

A Novel Information Model for Efficient Routing Protocols in Delay Tolerant Networks

Xiao Chen
Dept. of Computer Science
Texas State University
San Marcos, TX 78666
xc10@txstate.edu

Jian Shen
Dept. of Mathematics
Texas State University
San Marcos, TX 78666
js48@txstate.edu

Jie Wu
Dept. of Computer Science and Engineering
Florida Atlantic University
Boca Raton, FL 33431
jie@cse.fau.edu

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 about the network and their information processing is not adequate. In this paper, we develop an information model that can capture more contact information and uses regression functions for data processing. Simulation results show that our routing algorithms based on our information model can increase the delivery ratio of the messages and reduce the delivery latency of routing compared with existing ones.

I. INTRODUCTION

Delay Tolerant Network (DTN) is a hot research topic these days [2]. 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. The terms “contact” and “meet” will be used interchangeably in this paper. When they move away from each other out of the transmission range, 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 the next host, the message to be delivered needs to be stored in the local buffer of the current host until the connection comes again. Depending on the application, the contact between nodes may be there *periodically*, or can be *predictable*, or *nonpredictable*. Therefore, the network must tolerant 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.

Due to the uncertainty and time-varying nature of DTNs, routing poses unique challenges compared to conventional wireless networks. In the literature, some routing approaches are based on deterministic mobility [4], [5], [8], [9], [10], [11], [12], [13], [17], [14] where some others are based on general mobility in which nodes mobility cannot be predicted [1], [3], [16]. 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* and its derivatives [1], [3], [16]. 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 are 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 which are used by several papers in the literature [1], [3], [7] and [15]. That means, each node maintains a utility value for every other node in the network, calculated by different criteria such as the last number of times two nodes met, the average of nodes’ past meeting 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 believe 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 them. In this paper, we will develop an information model that can capture more comprehensive information about the network and use regression functions to process them. As our simulations show later in the paper, the routing schemes developed based on the information model can increase the delivery ratio of the

messages and reduce the delivery latency of routing compared with existing routing algorithms.

In summary, our algorithms use general mobility model and single-copy scheme. The contributions of this paper are: (i) we develop an information model that can capture more comprehensive information and adopts regression methods for data processing so that better routing algorithms can be derived; (ii) simulation results show that the routing algorithms based on our information model have better delivery ratio and delivery latency than the existing ones.

The rest of the paper is organized as follows: section II introduces the related work, section III puts forward our information model, section IV presents the routing algorithms based on our information model, section V shows the experimental results and section VI concludes the paper and points out the future work.

II. 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 cannot be predicted. The approaches based on deterministic mobility or semi-deterministic mobility include: the centralized routing approaches [5], [12], the ferry-based routing [13], [17], [14], the probability-based routing without a message delivery guarantee [4], [8], [9], and the scalable routing in DTN where nodes have strict repetitive motions [10], [11]. The approaches based on general mobility models include the flooding-based approaches such as [1], [3], [16]. In this paper, we will use the general mobility model which reflects the most flexible way of nodes movement where nodes can move dynamically in different directions with different speeds.

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 [16] 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 and its derivatives use multiple copies of a single message to find independent path to the destination so as to improve efficiency and robustness. However, they have non-neglectable drawbacks [15]: they consume 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, they are not scalable in terms of memory size needed and number of transmissions performed. Therefore, it is highly desirable to design efficient single-copy schemes for DTNs.

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 custodian,

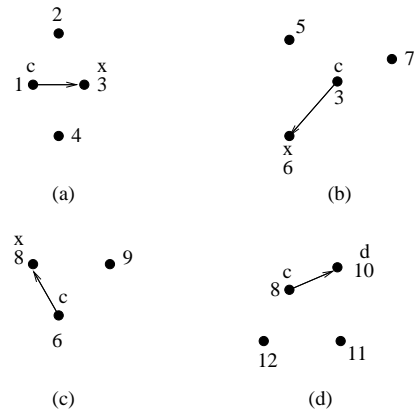


Fig. 1. An example of the routing process

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 carry the message closer to the destination. Thus a metric calculated by a utility function must be defined to compare the potential of each node to reach the destination. Figure 1 shows the process of routing. 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 subgraph (a). Node 1 becomes the custodian of the message. It has three neighbors: 2, 3 and 4. Using some utility function, it is calculated that node 3 is more likely to meet the destination in the future than others. So it 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 on each of its neighbors and itself. Then 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 literature, many papers design utility functions by recording contact history of nodes. This is motivated by a simple observation [3]: the history of contact between nodes contains valuable, but noisy information about the current network topology. Dubois-Ferriere, Grossglauser and Vetterli [3] 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 very simple but it may not reflect the nature of future mobility. Chen and Murphy [1] consider not only the last encounters but also the frequency of the nodes contacting destination D in the past and calculates the average. In this 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,

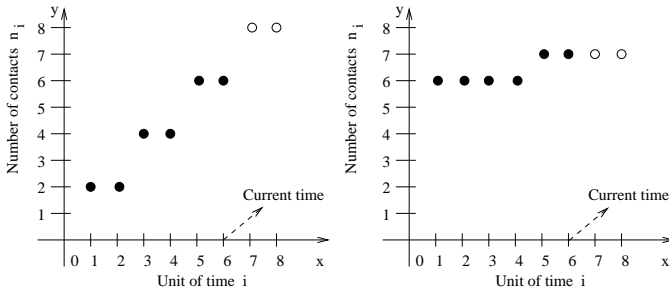


Fig. 2. The contact history of hosts A and B with destination D

Psounis and Raghavendra record the time elapsed since every other node was last encountered [15] because the elapsed time contains the relative location information of the nodes. And Juang, Oki and Wang assign each node a hierarchy level based on its past success in transferring data to the base station (destination) [7]. These two methods are good for applications where the nodes do not move very dynamically. If a network topology changes very 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 rate.

Therefore, a better information model that can capture more accurate mobility feature should be developed to guide the routing in DTNs.

III. INFORMATION MODEL

As stated above, we believe the information the existing papers record is not comprehensive and there is still room to improve by deciding which data to record and how to deal with them. The following example explains why.

See Figure 2. Suppose the current time is 6, the history of contacts of host A with destination D and B with D in the past six units of time are represented by the solid dots in the figures. In order to show the tendency, the future number of contacts of each host with D in time units 7 and 8 are also shown by not-filled dots. Now at time 6, a host needs to choose A or B to relay a message. That is, at time unit 7, which host, A or B , will have higher chance to meet D ?

- 1) If the last meeting times is considered as mentioned in the literature [3], 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 host A has a tendency to contact D more and more and B 's tendency becomes quite flat. At time unit 7, it is highly likely that 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 [1], 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. If the average method is used, the meetings in the past

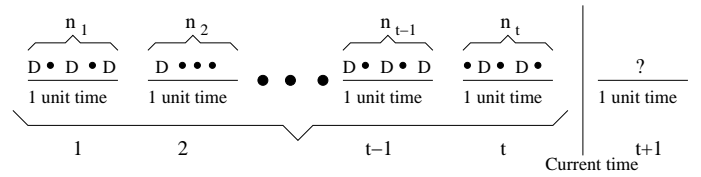


Fig. 3. Illustration of meeting D in the t units of time

are treated with equal weights. Actually more recent meetings should be more important than the meetings long time ago.

This example explains both the last meeting times and the average meeting times can mislead the routing sometimes. Therefore, an information model that can capture more accurate information should be developed to steer the routing in the right direction in DTNs.

In our information model, we require each host record all the hosts that it has met in the last t ($t \geq 1$) 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 one minute ago is more likely to see it again in the future than a host that sees D one day ago. We put different weights on different meeting times. The more recent the meeting times, 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 calculation methods (utility functions).

A. Weight and Frequency Method

In this method, see Figure 3, we observe t units time period 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 $t - 1$, \dots , and n_1 times in time unit 1. The formula to predict the future number of contacts is:

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

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

B. Regression Model Methods

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 . Then we can obtain t points (i, n_i) ($1 \leq i \leq t$) 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 use, we can have the following formulas.

a. 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 Figure 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

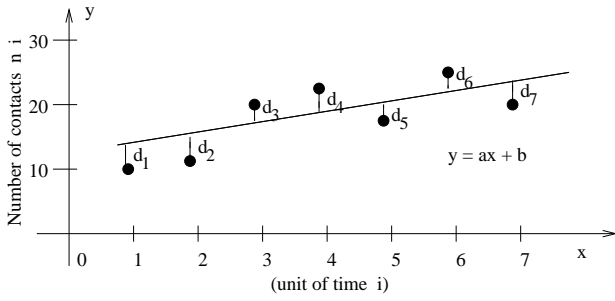


Fig. 4. Linear regression model

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 following function, which is the sum of squares of all y -differences:

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

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 $F(a, b)$:

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

Then

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

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

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

b. 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 $w_t > w_{t-1} > \dots > w_1$. (For example, one may use $w_i = i$.) The goal of the Weighted Linear Regression is to minimize the following sum of weighted squares of y -difference:

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

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

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

Then

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

and $n_{t+1} =$

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

c. 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 $Q(a, b, c)$ and find the unique critical point for $Q(a, b, c)$. Then we obtain

$$n_{t+1} = \begin{bmatrix} (t+1)^2 & t+1 & 1 \end{bmatrix} \begin{bmatrix} a \\ b \\ c \end{bmatrix},$$

where

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

d. 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 $w_t > w_{t-1} > \dots > w_1$. Then the goal of the Weighted Linear Regression is to minimize the following function:

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

Then $n_{t+1} = [(t+1)^2 \ t+1 \ 1]M$, where

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

e. 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_i) with $w_t > w_{t-1} > \dots > w_1$. Results can be obtained similarly. Mathematically speaking, the higher the order of the polynomial, the more accuracy on 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 for the approximation model and the complexity of the calculation.

IV. ROUTING ALGORITHMS

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

A. Information Model-based Routing Algorithms

Routing algorithms can be derived using the proposed information model to steer the forwarding in the right direction. Specifically, when a host needs to choose a candidate from its cluster of neighbors, each host in the cluster uses one of the methods above as the utility 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 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.

One problem that associates with these utility-based routing algorithms is that at the beginning it may take a while for the current custodian to find a better next 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 reduce the delivery latency is to initially use random forwarding, which is described below, until the utility value gets higher [15]. 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 the delivery latency.

B. Random Algorithm

In the random algorithm, the current custodian selects the next candidate randomly among the neighbors that it can reach instead of using any utility function.

Besides the Random Algorithm, in order to compare the performance of these algorithms, we set a lower-bound and an upper-bound delivery latency algorithms. The *optimal routing* algorithm is used as the lower-bound latency algorithm and the *direct routing* algorithm is used as the upper-bound latency algorithm.

C. 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.

D. Direct Routing Algorithm

In the direction 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.

V. EXPERIMENTAL RESULTS

In this section, simulations are conducted to compare the algorithms based on the information model with some existing ones such as Last-time [3] and Average [1] 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 must define some important metrics to evaluate their performance. In this paper one metric is *delivery ratio* and the other is *delivery latency* [6]. The delivery 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

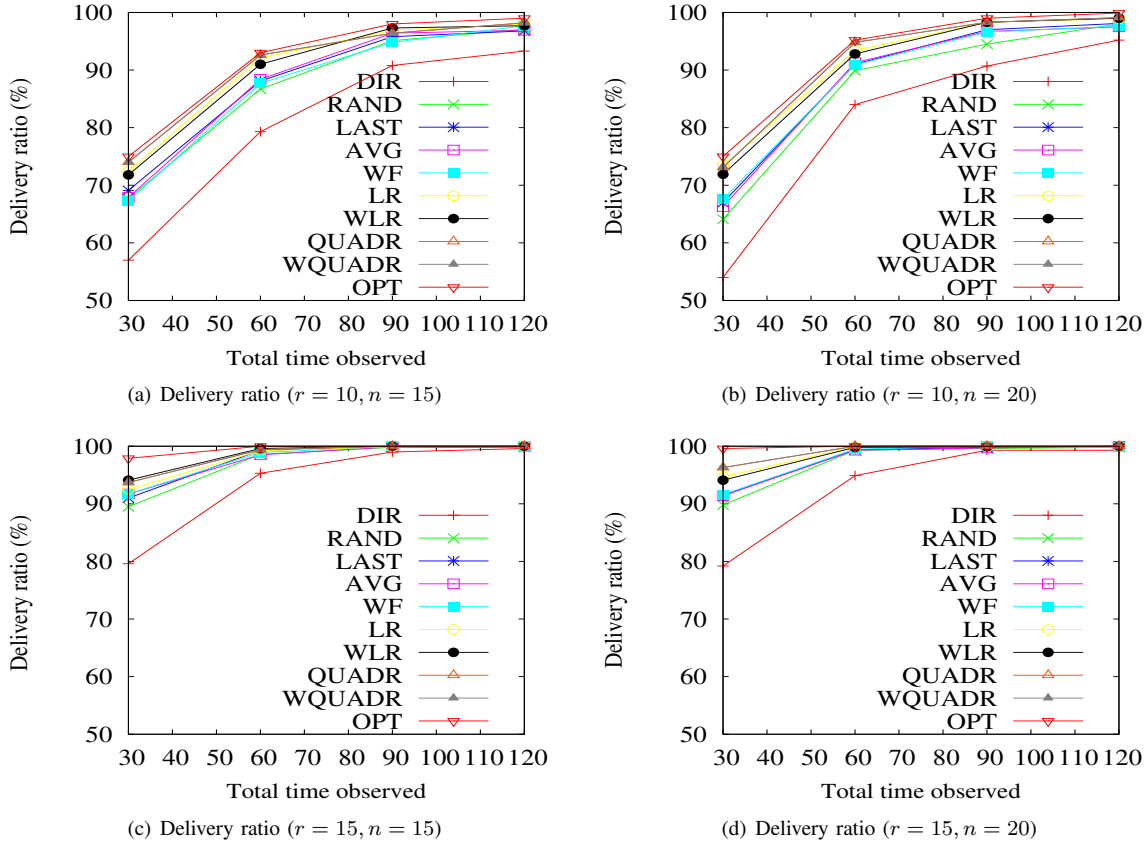


Fig. 5. Delivery ratio with different parameters

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.

Initially all nodes are randomly generated in a $100m \times 100m$ 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.

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 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 of parameters is run 1000 times. The delivery ratios for each algorithm are recorded and averaged for comparison in Figures 5(a), 5(b), 5(c) and 5(d).

From the figures, we can see that the delivery ratios of OPT and DIR provide the lower- and upper-bound of all the algorithms. Overall the regression utility 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 in terms of 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 Figures 6(a), 6(b), 6(c) and 6(d). Since these figures use the same legend as the figures above them, the legend is omitted in each figure for better presentation.

In both figures, DIR and OPT algorithms provide the upper- and lower-bounds of delivery latency. If a custodian keeps

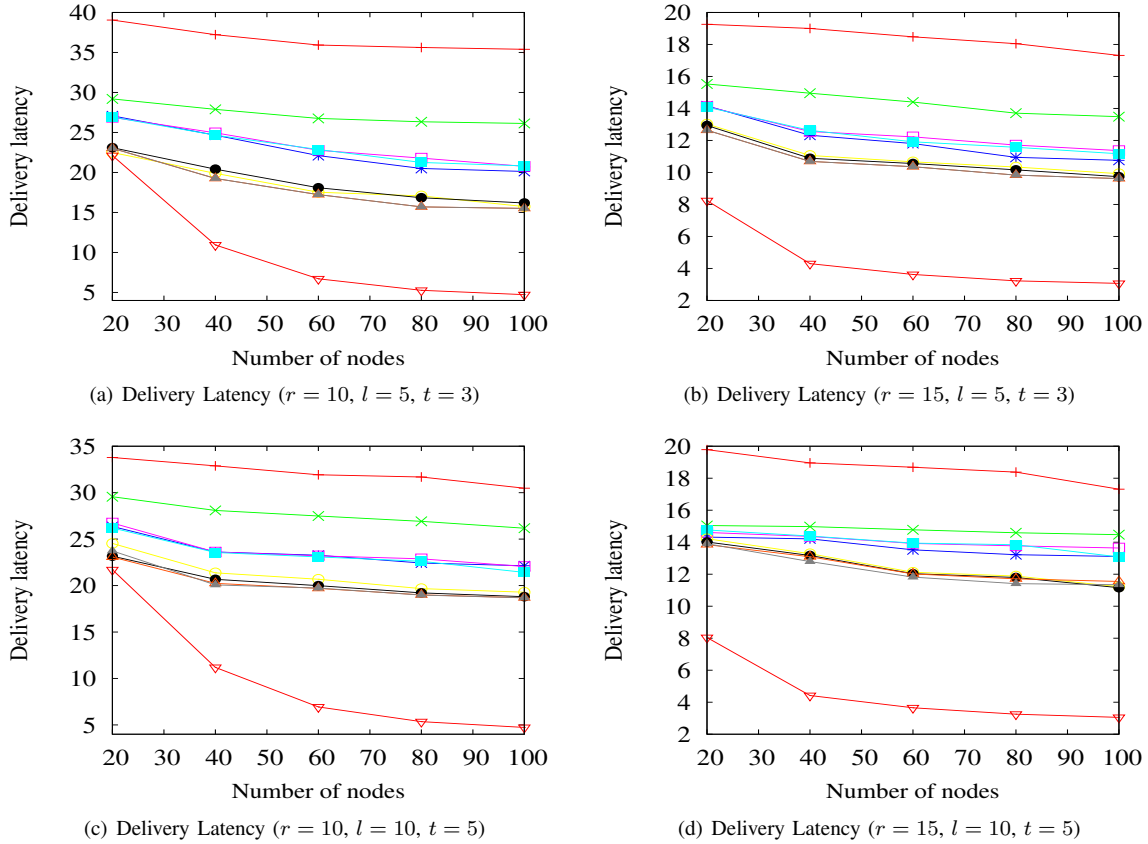


Fig. 6. Delivery latency with different parameters

holding the message until it meets the destination itself as in the DIR algorithm, the delivery latency is the longest. It will be better for the custodian to forward the message to some 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. The complexity of the algorithms does have a reward here. However, the differences among the four regression algorithms 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, it is not difficult to find the shortest path from the source to the destination. Also we can see that the gaps in delivery latency between the three algorithms and the four regression ones are larger in Figures 6(a) and 6(c) than those in Figures 6(b) and 6(d). That means in more sparse graphs where the routing is more difficult, the regression algorithms are more likely to deliver the message to the destination.

VI. CONCLUSIONS AND FUTURE WORK

In this paper, we have developed efficient routing algorithms for DTNs using our 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. Simulations have shown that our algorithms can give better performance than those using simpler information and simpler processing.

Our information model can be further improved. For example, it can be extended by incorporating 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 the 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 in 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 information model. How many copies are needed and how are the copies 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 grant CNS 0835834.

REFERENCES

- [1] X. C. Chen and A. L. Murphy, "Enabling Disconnected Transitive Communication in Mobile Ad Hoc Networks", *Proceedings of the Workshop on Principles of Mobile Computing (POMC)*, August, 2001, pp. 21-27.
- [2] DTN Research Group. <http://w5.www.dtnrg.org/>.
- [3] H. Dubois-Ferriere, M. Grossglauser, and M. Vetterli, "Age matters: efficient route discovery in mobile ad hoc networks using encounter ages," *Proceedings of ACM MobiHoc*, 2003.
- [4] J. Ghosh, S. J. Philip, and C. Qiao, "Sociological orbit aware location approximation and routing (SOLAR) in MANET," *Proceedings of ACM MobiHoc*, 2005.
- [5] S. Jain, K. Fall, and R. Patra, "Routing in a delay tolerant network," *Proceedings of ACM SIGCOMM*, 2004.
- [6] Evan P. C. Jones, Paul A. S. Ward, "Routing Strategies for Delay-Tolerant Networks".
- [7] P. Juang, H. Oki, Y. Wang, M. Martonosi, L. S. Peh, and D. Rubenstein, "Energy-efficient computing for wildlife tracking: design tradeoffs and early experiences with zebranet," *Proceedings of ASPLOS'02*, 2002, pp. 96-107.
- [8] J. Leguay, T. Friedman, and V. Conan, "DTN routing in a mobility pattern space," *Proceedings of ACM SIGCOMM Workshop on Delay-Tolerant Networking*, 2005.
- [9] A. Lindgren, A. Doria, and O. Schelen, "Probabilistic routing in intermittently connected networks," *Lecture Notes in Computer Science*, 3126:239-254, August 2004.
- [10] C. Liu, and J. Wu, "Scalable Routing in Delay Tolerant Networks," *Proceedings of MobiHoc*, 2007.
- [11] C. Liu and J. Wu, "Routing in a Cyclic MobiSpace," accepted to appear in *Proceedings of the 9th ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc)*, 2008.
- [12] S. Merugu, M. Ammar, and E. Zegura, "Routing in space and time in network with predictable mobility," *Technical report: GIT-CC-04-07*, College of Computing, Georgia Tech, 2004.
- [13] M. M. B. Tariq, M. Ammar, and E. Zegura, "Message ferry route design for sparse ad hoc networks with mobile nodes," *Proceedings of ACM MobiHoc*, 2005.
- [14] J. Wu, S. Yang, and F. Dai, "Logarithmic store-carry-forward routing in mobile ad hoc networks," *IEEE Transactions on Parallel and Distributed Systems*, 18(6), June 2007.
- [15] T. Spyropoulos, K. Psounis, and C. S. Raghavendra, "Single-copy routing in intermittently connected mobile networks," *Proceedings of IEEE Secon'04*, 2004.
- [16] A. Vahdat and D. Becker, "Epidemic Routing for Partially-Connected Ad Hoc Networks," *Technical report, CS-2000-06*, Duke University, July 2000.
- [17] W. Zhao, M. Ammar, and E. Zegura, "A message ferrying approach for data delivery in sparse mobile ad hoc networks," *Proceedings of ACM MobiHoc*, 2004.