

Message Forwarding in Cyclic MobiSpace: the Multi-copy Case

Cong Liu, Jie Wu, and Ionut Cardei
 Department of Computer Science and Engineering
 Florida Atlantic University
 Boca Raton, FL 33431
 cliu8@fau.edu, jie@cse.fau.edu, icardei@cse.fau.edu

Abstract—A key challenge of message forwarding in delay tolerant networks (DTNs) is to increase delivery rate and decrease delay and cost. When information for future connectivity is not available, opportunistic routing is preferred in DTNs in which messages are forwarded opportunistic/non-deterministically to nodes with higher delivery probabilities. Many real objects have non-deterministic but cyclic motions; however, few prior research work has investigated a multi-copy opportunistic message forwarding algorithm for DTNs with cyclic mobility patterns. Cyclic MobiSpace is a generalization of DTNs with cyclic mobility patterns. In this paper, we propose an optimal opportunistic multi-copy message forwarding algorithm in Cyclic MobiSpace. Specifically, we model a Cyclic MobiSpace as a state-space graph, and apply the *optimal stopping rule* to derive a delivery metric for each message state using the state-space graph. We perform simulation to compare our protocol, called Multi-copy Forwarding in Cyclic MobiSpace (MFC), against existing forwarding protocols, using UMassDieselNet trace. Simulation results show that, MFC delivers up to 100% more messages than the compared forwarding protocols under the same delay and forwarding cost.

Index Terms—Cyclic MobiSpace, delay tolerant networks (DTNs), simulation, trace.

I. INTRODUCTION

Delay tolerant networks (DTNs) are occasionally-connected networks that suffer from frequent network partitioning due to reasons such as high mobility, low density, short radio range, intermittent power, interference, obstruction, and attacks. Representative DTNs include sensor networks using scheduled connectivity, terrestrial wireless networks that cannot ordinarily maintain end-to-end connectivity, satellite networks with periodic connectivity, and underwater acoustic networks with moderate delays and frequent interruptions.

Routing in DTNs is an active research area. The Delay Tolerant Network Research Group (DTNRG) [1] has designed a complete architecture to support various protocols in DTNs. A DTN can be described abstractly using a space time graph [2] in which each edge corresponds to a contact. A *contact* is a period of time during which two nodes can communicate with each other. On the Internet, intermittent connectivity causes loss of data, whereas DTNs support communication between intermittently-connected nodes using the *store-carry-forward* routing mechanism.

Routing in DTNs poses some unique challenges compared to conventional data networks due to the uncertain and time-varying network connectivity. In [3], [2], optimal deterministic routes in a DTN can be found by constructing a space time graph with future connectivity information (oracle). In practical situations where no oracle is available to reveal future contacts, opportunistic routing [4], [5] is proposed in which one or more copies of a message is sent along different paths and each copy is forwarded opportunistically to the node where the message is estimated to have a higher chance of being delivered. Metrics for delivery can be either short-term metrics, which requires frequent updates, or long-term metrics, which are relatively stable over time.

Previous works have proposed a variety of long-term metrics including encounter frequency [5] and social similarity [6]. One advantage of long-term delivery metrics is that they are relatively stable once generated from historical connectivity information or prior knowledge on the contact pattern of nodes, avoiding the cost associated with frequent updates. On the other hand, many real objects have cyclic motion patterns, and therefore, it is possible and valuable in practice to increase the accuracy of a delivery metric by allowing it to be time-variant. In [7], Cyclic MobiSpace is defined to model networks with non-deterministic and cyclic mobility patterns. In Cyclic MobiSpace, a long term and cyclic delivery metric is proposed, called *expected minimum delay* (EMD), which is the expected time an optimal single-copy forwarding scheme takes to deliver a message generated at a specific time from a source to a destination. In this paper, we extend the notion of EMD from the single-copy to the multi-copy forwarding case, using the *optimal stopping rule* [8]. This extension is important because multi-copy forwarding scheme is more favorable in non-deterministic network scenarios such as DTNs. Also, as we will show, a delivery metric that is divided in the single-copy forwarding case (as are most existing delivery metric) is generally not accurate in the multi-copy forwarding case.

A MobiSpace is a Euclidean space (or a high dimensional space) where nodes can be either mobile or static and can communicate when moving into each other's transmission range. A *cyclic MobiSpace* [7] is a MobiSpace where nodal mobility exhibits a regular cyclic pattern and there exists a common motion cycle for all nodes. In a cyclic MobiSpace, if two nodes are often in contact at a particular time in previous cycles, then the probability that they will be in contact around

the same time in the next cycle is high. Cyclic MobiSpace is common in the real world since (1) most objects' motions exhibit regularity as they are repetitive, time-sensitive, and location-related; (2) a common motion cycle usually exists because most objects' motions are based on human-defined or natural cycles of time such as hour, day, and week.

With the extended EMD, we propose a routing protocol called *Multi-copy Forwarding in Cyclic MobiSpace* (MFC). Simulation is performed using UMassDieselNet trace [9], [10] to compare the routing performance of MFC against several protocols including spray-and-wait [11], a variation of delegation [12], and a variation of RCM [7]. Simulation results show that MFC delivers up to 100% more messages than the compared forwarding protocols under the same delay and forwarding cost. Simulation results also show that the performance of MFC degrades gracefully with reduced or partial routing information. This paper has the following contributions.

- 1) We propose the first cyclic delivery metric, the extended EMD (which we simply call EMD in the rest of the paper for convenience), for multi-copy message forwarding in cyclic MobiSpace.
- 2) We model the cyclic MobiSpace as a probabilistic state-space graph, and EMD by modeling message forwarding as an optimal stopping rule problem.
- 3) We perform real-trace-driven simulations to compare the routing performance of the proposed forwarding protocol with existing ones.

This paper is organized as follows. Section II reviews some background knowledge and introduces the basic idea of this paper. Section III discusses the calculation of EMD and the proposed MFC protocol in details. Section IV presents our simulation methods and results. Finally, Section V concludes the paper with directions for future research.

II. PRELIMINARIES AND OVERVIEW

A. Single-copy forwarding versus multi-copy forwarding

There are two types of message forwarding algorithms in the literature: single-copy forwarding [3], [2], [13] and multi-copy (opportunistic) forwarding [14], [4], [5], [9], [15], [16], [17], [6], [12]. In most DTNs, multi-copy opportunistic forwarding is used because deterministic future connectivity information is usually unavailable.

Single-copy forwarding is common in wired network and dense mobile ad hoc networks where connected and stable end-to-end paths between nodes generally exist. Single-copy forwarding may also be used in DTNs with deterministic nodal mobility. In single-copy forwarding, a message can be forwarded from the source to the destination by multiple nodes one after another, and each node that received the message will delete the message from the node's memory as soon as the node forwards the message; that is, only one copy of each message exists in the network at any time during the whole transmission process of the message.

In DTNs with non-deterministic mobility, multi-copy opportunistic forwarding allows multiple copies of the same message to be forwarded along different paths to deal with the fact

that none of these paths can guarantee delivery. Generally, the more copies of a message is made, the earlier the message is delivered. On the other hand, each additional copy is associated with an additional forwarding cost. Therefore, there is a trade-off between faster delivery and smaller overhead.

In this paper, we target the multi-copy forwarding schemes, and we focus on how to increase delivery rate and decrease delay with a constraint on the maximum number of forwardings associated with each message.

B. Representative DTN forwarding algorithms

Epidemic [14]. In epidemic, each node duplicates and forwards each message copy in its message buffer to every node it encounters that does not have the copy already. Epidemic effectively floods the network with copies of each message.

To reduce overhead, in many forwarding algorithms, a message is forwarded only from one node to a better node in terms of some delivery metric. Existing delivery metrics include encounter frequency [5], time elapsed since last encounter [9], [15], [16], [18], [19], location similarity [17], social similarity [6], [20], and geometric distance [21].

Delegation [12]. In delegation forwarding, each message copy maintains a threshold τ , which is initialized as the delivery metric value of the source node. When node i meets node j , if the metric value of node j is better than the copy's threshold τ , then (1) τ is set to the metric value of j , and (2) if j does not already has the copy, node i duplicates the copy and forwards it to node j . On average, delegation has an $O(\sqrt{N})$ forwarding cost [12]. However, forwarding cost being proportional to network size N may result in degraded performance in small networks (because of too few copies) and excessive cost in large networks (because of too many forwardings).

Spray-and-wait [11]. Spray-and-wait differs from epidemic in that it controls the number of copies (i.e. the total forwarding cost) of each message. A number L of logical tickets are given a message copy at its creation time. A copy can only be forwarded from node i to node j if either the copy owns $k > 1$ tickets or j is the destination. In the case that $k > 1$, after being forwarded, the two copies in i and j own $\lfloor k/2 \rfloor$ and $k - \lfloor k/2 \rfloor$ tickets, respectively. In terms of cost, spray-and-wait maintains a constant per message forwarding cost, which makes it scalable to arbitrarily large network size. On the other hand, the performance of spray-and-wait may degrades very fast as network size increases because it forwards copies arbitrarily without using any forwarding metric.

C. Motivation and overview

In this paper, we investigate the problem of how to minimize the expected delay in a forwarding scheme that controls its overhead as spray-and-wait does. Specifically, each message copy is initially associated with L logical tickets. When node i forwards to node j a copy with k tickets, the number of tickets owned by copy in i will be reduced to k_1 ($0 < k_1 < k$), and the copy in j will be given $k_2 = k - k_1$ tickets. To our best knowledge, no prior research has consider this ticket reassignment problem. In order to minimize delay, we not only

need to determine whether i forwards the copy but also how to select the optimal k_1 . Obviously, it is not necessary good for i to forward too many tickets to the first encountered node j because i may want to reserve more tickets for the better nodes that it is expected to meet in the future.

The multi-copy forwarding scheme gives rise to another problem to which existing works have not paid attention: a delivery metric that is not parameterized by a number of tickets is not accurate. This is because, for example, for a copy with one ticket, a node is a bad forwarder if the node is not a frequent contacting node of the message's destination. However, if the copy has two tickets, the same node can be a good forwarder in the case that the node has a frequent contacting node that is also a frequent contacting node of the message's destination.

In [7], a cyclic delivery metric, called *expected minimum delay* (EMD), is proposed as a cyclic delivery probability metric in Cyclic MobiSpace. EMD is the expected delay of a message assuming that optimal single-copy forwarding decisions are made in all nodes to minimize the delay of the message. In this paper, we extend EMD to the multiple-copy forwarding case. Note that our extended EMD may not be the minimum delay because of some assumptions that we will make. However, we continue to use the name of EMD for convenience.

The basic idea to calculate EMD is first to model a Cyclic MobiSpace as a time space graph, and then derive EMD by modeling each forwarding as an *optimal stopping rule problem* [8]. For each state in the state space graph, the optimal action can be selected regarding whether the message should be forwarded and, with a given k tickets, how to determine the new ticket reassignment k_1 and k_2 in order to minimize the expected delay. We then find the optimal actions and the EMDs for each k in each state using *backward induction* (a general solution of the optimal stopping rule problem) and *Markov decision process* (MDP) [22], [23].

The forwarding rule in MFC is simply described by the following example. At time t , when nodes i and j are connected, there is a message copy in node i , with k tickets. We first calculate the EMD D of the message in the case of no forwarding. We then calculate, in the case of forwarding, the EMDs D_1 and D_2 of the copies, in nodes i and j respectively, under all possible tickets assignment. Then, we select the assignment, k_1 and k_2 , such that the joint EMD $D' = \frac{1}{\frac{1}{D_1} + \frac{1}{D_2}}$ of the copies is maximized. If the $D' > D$, node i duplicates and forwards the message to j and assigns k_1 and k_2 to nodes i and j respectively.

D. Optimal stopping rule problem

In a stopping rule problem [8], you may observe a sequence of random variable X_1, X_2, \dots for as long as you wish, where X_1, X_2, \dots are random variables whose joint distribution is assumed to be known. For each stage $t = 1, 2, \dots$ after observing X_1, X_2, \dots, X_t , you may stop and receive the known reward y_t , or you may continue and observe X_{t+1} . The optimal stopping rule is to stop at some stage t to maximize the expected reward.

A stopping rule problem has a finite horizon if there is a known upper bound T on the number of stages at which one may stop. If stopping is required after observing X_1, \dots, X_T , we say the problem has horizon T . A finite horizon problem may be obtained as a special case of the general stopping rule problem by setting $y_{T+1} = \dots = y_\infty = -\infty$. In principle, such problems may be solved by the method of *backward induction*. Since we must stop at stage T , we first find the optimal rule at stage $T-1$. Then, knowing the optimal reward at stage $T-1$, we find the optimal rule at stage $T-2$, and so on back to the initial stage (stage 0). Let $V_t^{(T)}$ ($1 \leq t \leq T$) represent the maximum expected reward one can obtain starting from stage t . We define $V_T^{(T)} = y_T$ and then inductively for $t = T-1$, backward to $t = 0$,

$$V_t^{(T)} = \max \left\{ y_t, E(V_{t+1}^{(T)}) \right\}.$$

The meaning of the above equation is that, at stage t , we compare the reward for stopping, namely y_t , with the reward $E(V_{t+1}^{(T)})$ that we expect to be able to get by continuing and using the optimal rule for stages $t+1$ through T . The optimal reward is therefore the maximum of these two quantities, and it is optimal to stop at the earliest t when $y_t \geq E(V_{t+1}^{(T)})$.

Here, we use a house-selling scenario as a simple example for the finite horizon optimal stopping rule problem. Suppose you have a house to sell within T days. An offer comes in each day and X_t denotes the amount of the offer received on day t . X_1, \dots, X_T are independent and identically-distributed (i.i.d.), and are uniform over 0 to M . You may stop at any day t and receive $y_t = X_t$. You don't know the offers before they come in and you cannot recall a past offer. You need to find a stopping rule that maximizes the expected sales value.

Let us derive the optimal stopping rule using the backward induction method. Since we must sell the house by day T , the expected value $E(V_T^{(T)}) = E(y_T) = E(X_T) = \frac{M}{2}$. Inductively, at day t , $E(V_t^{(T)}) =$

$$E(\max \{ y_t, E(V_{t+1}^{(T)}) \}) = \int_0^M \max \{ x, E(V_{t+1}^{(T)}) \} dF(x) = \int_{E(V_{t+1}^{(T)})}^M x d \frac{x}{M} + \int_0^{E(V_{t+1}^{(T)})} E(V_{t+1}^{(T)}) d \frac{x}{M} = \frac{M^2 + (E(V_{t+1}^{(T)}))^2}{2M}.$$

Here, $F(x) = \frac{x}{M}$ is the cumulative distribution function of y_t , a uniform distribution over 0 to M . We can calculate $E(V_t^{(T)})$ inductively for $t = T-1$ down to 1. The optimal stopping rule is to sell the house on day t if $X_t \geq E(V_{t+1}^{(T)})$. In other words, the optimal stopping rule uses the expected reward in stage $t+1$ as the decision threshold for stage t .

III. MULTI-COPY FORWARDING IN CYCLIC MOBISPACE (MFC)

A. Discrete probabilistic (DP) contact

We divide the common motion cycle T of a network into small, fixed time slots. For each pair of nodes, we introduce a set of conceptual *discrete probabilistic contacts* (or simply DP contact). A DP contact is denoted as a tuple (t, p) , where

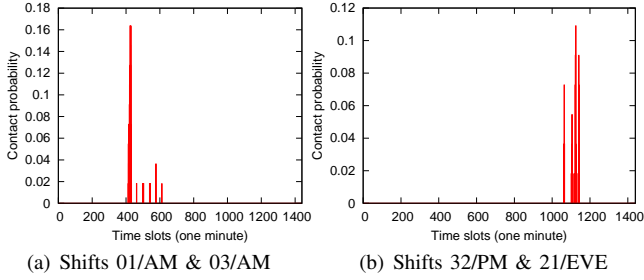


Fig. 1. Discrete probabilistic contacts between different sub-shifts in UMassDieselNet trace.

t is a time slot in T and p is the contact probability of two contacting nodes in the time slot t .

We use the DP contacts generated from UMassDieselNet trace [9], [10], as an example, where we consider each bus sub-shift as a node (see Section IV-A), one day as the common motion cycle, and one minute as the unit of a time slot. The DP contacts between sub-shifts 01/AM (the morning sub-shift of shift number 1) and 03/AM are shown in Figure 1(a). Figure 1(b) shows the DP contacts between another pair of sub-shifts.

From these figures, we can see that the DP contacts between two nodes in a realistic cyclic MobiSpace gather around a few consecutive time slots and the number of DP contacts is much smaller than the total number of time slots. The choice of time slot size is a trade-off between accuracy and routing information size, which is beyond the scope of this paper.

Figure 2(a) shows a sample cyclic MobiSpace that we will use throughout this paper. In this figure, nodes A , B , C , and D move in their trajectories, respectively. The motion cycle of A and B is 4 minutes, that of C is 3 minutes, and that of D is 2 minutes. Suppose nodes A , B , C , and D have contacts only during particular time slots in a common motion cycle $T = 12$ units, and these contacts are non-deterministic due to uncertainty in nodes' trajectories, communication failures, etc. The set of DP contacts for each pair of nodes in Figure 2(a) is shown in Figure 2(b). We assume the contact probabilities are uniformly 0.5.

B. State space graph

In a probabilistic state space graph $G(V, E)$, V is the set of states and E is the set of links. The state space graph G of the cyclic MobiSpace in Figure 2(a) is shown in Figure 3. G is generated as follows. For each node u , we create a set of states $\{t_i/u\}$ for each time slot t_i when u has one or more DP contacts. For example, four states $0/A$, $1/A$, $4/A$, and $8/A$ are created in G for node A in Figure 2(a), since A has three DP contacts $(0, 0.5)$, $(1, 0.5)$, $(4, 0.5)$, and $(8, 0.5)$ as shown in Figure 2(b). If node u has more than one contact (with different nodes) in the same time slot, then only one state is created for u corresponding to this time slot in G .

There are two types of links in G : *directional link* and *bidirectional link*. The directional link connects the consecutive states of the same node into a ring. For example, the four states of node A in G are connected into a ring by four directional

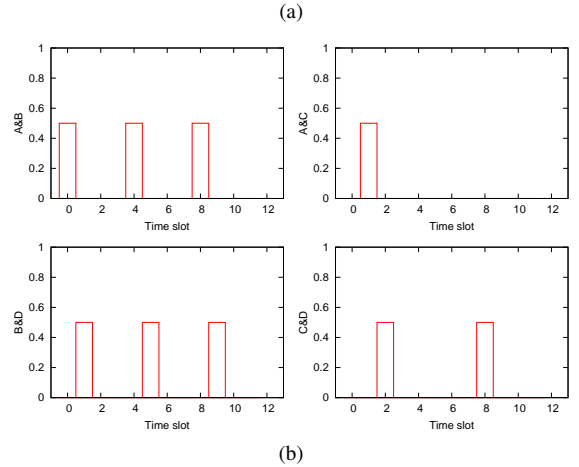
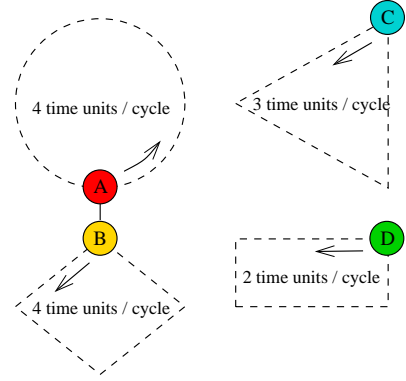


Fig. 2. A physical cyclic MobiSpace with a common motion cycle $T = 12$ (a). The discrete probabilistic contacts between all pair of nodes (b).

links shown by dashed lines in Figure 3. The bidirectional link in G is created corresponding to each DP contact. For each DP contact between nodes u and v in time slot t , a bidirectional link is created in G between states t/u and t/v (shown as a solid line in Figure 3).

Each state in G is a possible state of a message in the network, and each link in G is a possible state transition of a message. A message is in state $8/A$ if it is at node A in time slot 8. If the message is kept in node A from time slot 8 to time slot 0 (which is a time span of 4 slots), then the message transits from states $8/A$ to $0/A$ via a *directional link*. On the other hand, if the message is forwarded in time slot 8 from node A to node B , then it transits from states $8/A$ to $8/B$ via a *bidirectional link*.

Both types of links are labeled d/p , where d is the transition delay and p is the maximal transition probability. For a directional link, p is always equal to 1, which means a message can always be kept in a node without being forwarded. For a bidirectional link, p is equal to the contact probability of the corresponding DP contact, because the forwarding probability cannot exceed the contact probability between the nodes. For simplicity, we let $d = 0$ for all bidirectional links assuming that message forwarding is always restricted within a time slot. This assumption is based on the fact that the delay of message forwarding is generally much smaller than the size of the time slot itself.

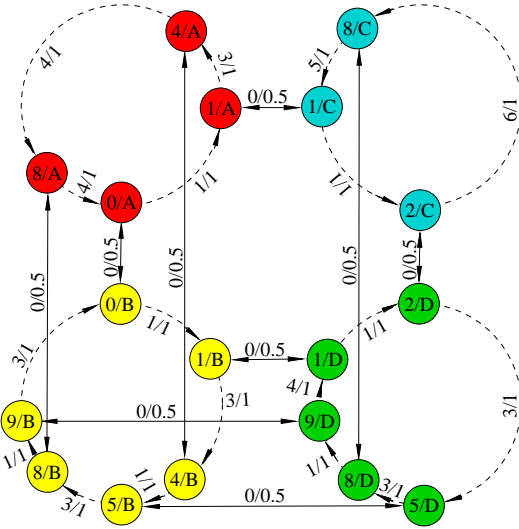


Fig. 3. The state-space graph of the cyclic MobiSpace in Figure 2(a).

C. The expected minimum delay (EMD)

In this subsection, we discuss the expected minimum delay (EMD) for the multi-copy forwarding case. The EMD of a message is the expected minimum time that it takes for one of the copies of the message to reach the destination, assuming all nodes that have a copy of the message make their optimal forwarding decisions. It is parameterized by the current node of the message, the message's destination, and the number of tickets owned by the message. We denote the EMD of a message as $D_{i,d,t,k}$, where i is the current node, t is the current time slot in T , d is the destination, and k is the number of ticket owned by the message.

For a given destination d , if a state belongs to d (i.e., $i = d$), then $D_{i,d,t,k} = 0$ for all t and k . Otherwise, for different d and k respectively, the EMDs $D_{i,d,t,k}$ states t/i are calculated using the Markov decision process (value iteration) [22], [23], which iteratively executes the following two steps until the EMDs converge.

- 1) Determine the optimal actions of each state t/i .
- 2) Update the $D_{i,d,t,k}$ s of the states under their optimal actions.

The optimal action of a state is its transition probabilities to other states in order to minimize EMDs. The optimal action is associated with the optimal forwarding decision which determines whether to forward a message copy in the state and how to reassign tickets between the two new copies after a forwarding. We will discuss optimal forwarding decisions for different number of tickets in the next two subsections. The updated EMD of a state s is $\sum_{s'} p_{s,s'} \times (d_{s,s'} + D_{s'})$, where s' is a state to which s can transit, $p_{s,s'}$ is the transition probability under the optimal action, and $D_{s'}$ is the EMD of state s' calculated in the previous iteration.

D. EMD with one ticket

With only one ticket, a message can only be forwarded to its destination. Also, to minimize delay, the message must be

TABLE I
EMDs WHEN C IS THE DESTINATION.

Tickets	1	2	3	4	5	6
1/D	7.0	7.0	5.05	4.6	4.44	4.29
2/D	6.0	6.0	4.66	4.34	4.23	4.17
5/D	9.0	9.0	6.32	5.67	5.46	5.33
8/D	6.0	6.0	4.28	3.87	3.75	3.67
9/D	11.0	11.0	7.56	6.74	6.5	6.33
0/A	13.0	13.0	7.26	6.03	5.68	5.33
1/A	12.0	12.0	7.01	6.08	5.78	5.63
4/A	21.0	21.0	11.02	9.15	8.55	8.27
8/A	17.0	17.0	9.55	7.92	7.44	7.14
0/B	∞	13.07	8.06	7.06	6.38	5.82
1/B	∞	12.13	8.58	7.6	7.09	6.51
4/B	∞	14.27	9.19	7.76	7.27	6.45
5/B	∞	13.27	8.88	7.8	7.33	6.68
8/B	∞	14.53	8.81	7.5	7.01	6.24
9/B	∞	13.53	8.79	7.79	7.26	6.65

Algorithm 1 Calculating EMD with $k = 1$ ticket

```

1: for each (node  $i$  in the network)
2:   if ( $i$  is destination  $d$ )
3:     for each (state  $t/i$  of node  $i$ )
4:        $D_{i,d,t,1} := 0$ 
5: for each (node  $i$  in the network)
6:   if ( $i$  is not destination  $d$ )
7:     while (not converged)
8:       for each (state  $t/i$  of node  $i$ )
9:         if (a link exists between  $t/i$  and  $t/d$ )
10:           $D_{i,d,t,1} := 0$ 
11:        else
12:           $t' :=$  the time slot of  $next(t/i)$ 
13:           $D_{i,d,t,1} := (t' - t) + D_{i,d,t',1}$ 

```

forwarded to its destination whenever possible. Therefore, the optimal action for a state s is (1) if there is a bidirectional link d/p between s and a state s' belonging to the destination, then s transits to s' with probability p and to the next state of s (denoted as $next(s)$) via the direction link with probability $1 - p$; (2) otherwise, s transits to $next(s)$ with probability 1. In Figure 3, $next(4/A) = 1/A$.

Let C be the destination, the EMDs of the states in the state space graph (Figure 3) are shown in Table I. When the number of ticket is one, all EMDs of the states belonging to node B are ∞ . This is because B does not have direct contact with C and therefore none of the states belonging to B have a directional link with a state belonging C . The EMD of state $1/A$ is $0.5 \times (0 + 0) + (1 - 0.5) \times (3 + 21) = 12$, where the transition probability from $1/A$ to $1/C$ is $p = 0.5$, the delay from $1/A$ to $1/C$ is 0, the EMD of $1/C$ is 0, the transition probability from $1/A$ to $4/A$ is $1 - p = (1 - 0.5)$, the delay from $1/A$ to $4/A$ is 3, and the EMD of $4/A$ is 21. As another example, the EMD of state $0/A$ is $1 \times (1 + 12) = 13$.

E. EMD with $k > 1$ tickets

When nodes i and j ($j \neq d$) are connected, for a message in i with $k > 1$ tickets, i has several forwarding options: (1) i does not forward the message to j , or (2) i duplicates and forwards the message to j , assigning k_1 ($0 < k_1 < k$)

tickets to the copy in i and $k_2 = k - k_1$ tickets to the copy in j . Associated with the above forwarding options, a message copy with k tickets can either (1) transit from state t/i to the next state t'/i ($t' > t$) with k tickets, or (2) transit to two states t'/i and t/j with k_1 ($0 < k_1 < k$) and $k_2 = k - k_1$ tickets respectively.

We can model each forwarding as an optimal stopping problem as follows. Node i has a message copy with $k > 1$ tickets that can be forwarded once. At the time of forwarding to another node j , the copy is logically regarded as being replaced by two new copies with $k_1 > 0$ and $k_2 > 0$ tickets respectively, and $k_1 + k_2 = k$. k_1 and k_2 are selected such that the joint EMD of the two copies is minimized. Candidate copy receivers j come in at time-slot t with probability p , where t can be the time slot when states t/i and t/j have a bidirectional link and p is the probability of the link. Upon connecting with j , i can either forward the copy to j or not. Whether i will forward depends on whether the joint EMD of the two copies (in the case of forwarding) is larger than the EMD of the original copy (in the case of no forwarding). For simplicity, we assume that EMDs are exponentially distributed, and therefore the joint EMD of two copies with EMDs D_1 and D_2 can be calculated by $\frac{1}{\frac{1}{D_1} + \frac{1}{D_2}}$.

Since a copy with k can either transit from a state to another state of the same node (in case of no forwarding) or transit to two states with k_1 and k_2 tickets ($0 < k_1, k_2 < k$), each EMD $D_{i,d,t,k}$ depends on (1) the EMDs $D_{i,d,t',k'}$ of the previous state of t/i for any $0 < k' \leq k$ and (2) the EMDs $D_{j,d,t',k''}$ of all states t'/j ($j \neq i$) for any $0 < k'' < k$. Therefore, we can use the backward induction to calculate the EMDs for k , from k equals 2 up to a require number of tickets. For each k , the EMDs of the states belonging to the same node can be updated independently from the EMDs (with k -tickets) of the states belonging to different nodes.

In the general case, a node can encounter several other nodes in a time-slot with different probability. To minimize EMD, the optimal action in a state is to maximize the transition probability to the states where the message has a larger EMDs. For example, suppose node i connects with nodes u , v , and w with probability p_1 , p_2 , and p_3 , respectively, at time slot t , the minimized joint EMDs when forwarding to nodes u , v , and w are D_1 , D_2 , and D_3 respectively, and $D_1 < D_2 < D_3 < D$, where D is the EMD in the case the message is not forwarded at t . The EMD of the message with k tickets in state t/i is then updated to its minimum value as $p_1 \times D_1 + (1 - p_1) \times p_2 \times D_2 + (1 - p_1) \times (1 - p_2) \times p_3 \times D_3 + (1 - p_1) \times (1 - p_2) \times (1 - p_3) \times D$.

The following is an example where we calculate the EMD of a message whose destination is C . We suppose the message has 2 tickets and is in state $1/B$ in the state space G (Figure 3). $1/B$ can transit to $1/D$ and $4/B$ with probability 0.5 and 1, respectively. Two forwarding options are possible: (1) in the case that the message is not forwarded, it transits to $4/B$ with 2 tickets, and the EMD of $4/B$ with 2 tickets is 14.27 (2) otherwise, the two copies transit to $4/B$ and $1/D$ respectively, and the only possible ticket reassignment is one ticket for each copy. Therefore, the joint EMD of the two copies is 7 since the EMDs of $4/B$ and $1/D$ with 1 ticket are ∞ and 7.

Algorithm 2 Calculating EMD with $k > 1$ ticket

```

1: for ( $k' := 2$  to  $k$ )
2:   for each (node  $i$  in the network)
3:     if ( $i$  is destination  $d$ )
4:       for each (state  $t/i$  of node  $i$ )
5:          $D_{i,d,t,k'} := 0$ 
6:     for each (node  $i$  in the network)
7:       if ( $i$  is not destination  $d$ )
8:         while (not converged)
9:           for each (state  $t/i$  of node  $i$ )
10:             $a := \text{optimal\_action}(t/i)$ 
11:             $D_{i,d,t,k'} = \text{update\_value}(t/i, a)$ 

```

Obviously, the optimal action is to maximize the probability of transiting to $1/D$, i.e., to maximize the probability of taking the second forwarding option. As a result, the resulting EMD of the message in state $1/B$ with 2 tickets is updated as $0.5 \times 7 + (1 - 0.5) \times (3 + 14.27) = 12.13$. The EMDs of all states in the state space graph (Figure 3) with different number of tickets are shown in Table I.

F. The MFC protocol

With EMDs, the forwarding rule in MFC is simple. When nodes i and j are connected, for a message copy in i with k tickets, we first determine the state s of the copy and get the EMD D of state s with k tickets. For each k' ($0 < k' < k$), we calculate the joint EMD $D_{k'}$ when the message copy is forwarded with k' tickets assigned to the copy in i and $k - k'$ tickets are assigned to the copy in j . Let $k'' = \arg \max_{k'} D_{k'}$. If $D_{k''} > D$, no forwarding is made. Otherwise, the message is forwarded with k'' tickets assigned to the copy in i and $k - k''$ tickets assigned to the copy in j .

G. Discussion

In Table I, the EMDs of nodes A and D for one and two tickets are the same. This is because, with two tickets, both A and D can only forward one copy to B , which does not have any contact with C . One may also notice that, in Figure 2(a), three copies should be enough to flood the network (except destination C). Therefore, EMDs with four or more tickets should equal those with three tickets. The fact that our EMDs keep decreasing as the number of tickets increases is due to the assumption we made when calculating the joint EMD: (1) we assumed that EMD is exponentially distributed, and (2) we did not consider the case where both nodes have copies of the same message when they are connected. In spite of these assumptions, the results of the simulation conducted in real trace show that MFC outperforms by far the other protocols that use delivery metrics defined in the single-copy forwarding case. We leave the study of joint EMD under other distributions, such as power-law distribution or composite distribution [24], [25], for future works.

The state of a message is determined by the hosting node and the current time slot. The state space graph G only contains *key states* that own bidirectional links. This is because

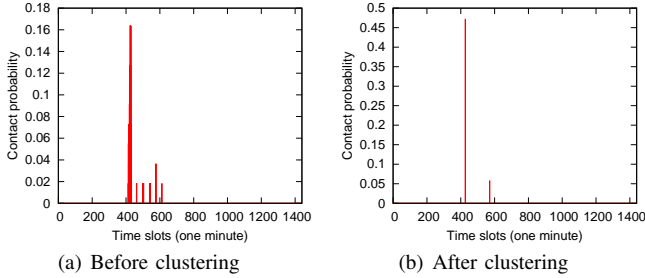


Fig. 4. Routing information compression by DP contact clustering.

the EMD of the other states can be easily calculated. For example, we can calculate the EMD of state 6/A (which is not a key state and is not shown in G) by adding 2 to the EMD of state 8/A. It can be proven that the Markov decision process applied on the state space graph converges when the initial EMD values of the states are properly initialized.

H. Routing information compression

The state space graph is built with DP contacts between all pairs of nodes in the network. In some network, the DP contact can be derive from prior knowledge about the mobility pattern of the network. In other networks, the DP contacts are generated from each node based on its contact histories with the other nodes. The DP contacts generated from a node is then disseminate to all the nodes in the network. In this case, the DP contacts are the routing information that the nodes need to exchange to run MFC protocol, and it is an implementation issue to reduce the size of this routing information to satisfy the communication and storage limitations in practical situations. To reduce the size of this routing information, which needs to be propagated in the network, we use a DP contact clustering algorithm to compress the size of DP contacts. Specifically, we use a virtual cluster-head DP contact to replace a group of nearby contacts. This method is effective in reducing the number of DP contacts while approximate the EMDs calculated from the original DP contact, because DP contacts tend to cluster together, as shown in Figures 1(a) and 1(b).

Clustering algorithms include hierarchical clustering and partitional clustering. Hierarchical clustering algorithms find successive clusters using previously established clusters. Hierarchical algorithms can be agglomerative (bottom-up) or divisive (top-down). Agglomerative algorithms begin with each element as a separate cluster and merge them into successively larger clusters. Divisive algorithms begin with the whole set and proceed to divide it into successively smaller clusters. Partitional clustering algorithms typically determine all clusters at once, but can also be used as divisive algorithms in the hierarchical clustering.

We use several agglomerative hierarchical clustering algorithms and the k -mean partitional clustering to perform contact clustering. In the agglomerative hierarchical clustering algorithms, each contact is initially a cluster, and iteratively the two clusters with the smallest distance are merged. The distance measurement between two clusters include (1) the

maximum distance between elements of each cluster, (2) the minimum distance between elements of each cluster, and (3) the mean distance between elements of each cluster. In our case, the distance between elements (DP contacts) is measured by the difference of their time-slots. We use a fixed number, k , of clusters to perform contact clustering in the case that the number of DP contacts are larger than k , then we observe how the routing performance degrades as k decreases.

An example of DP contact clustering is shown in Figures 4(a) and 4(b), where two cluster-head DP contacts are used to represent all original contacts that are close to each other. After replacing clusters of DP contacts with cluster-head DP contacts, we calculate EMDs from the state-space graph constructed from the cluster-head DP contacts. We desire those EMDs to be close to those calculated in the original time-space graphs with non-reduced contacts. We replace a set C_i of DP contacts with a cluster-head DP contact c_i as follows: (1) the contact probability $p(c_i)$ of c_i is the joint probability of the probabilities $p(c_j)$ of the contacts c_j in C_i , i.e.,

$$p(c_i) = 1 - \prod_{c_j \in C_i} (1 - p(c_j)),$$

and (2) the time-slot $t(c_i)$ of c_i is at the mean of the time-slots $t(c_j)$ of the contacts c_j in C_i weighted by the probabilities $p(c_j)$:

$$t(c_i) = \frac{\sum_{c_j \in C_i} p(c_j) \times t(c_j)}{\sum_{c_j \in C_i} p(c_j)}.$$

IV. SIMULATION

We evaluate our protocol, MFC, in the context of other routing algorithms using UMassDieselNet trace. Trace data available for the research community [26] include UMassDieselNet trace [9], [10], NUS student contact trace [27], Hagggle project [28], and MIT Reality Mining [29]. We will first briefly describe UMassDieselNet trace and our simulation method using this trace. Then we will present simulation results when using full routing information (DP contacts), reduced routing information (with DP contact clustering), and partial routing information (by randomly deleting DP contacts from the routing information), respectively. Simulation results show that MFC improves delivery rate, decreases end-to-end delay, and lowers forwarding cost in all traces when compared to other protocols.

We compare MFC against several other protocols including *epidemic* [14], *spray-and-wait* [11], *delegation** (a variation of delegation forwarding [12]), and *RCM** (a variation of RCM [7]). In *delegation**, we use the mean inter-meeting time of each node with the destination as forwarding quality. *RCM** is a simple multi-copy extension of RCM: when node i forwards a message copy with k tickets to node j , it assigns 1 ticket to the copy in i and $k - 1$ tickets to the copy in j ¹.

A. UMassDieselNet trace

Before presenting our simulation method and results, we give a brief description of UMassDieselNet [9], [10] testbed

¹We also implemented another version of *RCM** which reassigns tickets proportional to the quality values of the two nodes, but we found its performance is not as good as the current implementation of *RCM**.

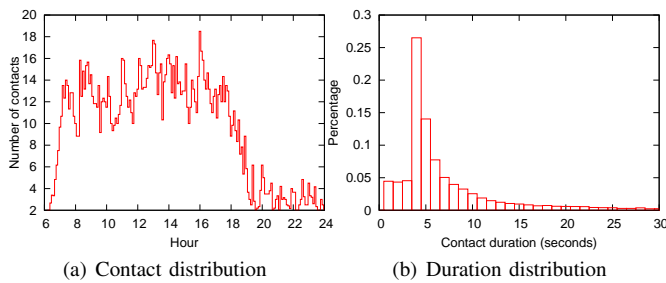


Fig. 5. Statistics in UMassDieselNet traces.

and the traces collected on this testbed. We then describe how we pre-process these traces.

In the UMassDieselNet bus system, consisting of 40 buses, the bus-to-bus contacts are logged, and the durations of these bus-to-bus contacts are relatively short. Our experiments are performed on traces collected over 55 days during the spring 2006 semester with weekends, spring break, and holidays removed due to reduced schedules. The bus system serves approximately ten routes. There are multiple shifts serving each of these routes. Shifts are further divided into morning (AM), midday (MID), afternoon (PM), and evening (EVE) sub-shifts. Drivers choose buses at random to run the AM sub-shifts. At the end of the AM sub-shift, the bus is often handed over to another driver to operate the next sub-shift on the same route or on another route. Unfortunately, all-bus-pairs contacts provided in the original traces show no discernible pattern. Significant effort is needed to obtain contacts at a sub-shift level that do exhibit periodic behavior.

We obtain the sub-shift level contact by the following steps. Each sub-shift has a fixed starting time (TIME_AT_GARAGE) and a fixed ending time (DRVR_CHNG) everyday. We obtain a mapping from sub-shifts to these times by parsing one of the dispatch records *DA_all.txt*. For example, sub-shift 21/AM (the AM sub-shift of shift 21) starts at 6:10AM and ends at 10:30AM. We also obtain a mapping from day and bus to the sub-shifts served by the bus on that day by parsing *DB_sheet.txt*. For example, on 3/1/2006, bus 3007 serves sub-shifts 21/AM, 21/MID, 21/PM, and 21/EVE. With the above two mappings, we translate 55 days of bus-to-bus contacts into contacts between sub-shifts. A virtual contact is created between two sub-shifts if a bus is handed over from one sub-shift to another. Figures 5(a) and 5(b) show the distribution of all contacts over a day and the distribution of the contact duration at the sub-shift level.

DP contacts between any pairs of sub-shifts is then generated from the 55 days of sub-shift based contacts. Examples of DP contacts between two pairs are shown in Figures 1(a) and 1(b). We set the time slot to be one minute. In the trace of a particular day, if two sub-shifts have one or more contact during the same time slot as a discrete probability contact, then the contact probability of the discrete probability contact is increased by $\frac{1}{55}$. With DP contacts, we can generate the probabilistic time-space graph and the probabilistic state-space graph for RCM, and the inter-meeting time between nodes used by delegation*. Details are omitted here.

TABLE II
SIMULATION SETTING.

parameter name	default	range
tickets	5	1~10
cluster-head DP contact %	-	1~5
routing info %	100%	10%~100%
simulation time	4 days	
simulation runs	55 times	
message rate	3 msgs/hr/node	
message creation time	during day 1	
number of sub-shift	92	
number of messages	6,624/day	
time slot unit	minute	
motion cycle	day	

The default settings of the UMassDieselNet trace simulation are shown in Table II. In different groups of simulations, messages are created with 1 to 10 tickets in the first and the second simulations with full routing information and reduced information. In the third simulation with partial routing information, we fixed the initial number of tickets to 5. We use 55 days of traces to run 55 simulations. Each simulation starts at one of the 55 days and last for the consecutive 4 days (day 55 is followed by day 1). Every node (sub-shift) sends a message to a random destination node every 20 minutes in the first day of the 4 days. That is, every of the 93 sub-shift sends 72 messages and totally 6,624 messages are sent in each simulation.

B. Simulation results with full routing information

We show the delivery rates, delays, and costs of the forwarding protocols we compared. Obviously, delivery rates of all protocols that use logical tickets increase as the number of tickets increases, which is shown in Figure 6(a). These figures show that MFC has a higher delivery rate than all other protocols except Delegation. Delegation makes forwarding decision solely based on the forwarding quality of the nodes and the threshold of the message, and it does not use logical tickets. Therefore, its performance is not affected by L .

From Figure 6(a), we observe that MFC is much better than RCM*, while RCM* is only slightly better than spray-and-wait. The delivery rate of MFC is almost 100% greater than that of RCM*. Although RCM* uses an optimal single-copy forwarding metric [7], RCM* is only about 10% better than spray-and-wait which does not use any forwarding at all. The reason is that, from a single-copy forwarding metric, we can only tell which node is better, but we cannot tell how forwarding tickets are reassigned to message copies to maximize delivery or to minimize delay.

The comparison of delays and costs are shown in Figures 6(b) and 6(c). We use the number of copies as the forwarding cost because the number of copies always equals the one plus the number of forwardings. For example, when the number of ticket is one, only the source and the destination can get the message, and therefore the number of copies is two and the number of forwarding is one. We only compare the delays and costs of the messages that are delivered by all protocols since those of the others are not measurable. The delay and cost of Delegation seems to decrease as the number of tickets

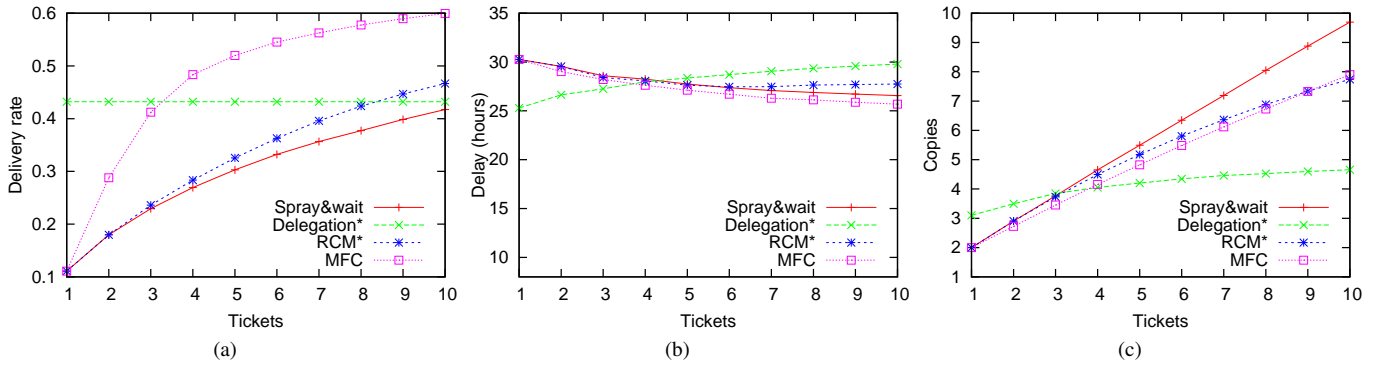


Fig. 6. Routing performance with different number of tickets.

TABLE III
DP CONTACTS VERSUS NUMBER OF CLUSTER-HEADS.

cluster-heads	1	2	3	4	5	∞
DP contacts	1,110	1,998	2,743	3,396	3,931	7,062

decreases. This is not true because when the number of tickets is small, the message get delivered is relatively easier to be delivered, and thus they have smaller delays and costs. For the other protocols, as expected, the delays decreases and the number of tickets increases. Except delegation, the number of copies of all is very close to the number of forwarding tickets.

C. Simulation results with reduced routing information

We reduce the amount of routing information (DP contacts) using different clustering algorithms as presented in Section III-H. We found that these clustering algorithms are not different from each other in terms of their impact on routing performance. Therefore, we only shown the result of using the maximum distance agglomerative hierarchical clustering algorithm. We vary the maximum number of cluster-head DP contacts of two nodes between 1 to 5.

Table III shows the amount of reduced DP contacts under different number of maximum cluster-heads allowed in the clustering algorithms, which is significant. Figures 7(a), 7(b), and 7(c) compares the performance of MFC using and not using DP contacts clustering algorithm. In these figures, $MFC(k)$ represents the situation with k maximum cluster-heads. It is surprising to see that, when k ranges from 1 to 5, the delivery rate are almost the same. This is probably because most pair of buses in the trace meet at the intersection of their routes, and therefore their DP contacts are clustered around a single point (as indicated by Figures 1(a) and 1(b)) and can be represented a single cluster-head DP contacts without much effect on the resulting EMDs. The fact that MFC's delivery rate reduces 5% when using 1/7 of routing information shows that the our DP contact compression using clustering is efficient.

D. Simulation results with partial routing information

In our simulation using partial routing information, we perform simulation using DP contacts that are randomly deleted. The amount of DP contacts that is deleted ranges

from 10% to 100%. As shown in Figure 8(a), the delivery rate of all protocols, except spray-and-wait, degrade as the percentage of routing information decreases. It is amazing that with only 10% of routing information, MFC delivers 80% of the message with 100% routing information. Compared with delegate, the MFC degrades more gracefully as the percentage of routing information decreases. The fact that delegation delivers less message than spray-and-wait is because its number of forwardings is much smaller as shown in Figure 8(c).

Figures 8(b) and 8(c) show that the delays and costs are hardly affected by the percentage of routing information. These result shown that while MFC is designed on networks with regular mobility, the performance of MFC is good when a small percentage of routing information is available. In future works, we will study the performance of MFC in network with less regular mobility, and we expect that MFC keeps a good performance when only a small amount of routing information is timely.

E. Summary of simulation

To sum up, MFC outperforms the other routing protocols that use historical connectivity information (DP contacts), in terms of both delivery rate, delay and forwarding cost. MFC keeps a superior routing performance using reduced or partial routing information, which shows the good scalability of MFC.

V. CONCLUSION

In this paper we presented the first research investigating a new cyclic delivery metric for multi-copy forwarding in cyclic MobiSpace. We modeled a Cyclic MobiSpace as a state-space graph, and applied the optimal stopping rule to derive an extended delivery metric, EMD, for each message. We performed simulation to compare the proposed protocol, MFC, against several forwarding protocols, using UMassDieselNet trace. Simulation results show that, MFC outperforms other protocols in terms of both delivery rate, delay, and cost. Our future work following this paper include more accurate calculation method of the joint EMD and conducting extensive simulation evaluation on other trace data and comparing MFC with more protocols based .

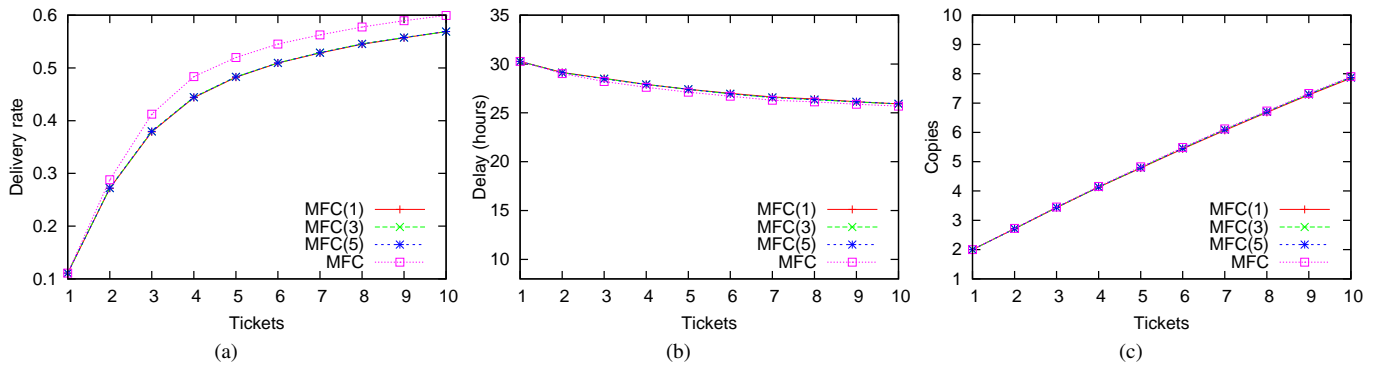


Fig. 7. Routing performance with reducing routing information.

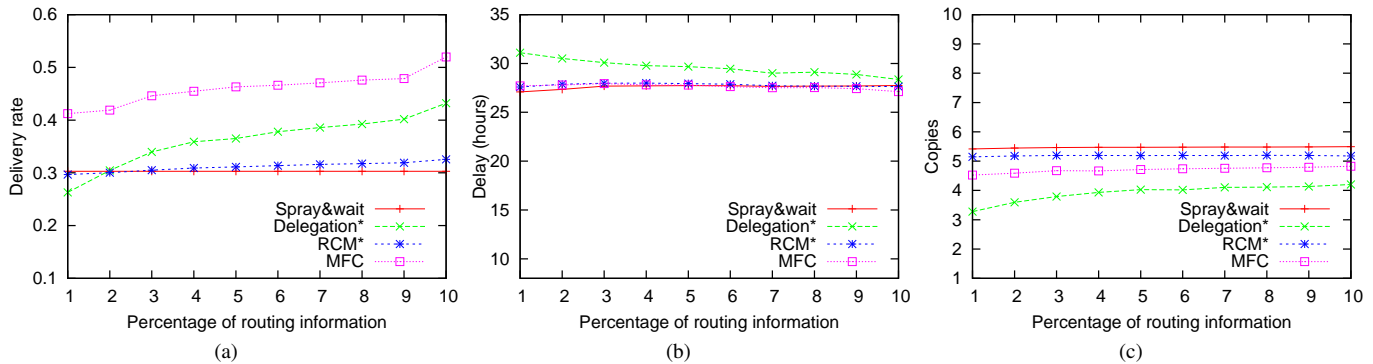


Fig. 8. Routing performance with partial routing information.

REFERENCES

- [1] V. Cerf, S. Burleigh, A. Hooke, L. Torgerson, R. Durst, K. Scott, K. Fall, and H. Weiss. Delay-tolerant Network Architecture. In *Internet draft: draft-irrf-dtnrg-arch.txt*, DTN Research Group, 2006.
- [2] S. Merugu, M. Ammar, and E. Zegura. Routing in Space and Time in Network with Predictable Mobility. In *Technical report: GIT-CC-04-07*, College of Computing, Georgia Tech, 2004.
- [3] S. Jain, K. Fall, and R. Patra. Routing in a Delay Tolerant Network. In *Proc. of ACM SIGCOMM*, 2004.
- [4] J. Haas, J. Y. Halpern, and L. Li. Gossip-Based Ad Hoc Routing. In *Proc. of IEEE INFOCOM*, 2002.
- [5] A. Lindgren, A. Doria, and O. Schelen. Probabilistic Routing in Intermittently Connected Networks. *Lecture Notes in Computer Science*, 3126:239–254, August 2004.
- [6] D. Elizabeth and H. Mads. Social Network Analysis for Routing in Disconnected Delay-Tolerant MANETs. In *Proc. of ACM MobiHoc*, 2007.
- [7] C. Liu and J. Wu. Routing in a Cyclic MobiSpace. In *Proc. of ACM MobiHoc*, 2008.
- [8] Optimal Stopping and Applications. <http://www.math.ucla.edu/~tom/Stopping/Contents.html>.
- [9] J. Burgess, B. Gallagher, D. Jensen, and B. N. Levine. MaxProp: Routing for Vehicle-Based Disruption-Tolerant Networking. In *Proc. of IEEE INFOCOM*, 2006.
- [10] X. Zhang, J. F. Kurose, B. Levine, D. Towsley, and H. Zhang. Study of a Bus-Based Disruption Tolerant Network: Mobility Modeling and Impact on Routing. In *Proc. of ACM MobiCom*, 2007.
- [11] T. Spyropoulos, K. Psounis, and C. Raghavendra. Spray and Wait: An Efficient Routing Scheme for Intermittently Connected Mobile Networks. In *Proc. of ACM WDTN*, 2005.
- [12] V. Erramilli, M. Crovella, A. Chaintreau, and C. Diot. Delegation Forwarding. In *Proc. of ACM MobiHoc*, 2008.
- [13] C. Liu and J. Wu. Scalable Routing in Delay Tolerant Networks. In *Proc. of ACM MobiHoc*, 2007.
- [14] A. Vahdate and D. Becker. Epidemic Routing for Partially-connected Ad Hoc Networks. In *Technical Report*, Duke University, 2002.
- [15] T. Spyropoulos, K. Psounis, and C. Raghavendra. Spray and Focus: Efficient Mobility-Assisted Routing for Heterogeneous and Correlated Mobility. In *Proc. of IEEE PerCom*, 2007.
- [16] A. Balasubramanian, B. N. Levine, and A. Venkataramani. DTN Routing as a Resource Allocation Problem. In *Proc. ACM SIGCOMM*, 2007.
- [17] J. Leguay, T. Friedman, and V. Conan. DTN Routing in a Mobility Pattern Space. In *Proc. of ACM WDTN*, 2005.
- [18] H. Dubois-Ferriere, M. Grossglauser, and M. Vetterli. Age Matters: Efficient Route Discovery in Mobile Ad Hoc Networks Using Encounter Ages. In *Proc. of ACM MobiHoc*, 2003.
- [19] M. Grossglauser and M. Vetterli. Locating Nodes with Ease: Last Encounter Routing in Ad Hoc Networks through Mobility Diffusion. In *Proc. of IEEE INFOCOM*, 2003.
- [20] P. Hui, J. Crowcroft, and E. Yoneki. BUBBLE Rap: Social-based Forwarding in Delay Tolerant Networks. In *Proc. of ACM MobiHoc*, 2008.
- [21] U. Acer, S. Kalyanaraman, and A. Abouzeid. Weak State Routing for Large Scale Dynamic Networks. In *Proc. of ACM MobiCom*, 2007.
- [22] R. Bellman. *Dynamic Programming*. Princeton University Press, 1957.
- [23] R. A. Howard. *Dynamic Programming and Markov Processes*. The MIT Press, 1960.
- [24] H. Cai and D.Y. Eun. Crossing Over the Bounded Domain: From Exponential to Power-law Inter-meeting Time in MANET. In *Proc. of ACM MobiCom*, 2007.
- [25] H. Cai and D.Y. Eun. Toward Stochastic Anatomy of Inter-meeting Time Distribution under General Mobility Models. In *Proc. of ACM MobiHoc*, 2008.
- [26] CRAWDAD data set. Downloaded from <http://crawdad.cs.dartmouth.edu/>.
- [27] V. Srinivasan, M. Motani, and W. T. Ooi. Analysis and Implications of Student Contact Patterns Derived from Campus Schedules. In *Proc. of ACM MobiCom*, 2006.
- [28] J. Scott, R. Gass, J. Crowcroft, P. Hui, C. Diot, and A. Chaintreau. CRAWDAD data set cambridge/haggle (v. 2006-09-15). <http://crawdad.cs.dartmouth.edu/cambridge/haggle>, September 2006.
- [29] L. Song and D. F. Kotz. Evaluating Opportunistic Routing Protocols with Large Realistic Contact Traces. In *Proc. of ACM CHANTS*, 2007.