, C., 2005. Sociological orbit aware location approximation and routing (SOLAR) in MANET. In: Proceedings of ACM MobiHoc.

Jain, S., Fall, K., Patra, R., 2004. Routing in a delay tolerant network. In: Proceedings of ACM SIGCOMM.

Jones, E.P.C., Ward, P.A.S., 2006. Routing strategies for delay-tolerant networks. Computer Communication Review.

Juang, P., Oki, H., Wang, Y., Martonosi, M., Peh, L.S., Rubenstein, D., 2002. Energy-efficient computing for wildlife tracking: design tradeoffs and early experiences with zebranet. In: Proceedings of ASPLOS'02, pp. 96–107.

Leguay, J., Friedman, T., Conan, V., 2005. DTN routing in a mobility pattern space. In: Proceedings of ACM SIGCOMM Workshop on Delay-Tolerant Networking.

Lindgren, A., Doria, A., Schelen, O., 2004. Probabilistic routing in intermittently connected networks. Lecture Notes in Computer Science 3126, 239–254.

Liu, C., Wu, J., 2007. Scalable routing in delay tolerant networks. In: Proceedings of MobiHoc.

Liu, C., Wu, J., 2008. Routing in a cyclic mobispace. In: Proceedings of the 9th ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc).

Merugu, S., Ammar, M., Zegura, E., 2004. Routing in Space and Time in Network with Predictable Mobility, Technical Report: GIT-CC-04-07, College of Computing, Georgia Tech.

Spyropoulos, T., Psounis, K., Raghavendra, C.S., 2004. Single-copy routing in intermittently connected mobile networks. In: Proceedings of IEEE Secon'04.

Tariq, M.M.B., Ammar, M., Zegura, E., 2005. Message ferry route design for sparse ad hoc networks with mobile nodes. In: Proceedings of ACM MobiHoc.

Vahdat, A., Becker, D., 2000. Epidemic Routing for Partially-connected Ad Hoc Networks, Technical Report: CS-2000-06, Duke University.

Wu, J., Yang, S., Dai, F., 2007. Logarithmic store-carry-forward routing in mobile ad hoc networks. IEEE Transactions on Parallel and Distributed Systems 18 (6).

Zhao, W., Ammar, M., Zegura, E., 2004. A message ferrying approach for data delivery in sparse mobile ad hoc networks. In: Proceedings of ACM MobiHoc, 2004.

Xiao Chen is an associate professor of Computer Science at Texas State University - San Marcos. She received the Ph.D. degree in Computer Engineering from Florida Atlantic University in 1999, the Master degree and Bachelor degree in Computer Science from Shanghai University of Science and Technology (now Shanghai University) in 1995 and 1992. Her research interests are in the areas of delay tolerant networks, sensor networks and ad hoc wireless networks.

Jian Shen received the B.S. and M.S. degrees from University of Science and Technology of China and the Ph.D. degree from Queen's University, Canada, all in mathematics, in 1991, 1994, and 1998, respectively. He held a postdoctoral appointment at University of Wisconsin - Madison during 1998–1999, and was a visiting assistant professor at University of Illinois at Urbana-Champaign in 2000. In 2001, he joined the Department of Mathematics at Texas State University-San Marcos, where he is currently a Professor of Mathematics. His research interests include discrete mathematics, combinatorial matrix theory, probabilistic methods and theoretical computer science.

Jie Wu is a distinguished professor and chair in the Department of Computer and Information Sciences at Temple University, and was a program director at US National Science Foundation. He has published over 450 papers in various journals and conference proceedings. His research interests are in the areas of wireless networks and mobile computing, routing protocols, fault-tolerant computing, and interconnection networks. He served as the chair of IEEE TCDP committee. He also served as an editor, program chair, and co-chair of numerous international committees, conferences, and journals. He is the author of the text Distributed System Design published by the CRC press. He is an IEEE fellow.