Predict and Relay: An Efficient Routing in Disruption-Tolerant Networks *

Quan Yuan, Ionut Cardei and Jie Wu
Department of Computer Science and Engineering
Florida Atlantic University
Boca Raton, FL 33431
{qyuan@, icardei@cse., jie@cse.}fau.edu

ABSTRACT

Routing is one of the most challenging open problems in disruption-tolerant networks (DTNs) because of the short-lived wireless connectivity environment. To deal with this issue, researchers have investigated routing based on the prediction of future contacts, taking advantage of nodes' mobility history. However, most of the previous work focused on the prediction of whether two nodes would have a contact, without considering the time of the contact. This paper proposes predict and relay (PER), an efficient routing algorithm for DTNs, where nodes determine the probability distribution of future contact times and choose a proper next hop in order to improve the end-to-end delivery probability. The algorithm is based on two observations: one is that nodes usually move around a set of well-visited landmark points instead of moving randomly; the other is that node mobility behavior is semi-deterministic and could be predicted once there is sufficient mobility history information. Specifically, our approach employs a time-homogeneous semi-markov process model that describes node mobility as transitions between landmarks. Landmark transition and sojourn time probability distributions are determined from nodes' mobility history. A simulation study shows that this approach improves the delivery ratio and also reduces the delivery latency compared to traditional DTN routing schemes.

Categories and Subject Descriptors

C.2.2 [**Computer Systems Organization**]: COMPUTER-COMMUNICATION NETWORKS—*Store and forward networks*

General Terms

Algorithms, Measurement, Performance

*This work was supported in part by NSF grants CNS 0847664.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
MobiHoc'09, May 18–21, 2009, New Orleans, Louisiana, USA.
Copyright 2009 ACM 978-1-60558-531-4/09/05 ...$5.00.

Keywords

Disruption-Tolerant Networks (DTNs), Landmarks, Time-related Markov model, Prediction, Routing.

1. INTRODUCTION

Wireless ad hoc networks have been traditionally modeled as connected graphs with stable end-to-end paths. However, for emerging wireless applications, such as sensor networks for wildlife tracking and MANETs operating in challenging environments [1, 2], wireless links are short-lived and end-to-end connectivity turns out to be sporadic. Such phenomena are prevalent in disruption tolerant networks (DTNs) [3, 4, 5, 6], where the connection between nodes in the network changes over time, and the communication suffers from frequent disruptions, making the network partially connected. The intermittent end-to-end paths and the changing topology make conventional MANET routing protocols fail, as they are designed with the assumption that the network stays connected. Routing in DTNs is an especially challenging problem because of the temporal scheduling element in a dynamic topology, which is not present in traditional MANETs. Nodes have to decide who the next hop is, but also when to forward, as they route packets to destinations in a store-and-carry way.

Researchers have proposed a number of broad methods to solve the above issue. In general, previous related works fall into three categories: *mobile resource-based*, *opportunity-based*, and *prediction-based*. In the first category [7, 8], systems employ mobile resources (data mules and mobile agents) as message ferries. These can be directed to pick up, move towards the next hop, and deliver messages to implement end to end store-and-carry message delivery. In opportunity-based schemes [9, 10], nodes forward messages during contacts that are unscheduled or random. For the prediction-based schemes [11, 12] inter-node contacts and mobility behavior are predicted, generally using prior contact history. The next hop and the contact in which a message is forwarded are selected using the predictions such that a quality of service (QoS) metric (e.g. delay or delivery ratio) is maximized. Most of the existing prediction-based routing protocols focus on the prediction of whether two nodes would have a contact in the future without considering *when* the contact happens. We believe that lack of contact timing information undermines the contact prediction accuracy, and consequently reduces routing performance.

In this paper, we propose predict and relay (PER), a routing method for DTNs that relies on predicting future contacts. We use a model based on a time-homogeneous semi-markov process model to predict the probability distribution of the time of contact and the probability that two nodes will have a contact in the future.

Our study is inspired by two observations from reality pointed out, in [13]. One is that nodes in a network within a social environment do not move completely randomly. Instead, they usually move around a set of well-visited locations that we call landmarks [14] in this paper. Specifically, nodes show preference for a small number of landmarks and would move less often to the neighborhood of other landmarks [15]. While near a landmark that is visited by other users, a communication device may use the opportunity to establish contacts with other nodes and exchange DTN messages. The second observation is that in some social environments the node trajectory in time is almost deterministic [16]. This means a node has its own mobility schedule and it generally moves between landmarks according to that schedule, subject to few random deviations. For example, a student on a campus moves between classrooms, the dormitory, cafetaria, and the gym. The dwell time at each landmark and the landmark trajectory are fairly regular, with small variations. Nodes keep one schedule for a relatively long interval (e.g. a semester), so it can be assumed they operate in steady state with a few deviations.

The objective of our work is to explore the solutions to the routing problem in DTNs with a semi-markov model. The main contributions of this article are: a) a landmark trajectory prediction method that uses a time-homogeneous semi-markov process to determine the probability distribution of node arrival time at landmarks, b) a method to determine a probabilistic *contact profile* that predicts internode contacts, and c) a set of message forwarding rules that improve the message delivery ratio by controlling the selection of the contact in which a message is transmitted to the next hop. Simulation results show that our approach raises the delivery ratio using the improved contact prediction accuracy, compared with other traditional routing protocols. Furthermore, results show that PER algorithms also reduce the delivery latency in DTNs.

The remainder of this paper is organized as follows. In Section 2, we discuss the existing routing approaches in DTNs. Section 3 describes the overview of the predict and relay schemes. Section 4 presents the system model and detailed routing schemes in our protocol. Section 5 provides simulation results and we conclude our work in Section 6.

2. RELATED WORK

In the past, several routing schemes have been proposed to improve the routing performance in DTNs. This section reviews the related work in the literature and highlights the differences among them. Due to the limited space, we focus on results that inspired our work or that are widely cited.

As mentioned before, there are three categories for current routing schemes [17, 18, 19, 20, 21] in DTNs: *moving resources-based*, *opportunity-based* and *prediction-based*. In the first category, systems usually employ extra moving resources, such as data mules and moving agents, as ferries for message delivery. Researchers in [7] present an architecture to collect data in sparse sensor networks, which uses data MULEs to pick up data from the sensors when in close range, buffer it, and drop off the data to wired access points. Similarly, in [8], buoys monitor the water quality on a lake and onboard sensors relay measurements using nodes on tourist tour-boats and pleasure cruisers. Both of these approaches improve routing performance with additional mobile nodes, although controlling these resources leads to extra cost and overhead.

The opportunity-based schemes utilize neither the mobile resources nor the prediction methods for routing. Instead, messages are exchanged only when two nodes meet at the same place by chance. For example, Vahdat and Becker [10] use the epidemic routing scheme by flooding. Further, the ZebraNet [22] project applies such an approach to research on animal migrations and inter-species interactions. Data is flooded in the network and eventually reaches access points. Spray and Wait [23] protocol is a multi-copy routing protocol that controls the flooding overhead by limiting the number of message copies distributed in the Spraying phase and then relies on *direct delivery* when a message is transmitted to the final destination, then waits until the destination meets one of them. Harras et al. [24] have improved and evaluated the controlled message flooding schemes with heuristics, for instance, on hop limits or timeouts. Approaches falling into this category usually distribute multiple copies in the network, to ensure a high reliability of delivery and a low latency. But they also bring in a high price of the buffer occupancy and bandwidth consumption.

In the prediction-based schemes, nodes' mobility is estimated based on a history of observations. A typical example is utility-routing [25, 26], where each node maintains a utility value for every other node that is updated using the time between contacts. A node forwards a message copy only to nodes with a higher utility for the message destination. The utility value is considered as the predictor of two nodes' future likelihood of contact. In [27], Burns et al. propose a routing protocol that uses past frequencies of contacts, as well as the past contacts. LeBrun et al. [28] propose a routing algorithm for vehicular DTNs that uses the current position and trajectories of nodes to predict their future distance to the destination. In [11], researchers present MaxProp, a protocol that mainly relies on the prediction of the path likelihoods according to historical contact data. Protocol performance evaluations are conducted on 60 days' trace data from a real DTN network deployed on 30 buses. MobySpace [29] is another prediction-based generic algorithm for DTN routing, that uses a high-dimensional Euclidean space constructed upon nodes' mobility patterns. The frequency of visits of nodes to each possible location is recorded as the basis of the future distance calculation in the Euclidean space. Most of these protocols focus on whether two nodes will have a contact without sufficiently considering *when* the contact happens. Our approach, however, employs a time-homogeneous semi-markov process model to predict both the contacts and their time. We predict when two specified nodes have a contact based on their history information. Since time is considered, our future contact prediction is more accurate than the traditional ones.

3. PREDICT AND RELAY

We consider a DTN with a finite number of mobile nodes with unique IDs that move mostly between a set of landmarks. A landmark is defined as a place where nodes can communicate directly, i.e. any two nodes that are located at a landmark at the same time can establish a contact to exchange messages. Nodes at different landmarks cannot establish a contact. Landmarks are also assigned unique IDs.

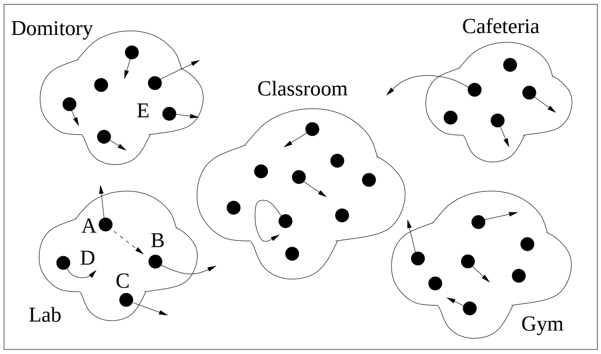


Figure 1: An example of a landmark-based mobility model. The solid line denotes the mobility behavior of nodes. The dashed line stands for forwarding packets.

As described in the introduction, the networks of social nature have nodes follow a semi-deterministic trajectory, with small deviations from a repetitive sequence of landmarks with constant dwell times. For illustration in this article, we use a campus network where the landmarks are independent WLANs installed in classrooms and buildings, while nodes are the students and faculty with PDAs or laptops. We acknowledge that in the real world the WLANs are actually connected by a backbone. Another good example of this type of DTN is a network in a rural area where the landmarks consist of WLANs located in villages, interconnected by buses that carry messages on portable computers. A campus DTN is shown in Figure 1. In both examples, nodes follow a scheduled route that is subject to random deviations.

In prediction-based routing schemes, history information is used to predict nodes' future mobility, which becomes the basis of the decision to forward messages towards the destination. Most of the previous prediction-based DTN routing methods predict whether two nodes would encounter, but consider *when* two nodes will meet insufficiently. We argue that figuring out when two nodes will meet with a probability distribution could improve the delivery ratio, as well as reduce the delivery latency.

PER is a single-copy DTN routing protocol – only a single instance of a message is forwarded towards the destination. Each message carries in its header a time-to-live (TTL) field. After the TTL, expires the message should be dropped. Messages are forwarded hop-by-hop in a succession of contacts using a greedy approach. At each forwarding step, PER selects the next hop that has the highest probability of delivery to the destination.

When a node u has to forward a message from its queue (e.g. at the beginning of a contact, or when a message is received from an application) it computes a probability metric $f(x)$ for all nodes currently in contact with u (the set N_u), and for itself, $x \in \{u\} \cup N_u$. This metric indicates the delivery performance to the destination if node u selects node x as the next hop and forwards the message to x.

The current node then selects the next hop h as the node for which the delivery probability metric is maximized:

$$h = \arg\max_{x \in \{u\} \cup N_u} f(x). \quad (1)$$

If the selected next hop is the current node ($h = u$), then the message will not be forwarded.

The formulation of the probability metric $f(x)$ is detailed in section 4.3 and is based on predicting node mobility over a finite time horizon. In contrast with prior research, that mainly focuses on estimating the probability of contact regardless of the contact time, our method uses landmark trajectory prediction to determine the probabilities of contact for each time unit. The prediction of nodes' future mobility relies on their trajectory history, that is recorded and disseminated throughout the network in an epidemic fashion. We believe this approach is feasible in a network that has reached steady state.

Figure 1 gives an example of the PER process. Node A needs to send a message to node E. Located at the same landmark with node A are nodes B, C, D. Based on the history mobility information, A predicts that before the TTL expires node B has a better delivery performance to node E than all other nodes in the Lab. Therefore, node A will forward the message to node B. In this scenario, node B will leave the Lab landmark later and will meet E in the $Classroom$ for delivery.

4. SYSTEM MODEL

4.1 Assumptions

In this article we focus on the effectiveness of a time-based mobility prediction for DTN routing. We make, therefore, some simplifying assumptions that will be addressed as part of our future work. We assume that during a contact, nodes can successfully transfer all messages that need to be forwarded. This requires a reliable transport and high bandwidth or long enough contacts. These conditions can be more guaranteed when two nodes tend to dwell at the same landmark for at least minutes in a DTN supported by 802.11a/g/n WLANs, like a campus environment.

Moreover, each landmark has a unique landmark id in the network, and nodes are aware of which landmark they are located at at anytime. Also, we assume that the whole network is composed of the neighborhoods of landmarks, which means a node is always associated to a certain landmark in the network. Nodes do not spend any time on the transition between landmarks.

4.2 TH-SMP Model

We model the mobility of a node m with a time homogeneous semi-markov (TH-SMP), (X_n^m, T_n^m) with discrete time. The states are represented by the landmarks $L = 1, ..., l$. A node that moves between two landmarks transitions in the markov process between the corresponding states. We assume the transition probabilities between states have the markov memoryless property, meaning that the probability of a node m to transition from state X_i^m to state X_{i+1}^m is independent of state X_{i-1}^m. Thus, process (X_n^m) is a standard discrete-time markov Chain. The random variable T_n^m represents the time instant of the transition $X_n^m \rightarrow X_{n+1}^m$. Random variable $T_{n+1}^m - T_n^m$ describes the landmark sojourn time, or state holding time. Note that the sojourn time does not include the time when the nodes are in transit between landmarks. These random variables are i.i.d., with distributions that do not change over time (time-homogeneous) and can be different from the geometric or the exponential distributions (semi-markov).

The associated time-homogeneous semi-markov kernel Q is defined by:

$$\begin{aligned} Q_{ij}^m(t) &= P(X_{n+1}^m = j, T_{n+1}^m - T_n^m \le t | X_0^m, ..., X_n^m; \\ & \quad T_0^m, ..., T_n^m) \\ &= P(X_{n+1}^m = j, T_{n+1}^m - T_n^m \le t | X_n^m = i) \end{aligned}$$

Suppose $P^m = |p_{ij}^m|$ is the transition probability matrix of the (X_n^m) embedded markov chain for node m. Then the transition probability from state i to state j is

$$p_{ij}^m = \lim_{t \to \infty} Q_{ij}^m(t) \qquad i, j \in L.$$

Also, we derive the probability $S_i^m(t)$ that node m will leave the neighborhood of landmark i on or before time unit t:

$$S_i^m(t) = P(T_{n+1}^m - T_n^m \le t | X_n^m = i) = \sum_{j=1}^{l} Q_{ij}^m(t). \tag{2}$$

Note that $S_i^m(t)$ also indicates the distribution of the dwell time at landmark i for node m, regardless of the next landmark.

Let $Z^m = (Z_t^m, t \in \mathbb{N}^*)$ be another TH-SMP that describes the state (landmark) occupied by node m at time t. The transition probabilities for process Z are defined by $\phi_{ij}^m(t) = P(Z_t^m = j | Z_0^m = i)$. In the following, we drop the ^{m} superscript to simplify the notation. If we know that a node is currently in state i, after t time units, it will be in state j with probability $\phi_{ij}(t)$. ϕ provides the prediction of the node's location at a landmark at an arbitrary time $t > 0$ knowing its current location. The derivation of ϕ is described next.

For a fixed current state i, $\phi_{ij}(t)$ forms the probability mass function of the random variable that indicates the state at time t. Thus, $\sum_{j=1}^{l} \phi_{ij}(t) = 1$ for any initial state i and future time $t > 0$. For the border case $t = 0$, $\phi_{ij}(0) = \delta_{ij}$, where δ is the kronecker symbol.

To determine $\phi_{ij}(t)$ we start with a special case when the process stays in state i between time 0 and t, with no transitions.

$$\begin{aligned} P(X_t = i | X_0 = i, T_1 \ge t) &= P(T_1 - T_0 \ge t \mid X_0 = i) \\ &= 1 - S_i(t). \end{aligned}$$

If the node transitions at least once between times 0 and t, we consider on the time k of the first transition from i, and on the state r to which the process moves immediately after state i. We obtain:

$$\begin{aligned} & P(X_t = j \mid X_0 = i \text{ and at least one transition}) \\ & \qquad = \sum_{r=1}^{l} \sum_{k=1}^{t-1} \dot{Q}_{ir}(k) \phi_{rj}(t-k), \end{aligned}$$

where $\dot{Q}_{ir}(k) = \frac{dQ_{ir}(k)}{dk} = Q_{ir}(k) - Q_{ir}(k-1)$ is the time derivative of Q, assuming a time step equal to the unit.

Putting it together, we obtain:

$$\phi_{ij}(t) = (1 - S_i(t))\delta_{ij} + \sum_{r=1}^{l} \sum_{k=1}^{t-1} \dot{Q}_{ir}(k) \phi_{rj}(t-k) \tag{3}$$

We first note that ϕ can be calculated iteratively, as $\phi_{ij}(t)$ depends on probabilities $\phi_{lj}(t-k)$ computed in the previous steps.

Specifically, since we consider the time discrete in our model, Eq. 3 is rewritten as follows,

$$\begin{aligned} \phi_{ij}(k) &= P(Z_k = j | Z_0 = i) \\ &= d_{ij}(k) + \sum_{r=1}^{l} \sum_{\tau=1}^{k} v_{ir}(\tau) \phi_{rj}(k-\tau), \end{aligned} \tag{4}$$

where $d_{ij}(k) = (1 - S_i(k))\delta_{ij}$, $v_{ij}(k) = \dot{Q}_{ij}(k)$, and $k \in \mathbb{N}$. Furthermore, $v_{ij}(k)$ can be approximated by the following equation,

$$v_{ij}(k) = \begin{cases} Q_{ij}(1) & \text{for } k = 1 \\ Q_{ij}(k) - Q_{ij}(k-1) & \text{for } k > 1. \end{cases}$$

Using the assumption that the landmark dwell time random variables are independent from the embedded state transition process (X_{ij}), we derive:

$$\begin{aligned} Q_{ij}(k) &= P(X_{n+1} = j, T_{n+1} - T_n \le k | X_n = i) \\ &= P(X_{n+1} = j | X_n = i) \cdot \\ & \quad \cdot P(T_{n+1} - T_n \le k | X_{n+1} = j, X_n = i) \\ &= p_{ij} S_{ij}(k) \end{aligned}$$

$S_{ij}^m(k)$ is the probability that node m will move from landmark i to landmark j at, or before time k. The time parameter k can be used to represent a relative time offset. Based on the Markov property of the underlying processes, if the state i of a node is known at a time k_0, than at time $k > k_0$, the probability of that node being in state j is $\phi_{ij}(k - k_0)$.

With sojourn time probability distributions S_{ij}^m and the transition probability matrix P^m, we can predict the future landmark location of node m based on its current location using probability distributions $\phi_{ij}^m(k)$. Section 4.4 describes how to derive these probabilities.

4.3 Contact Probabilities

In this section we propose additional metrics to be used for studying various probabilistic delivery probability metrics $f(x)$ during the packet forwarding defined in section 3.

Distributions $\phi_{ij}^m(k)$ give the probability that the future location at time k of a node m will be j considering that at time 0 the location was landmark i. Assuming that trajectories of nodes are independent of each other and that the most recent known state of node a is s_a (at time k_a), and for node b is s_b (at time k_b, with $k_a < k, k_b < k$), the probability of contact between a and b at a landmark i at time k is

$$C_{ab}^i(k) = \phi_{s_a i}^a(k - k_a) \cdot \phi_{s_b i}^b(k - k_b) \text{ for } k > 0$$

Then, the probability that a and b are in contact at a time k at any landmark is

$$C_{ab}(k) = \sum_{i \in L} C_{ab}^i(k) \text{ for } k > 0. \tag{5}$$

We note that $C_{ab}(k)$ does not define a proper probability mass function, as $0 \le \sum_k C_{ab}(k) \lesseqgtr 1$.

For our study of probabilistic delivery probability metrics $f(x)$, we define the probability that two nodes begin their first contact at time k. Note that when we talk about that nodes a and b begin their first contact at time k, it means that they had no contacts in any prior time units in a considered interval. Assuming that node trajectories are independent, the probability of the first contact at time k is defined as,

$$R_{ab}(k) = C_{ab}(k) \prod_{t=0}^{k-1} (1 - C_{ab}(t)) \quad \text{for} \quad k > 0. \tag{6}$$

Denote the maximum message acceptable delivery delay by D, which means packets are required to reach the destination in time D. Moreover, Let n_c be the chosen neighbor for evaluating the delivery probability metrics, and d is the destination. The prediction metric functions are defined as follows:

Function 1: This prediction metric function is defined in terms of the maximum probability of contact in time [1, D], which is defined as,

$$f_1 = \max_k C_{n_c d}(k), 1 \leq k \leq D \tag{7}$$

Function 2: We define Function 2 based on the maximum average probability of contact in time [1, D], which is,

$$f_2 = \sum_{k=1}^{D} C_{n_c d}(k) \tag{8}$$

Function 3: For this prediction metric function, we mainly focus on the first contact probability. Thus, maximum probability of the first contact before the deadline is the basis of Function 3, which is,

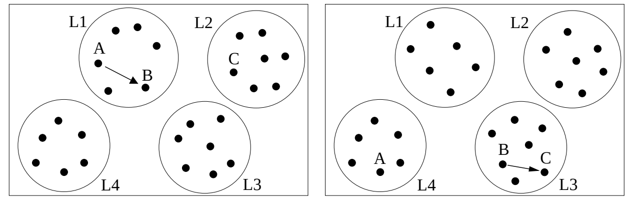
$$f_3 = \sum_{k=1}^{D} R_{n_c d}(k) \tag{9}$$

Relay node selection is done based on the above prediction functions. For each message that is taken from the queue during a contact, the prediction metric is computed using only one of the three prediction metric functions for each neighbor (another node that is in contact with the current node). In a greedy approach, the node with the highest metric value is picked as the relay node to forward the packet. Intuitively, the chosen neighbor should have the largest contact probability with the destination in the future D time steps. We refer to the PER algorithms using Function 1, 2, and 3 as PER1, PER2, and PER3, respectively. Once the relay nod is determined, the message is forwarded to it. If the selected node is the current node itself, then the message will be kept in the queue for a later transmission.The corresponding algorithm is shown in Algorithm 1.

Algorithm 1 Predict and Relay Algorithm

[1] Node exchanges and updates the history mobility information of other nodes with its neighbors.
[2] Node uses one of the three prediction functions (PER1, PER2, and PER3) to picks a neighbor as the next hop.
[3] Node forwards the packet to the chosen node in [2].

Figure 2 is an example to illustrate the PER algorithm, where node A sends a packet to node C at time t. Note that at time t, node A is at landmark $L1$ and node C is at landmark $L2$. With the history mobility information, A first uses one of the three prediction metric functions, for example f_1, to find the neighbor node B, who is most likely to meet the destination node C in the future. Then A relays



(a) Network Topology at Time t (b) Network Topology at Time $t + \Delta t$

Figure 2: An example to illustrate the process of PER.

the packet to B. After Δt, B carries the packet and moves to landmark $L3$, where it meets node C. Finally, C gets the packet from B.

4.4 Deriving Mobility Parameters

4.4.1 Two Parameters

To determine the prediction functions f, PER needs to compute two parameters, the transition probability matrix P^m and the sojourn time probability distribution matrix, $S_{ij}^m(k)$ for each node m. In this section we describe a method to determine these two parameters using node mobility history.

P^m is the transition probability matrix of the embedded Markov chain for node m. Figure 3 shows an example transition probability matrix for node m that visits four landmarks: Classroom, Gym, Dormitory, and Lab.

$$P^m = \begin{pmatrix} p_{11}^m & p_{12}^m & p_{13}^m & p_{14}^m \\ p_{21}^m & p_{22}^m & p_{23}^m & p_{24}^m \\ p_{31}^m & p_{32}^m & p_{33}^m & p_{34}^m \\ p_{41}^m & p_{42}^m & p_{43}^m & p_{44}^m \end{pmatrix}$$

At any on those landmarks, the node could pick to stay for a while or move to another landmark according to its preferred probability. For example, if the node is at the gym, it then may: 1) move to the Lab with probability p_{43}^m; 2) or stay in the gym with the probability p_{44}^m; 3) or go to the dormitory with probability p_{42}^m; 4) or head for the Classroom with probability p_{41}^m. Those mobility probabilities constitute the transition probability matrix P^m. Note that each node has its own transition probability matrix that reflects its trajectory.

We now define the P^m probabilities as follows.

Definition 1: The probability p_{ij}^m that node m moves from landmark i to landmark j is defined as the observed transition frequency:

$$p_{ij}^m = num_{ij}^m / num_i^m,$$

where num_i^m stands for the number of transitions from landmark i without considering the next landmark, and num_{ij}^m is the number of transitions from landmark i to landmark j. Obviously, $num_{ij}^m \leq num_i^m$ and $p_{ij}^m \leq 1$. By keeping track of num_i^m and num_{ij}^m, each node could generate and refine its own P matrix over time.

We calculate the sojourn time probability distribution $S_{ij}^m(k)$, $k \in \mathbb{N}$ as follows.

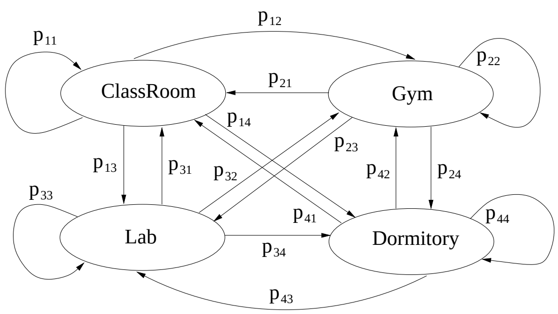


Figure 3: The markov model of the node m's mobility.

Definition 2: The sojourn time probability distribution at landmark i when followed by a transition to landmark j, $S_{ij}^m(k)$, is defined as:

$$S_{ij}^m(k) = P(t_{ij}^m < k),$$

where t_{ij}^m is the sojourn time or state holding time at landmark i when j is the next visited landmark. We assume that when the network reaches steady state, the mobility history provides a representative sample from which the sojourn time distribution can be drawn. Therefore the probabilities $P(t_{ij}^m < k)$ are computed with the equation:

$$P(t_{ij}^m < k) = \sum_{n=0}^{k-1} P(t_{ij}^m = n), \qquad (10)$$

In markov processes, the sojourn time is usually considered to have an exponential distribution. Our use of a semi-markov model eliminates this constraint and is more reflective of real world processes.

Computing probabilities $P(t_{ij}^m < k)$ is relatively simple. For example, node m can measure all times t_{ij}^m whenever it moves from landmark i to landmark j. In that way, the distribution of the sojourn time probability is a discrete distribution. For instance, assume that we have 6 measurements for t_{ij}^m, which are 2, 4, 4, 5, 4, 6. Then the $P(t_{ij}^m < 5)$ is 2/3.

Next we describe how a node can determine the sojourn time distributions for all other nodes in the network.

4.4.2 History Information Exchange

To predict node mobility in the PER algorithms, every node needs to know other nodes' mobility history information. Specifically, the history mobility information is defined as a 5-tuple $\langle nodeID, P, S, T_{rec}, LandmarkID_{cur}\rangle$, where P is the transition probability matrix, S is the sojourn time probability distribution matrix, T_{rec} is the recording time when the record is generated, and $LandmarkID_{cur}$ is the recorded landmark where the node is located when the record is generated. Whenever two nodes become neighbors, they will exchange the history mobility records they have. A node adds the record of its new neighbor into its local database, and updates the history information with the new data by comparing the parameter T_{rec}. Note that it is possible that nodes only have a partial view of the whole network. However, when a node calculates $\phi_{ij}(k)$ for the neighbor and the destination, it is very possible that T_{rec} from their records be different, which means the start time for computation is different. Therefore, the equations for prediction in the above are not right any more. To improve the accuracy of prediction under this scenario, we utilize the following modifications.

When node A wants to send a packet to node B, node A first looks up B's history mobility information locally, which is $\langle nodeID_B, P_B, S_B, T_{recB}, LandmarkID_{curB}\rangle$. If A needs to know where node B is at time t, the following equation is used,

$$\begin{aligned} & \phi_{LandmarkID_{curB}j}(t - T_{recB}) \\ = \ & P(Z_t = j | Z_{T_{recB}} = LandmarkID_{curB}) for\ j \in L, \end{aligned}$$

where Z_t is the landmark where the node is at time t.

Similarly, node A can adjust the calculation of $\phi_{ij}(k)$ for a neighbor. The only difference is that it finds the neighbor's latest record, and replaces the start time with the recording time T_{rec} in the record, as well as the start landmark with the recorded landmark $LandmarkID_{cur}$ in the record. Note that the predicted time window for those two nodes may be different in this way, to make sure that the mobility behavior at the same future time spot is predicted.

The reader will notice that the size per node of the P matrix ($|L|^2$) and especially the S matrix could be large – $|L|^2 H$ for S, where H is the prediction window. Nevertheless, we expect in the real world these matrices to be very sparse due to the typical routine found in DTNs with social nature, where the node trajectory is almost deterministic with small deviations, such as the network of students in a campus or in public transportation.

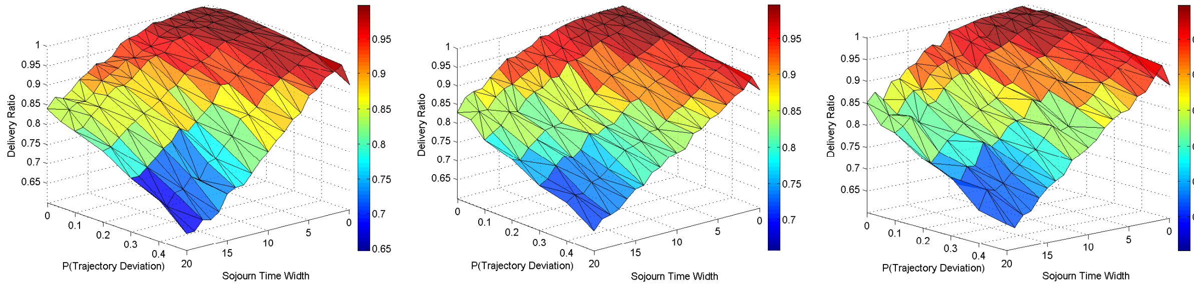
5. PERFORMANCE EVALUATION

In this section, we would like to evaluate our three PER algorithms, and contrast their performance against several simple single-copy routing algorithms and epidemic routing. Our main object is to investigate whether the three PER algorithms can increase the delivery ratio, compared to other single-copy routing algorithms. Also, we want to see how the three PER algorithms provide better end-to-end delivery latency.

5.1 Evaluation Methodology

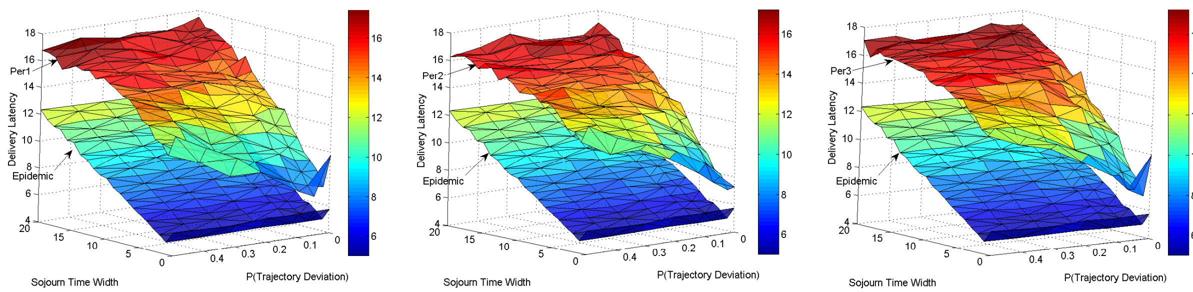
We have used a custom packet-based simulator implemented in Java to evaluate and compare the performance of the different routing protocols.

Mobility Model: In our simulation, we utilize a landmark based DTN model, where there are several predefined landmarks in the network. Nodes usually resolve around these landmarks. That is, nodes would stay in the neighborhood of a landmark or move to the neighborhood of other landmarks with their own preferred probability. Two nodes can only communicate when they are associated to the same landmark. In our case, we simulate scenarios with 12 landmarks and 30 nodes. Initially, nodes are uniformly distributed among the landmarks. Moreover, we assume every node has a trajectory deviation probability p, which is used to simulate nodes' semi-deterministic mobility behavior. That is, every node has a probability $1 - p$ to visit a landmark from where it is currently, and visit any other $|L| - 1$ landmarks with probability $p/(|L| - 1)$. We also require that the probability that a node moves from landmark i to landmark i is 0, $P_{ii} = 0$. p is a simulation parameter that varies from 0 to 0.5 with a step of 0.1.



(a) Per1 (b) Per2 (c) Per3

Figure 4: Delivery ratio for three PER algorithms.



(a) Per1 (b) Per2 (c) Per3

Figure 5: End-to-end delivery latency for three PER algorithms.

We define the maximum sojourn time that a node moves from a landmark to another as $2 + w$, where w is called the *sojourn time width window* and varies from 0 to 20 time units. The sojourn time of a node is uniformly picked in $[2, 2 + w]$.

Routing Protocols: We have implemented and compared the following routing schemes. Note that the names in the parentheses are used to refer to the routing schemes in result plots. We mainly focus on the single-copy schemes in our simulation study.

Epidemic routing ("Epidemic"): a node would spread the message it has to any nodes it encounters that have not seen it yet. Eventually, every node in the network will obtain a copy of that message. This protocol relies on multi-copy delivery, such that messages can reach the destination on multiple paths. We implement this approach to investigate the optimal end-to-end delivery latency between two nodes.

Utility-based routing ("Utility"): each node maintains a utility value for every other node in the network, based on a timer indicating the time elapsed since the two nodes last encountered each other. Here, the smaller the time elapsed is, the bigger the utility value will be. A node would forward the message only to the neighbor who has the larger utility for the destination.

Random selection ("Random"): a node would randomly pick a neighbor as a relay node to forward the message until the message reaches its destination.

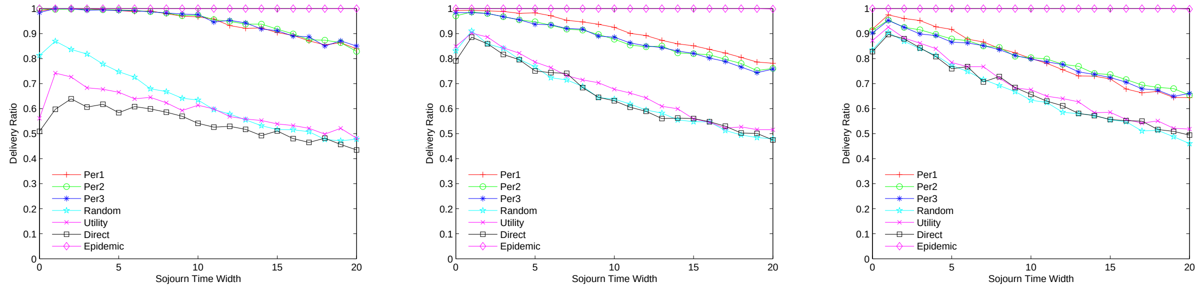
Direct delivery ("Direct"): source does not forward the message to anyone unless it encounters the destination.

PER ("Per$X$"): a node would distribute a message with the schemes described in this paper. Since we have three different criteria to indicate whether to forward messages for a node, we employ Per1, Per2, and Per3 to refer to our three schemes respectively. The prediction time window for the three PER algorithms is fixed to 60 time units.

Message Generation: we use the poisson distribution to model message generation in the network. In detail, we regulate that the average message arrival rate in the network is 10 time units. For each message, we randomly select a pair of nodes as source and destination respectively. In all scenarios considered, each message is assigned a TTL value of 40 time units.

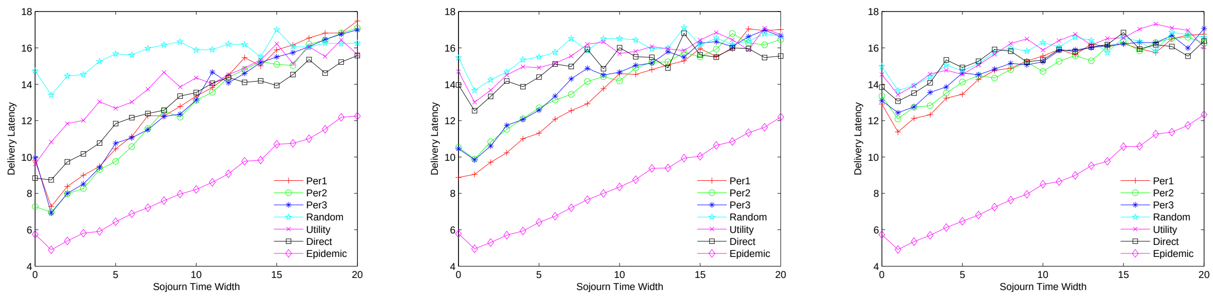
To apply our PER algorithms, a node needs to generate its transition probability matrix P and sojourn time probability distribution $S_{ij}(k)$ first. Note that at the beginning, the acquired P and $S_{ij}(k)$ is not stable, since the collected mobility history information to generate those two parameters is not sufficient enough. Therefore, to better evaluate the system performance, we run the simulation for a "warm-up period" to reach steady state and make the collected history mobility information sufficient enough to generate P and $S_{ij}(k)$. After that, the simulator runs 2048 time units for each scenario to collect data, and we run each scenario ten times to report the average.

The common goals of any DTN routing protocol is to maximize the delivery ratio, and to minimize the latency between source and destination. Therefore, to compare the performance, we use the following metrics: 1) *Delivery ratio* is defined as the ratio of the number of successfully deliv-



(a) Trajectory Deviation, p=0.0 (b) Trajectory Deviation, p=0.2 (c) Trajectory Deviation, p=0.5

Figure 6: Comparison of delivery ratio under different trajectory deviation probability p.



(a) Trajectory Deviation, p=0.0 (b) Trajectory Deviation, p=0.2 (c) Trajectory Deviation, p=0.5

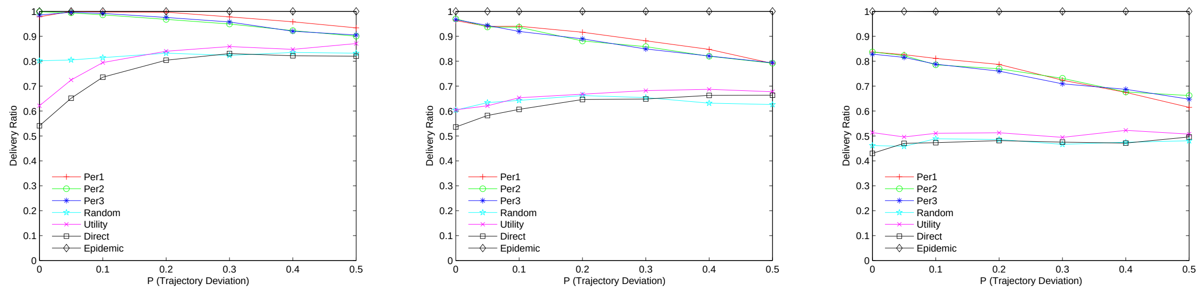
Figure 7: Comparison of delivery latency under different trajectory deviation probability p.

ered messages to the number of all messages generated in the network; 2) *Delivery latency* is the average end-to-end delivery latency between a pair of source and destination in the network.

5.2 Simulation Results

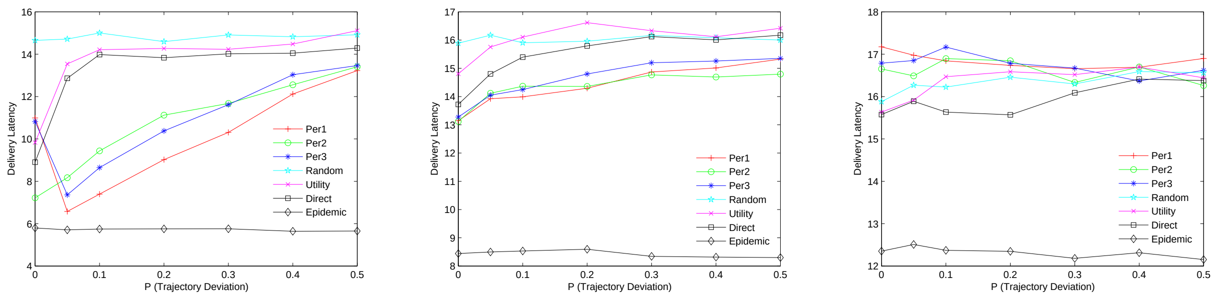
We first investigate the performance of three PER algorithms under different trajectory deviation probability p and the sojourn time window width w. Specifically, we would like to see how those two parameters affect the delivery ratio and the end-to-end delivery latency of the three PER algorithms. With this motivation, we plot those three PER algorithms' performance separately. Figure 4 records the delivery ratio of the three PER algorithms as a function of p and w. As can be seen there, all three PER algorithms have similar trends for the delivery ratio with p and w varying. The delivery ratio increases as p decreases and w declines. In other words, the more deterministic the node behavior is, and the less sojourn time a node has, the better the prediction of all three PER algorithms are. Actually, when p becomes 0, it indicates that a node only has one successive landmark to go from where it is. When w becomes 0 means a node moves more often. In the extreme case, where p is 0 and w is 0, the network is very deterministic and messages can always be successfully delivered with sufficient mobility in the network. There is a similar trend in Figure 5, which presents the end-to-end delivery latency of the three PER algorithms. To better evaluate, we also draw the delivery latency of epidemic routing as the optimal value, since epidemic routing always provides the lower limit for the delivery latency. We see that as p decreases and w declines, the delivery latency roughly decreases. Moreover, the three PER algorithms do not increase the delivery latency too much, compared to epidemic routing. The motivation of Figure 4 and Figure 5 is to give an overview of how p and w impact delivery ratio and delivery latency of the three PER routing algorithms.

Next, we compare the delivery ratio and the delivery latency of the three PER algorithms with other approaches under three different p scenarios with 0.0, 0.2, and 0.5 respectively. In the $p = 0.0$ case, nodes' mobility are very deterministic. Therefore, the nodes' future mobility prediction becomes accurate. For $p = 0.5$, the network's entropy is higher as nodes move more randomly and the trajectory prediction becomes less accurate. Case $p = 0.2$ is on the middle level. We just want to evaluate how the three PER algorithms perform compared to other schemes under those three scenarios. Figure 6 plots the delivery ratio under the three different p scenarios with different w. We see that all three PER algorithms gain larger delivery ratios than utility-based routing, random selection routing, and direct delivery approach. This is because, the three PER algorithms have better mobility prediction when forwarding messages, which could increase the possibility to contact the destination. Also, as w increases, the delivery ratios of all the routing schemes go down. A common reason is that because w rises, message delivery delay increases, even beyond the predefined TTL. Therefore, the delivery ratio goes down. But for the three PER algorithms, another reason is that when w



(a) Sojourn time width, w=0 (b) Sojourn time width, w=10 (c) Sojourn time width, w=20

Figure 8: Comparison of delivery ratio of routing protocols under different sojourn time window width w.



(a) Sojourn time width, w=0 (b) Sojourn time width, w=10 (c) Sojourn time width, w=20

Figure 9: Comparison of delivery latency of routing protocols under different sojourn time window width w.

increases, the accuracy of $S_{ij}(k)$ falls, which then noises the accuracy of the future mobility prediction. Besides, we find that, in $p = 0.0$, PER algorithms perform much better than other routing schemes, while in $p = 0.5$, the advantage is reduced and all routing schemes work very similar. In other words, the more deterministic the nodes' mobility behavior is, the better the three PER algorithms will perform.

Figure 7 records the delivery latency under the three different p scenarios where w varies. Although the three PER algorithms are designed to increase the deliver ratio in routing, their performance in terms of delivery latency is better too, compared to other routing protocols. We can see that in all the three scenarios, the PER algorithms have less delivery latency, even though, as p increases the delivery latency of the three PER algorithms becomes closer to other protocols. Thus, we believe that accurate prediction during messages' forwarding process could also help reduce the delivery latency. In addition, as w is going up, the delivery latency of the three PER algorithms is raised. The reason is obvious. Because nodes become less mobile, it will cost more time to deliver messages with nodes' mobility.

We then compare the performance of the three PER algorithms with other approaches under three different w with 0, 10, and 20, respectively. As explained before, w controls the width of the uniform distribution from which the node sojourn time is sampled. The intent behind this experiment is to check how those routing protocols perform under different nodes' mobility. Note that a larger w implies that nodes are less mobile. Thus, $w = 0$ means nodes transfer frequently among the neighborhoods of the landmarks in the network, while for $w = 20$, nodes' transition between nodes occurs less often, on average each (2+20)/2=11 time units. In Figure 8, we show the delivery ratio under the three different w scenarios where p varies. In all three scenarios, the three PER algorithms present better delivery ratio than other protocols. Additionally, as p increases, which implies that node mobility becomes less deterministic, the delivery ratio of the three PER algorithms falls down. But p does not influence the delivery ratio of other routing protocols. This trend indicates that when the node mobility is less random, applying the PER protocols yields better results.

Figure 9 summarizes the end-to-end delivery latency of all routing protocols under the three different w scenarios. We find that the higher the node mobility is and the less random the transition times are (lower w), the lower the latency is for PER protocols. When the width of the transition time window grows (w=20), the PER prediction accuracy falls and also the time spent by messages in queues grows. As a consequence, the latency of PER protocols grows comparable to that of the other protocols. Overall, PER algorithms are more effective when the randomness of the node trajectories is low, as seen in scenarios with reduced trajectory deviation probability (p) and sojourn time window width (w). In addition, similar to the trend in the delivery ratio figures, p does not influence the delivery latency of other routing protocols much. However, as p grows, which indicates that mobility becomes less deterministic, we see that the delivery latency of our three PER algorithms goes down.

6. CONCLUSIONS

In this paper, we propose the Predict and Relay scheme, an efficient routing scheme in DTNs. We introduce a time-homogeneous semi-markov process model to predict the future contacts of two specified nodes at a specified time. With this model, a node estimates the future contacts of its neighbors and the destination, and then selects a proper neighbor as the next hop to forward the message. This paper defines three different prediction functions to assist in choosing the proper neighbor for message delivery. Simulation results show that our approach raised the delivery ratio by relying on contact time prediction, compared to other traditional routing protocols. In addition, we see that our algorithms also reduce the delivery latency when routing in DTNs.

Future work remains to be done on the validations of our protocols on real data and applications in different environments. We plan to evaluate the PER algorithms on traces coming from real social contexts, such as campus networks. Moreover, we will work on addressing the assumptions, such as the zero transition time issue, and designing a multi-copy routing scheme with our model.

7. REFERENCES

[1] I. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci. A survey on sensor networks. In *IEEE Communications Magazine*, 2002.

[2] E. Brewer, M. Demmer, B. Du, M. Ho, M. Kam, S. Nedevschi, J. Pal, R. Patra, S. Surana, and K. Fall. The case for technology in developing regions. In *IEEE Computer*, 2005.

[3] N. Banerjee, M. Corner, and B. Levine. An energy-efficient architecture for dtn throwboxes. In *Proceedings of IEEE INFOCOM*, 2007.

[4] K. Fall. A delay-tolerant network architecture for challenged internets. In *Proceedings of ACM SIGCOMM*, 2003.

[5] P. Hui, J. Crowcroft, and E. Yoneki. Bubble rap: Social based forwarding in delay tolerant networks. In *Proceedings of ACM Mobihoc*, 2008.

[6] A. Balasubramanian, B. N. Levine, and A. Venkataramani. Dtn routing as a resource allocation problem. In *Proceedings of ACM SIGCOMM*, 2007.

[7] R. Shah, S. Jain, S. Roy, and W. Brunette. Data mules: Modeling a three-tier architecture for sparse sensor networks. In *Tech. Rep. IRS-TR-03-001, Intel Research Seattle*, 2003.

[8] Sensor networking with delay tolerance (SeNDT). http://down.dsg.cs.tcd.ie/sendt/.

[9] J. Wu, M. Lu, and F. Li. Utility-based opportunistic routing in multi-hop wireless networks. In *Proceedings of IEEE ICDCS*, 2008.

[10] A. Vahdat and D. Becker. Epidemic routing for partially connected ad hoc networks. In *Tech. Rep. CS-200006, Duke University*, 2000.

[11] J. Burgess, B. Gallagher, D. Jensen, and B. N. Levine. Maxprop: Routing for vehicle-based disruption-tolerant networks. In *Proceedings of IEEE INFOCOM*, 2006.

[12] I. Cardei, C. Liu, J. Wu, and Q. Yuan. DTN routing with probabilistic trajectory prediction. In *Proceedings of the International Conference on Wireless Algorithms, Systems and Applications (WASA)*, 2008.

[13] J. Ghosh, S. J. Philip, and C. Qiao. Sociological orbit aware location approximation and routing (solar) in dtn. Technical report, State University of New York at Buffalo, April 2005. 2005-12.

[14] K. Lee, M. Le, J. Haerri, and M. Gerla. Louvre: Landmark overlays for urban vehicular routing environments. In *Proceedings of IEEE WiVeC*, 2008.

[15] J. Yoon, B. Noble, M. Liu, and M. Kim. Building realistic mobility models from coarse-grained traces. In *Proceedings of ACM MobiSys*, June 2006.

[16] C. Liu and J. Wu. Routing in a cyclic mobispace. In *Proceedings of ACM MobiHoc*, 2008.

[17] X. Chen and A. L. Murphy. Enabling disconnected transitive communication in mobile ad hoc networks. In *Proceedings of ACM POMC*, 2001.

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

[19] M. Musolesi, S. Hailes, and C. Mascolo. Adaptive routing for intermittently connected mobile ad hoc networks. In *Proceedings of IEEE International Symposium on a World of Wireless Mobile and Multimedia Networks (WoWMoM)*, 2005.

[20] T. Small and Z. J. Haas. Resource and performance tradeoffs in delay tolerant wireless networks. In *Proceedings of WDTN*, 2005.

[21] T. Spyropoulos, K. Psounis, and C. S. Raghavendra. Single-copy routing in intermittently connected mobile networks. In *Proceedings of IEEE SECON*, 2004.

[22] P. Juang, H. Oki, Y. Wang, M. Martonosi, L. Peh, and D. Rubenstein. Energy-efficient computing for wildlife tracking: Design tradeoffs and early experiences with zebranet. In *Proceedings of ASPLOS-X*, 2002.

[23] T. Spyropoulos, K. Psounis, and C. Raghavendra. Spray and wait: An efficient routing scheme for intermittently connected mobile networks. In *Proceedings of WDTN*, 2005.

[24] K. Harras, K. Almeroth, and E. Belding-Royer. Delay tolerant mobile networks (DTMNs): Controlled flooding schemes in sparse mobile networks. In *Proceedings of Netwoking*, 2005.

[25] A. Lindgren, A. Doria, and O. Schelen. Probabilistic routing in intermittently connected networks. In *SIGMOBILE Mobile Computing and Communications Review*, 2003.

[26] X. Zhang, G. Neglia, J. Kurose, and D. Towsley. Performance modeling of epidemic routing. In *Proceedings of IFIP Networking*, 2006.

[27] B. Burns, O. Brock, and B. N. Levine. Mv routing and capacity building in disruption tolerant networks. In *Proceedings of IEEE INFOCOM*, 2005.

[28] J. LeBrun, C. Chuah, and D. Ghosal. Knowledge based opportunistic forwarding in vehicular wireless ad hoc networks. In *Proceedings of VTC Spring*, 2005.

[29] J. Leguay, T. Friedman, and V. Conan. Evaluating mobility pattern space routing. In *Proceedings of IEEE INFOCOM*, 2006.