

Scalable Routing in Cyclic Mobile Networks

Cong Liu and Jie Wu, *Fellow, IEEE*

Abstract—The nonexistence of an end-to-end path poses a challenge in adapting traditional routing algorithms to delay-tolerant networks (DTNs). Previous works have covered centralized routing approaches based on deterministic mobility, ferry-based routing with deterministic or semideterministic mobility, flooding-based approaches for networks with general mobility, and probability-based routing for semideterministic mobility models. Unfortunately, none of these methods can guarantee both scalability and delivery. In this paper, we extend the investigation of scalable deterministic routing in DTNs with repetitive mobility based on our previous works. Instead of routing with global contact knowledge, we propose a routing algorithm that routes on contact information compressed by three combined methods. We address the challenge of efficient information aggregation and compression in the time-space domain while maintaining critical information for efficient routing. Then, we extend it to handle a moderate level of uncertainty in contact prediction. Analytical studies and simulation results show that the performance of our proposed routing algorithm, *DTN Hierarchical Routing* (DHR), is comparable to that of the optimal time-space Dijkstra algorithm in terms of delay and hop count. At the same time, the per-node storage overhead is substantially reduced and becomes scalable.

Index Terms—Contact, delay-tolerant networks (DTNs), delivery, hierarchical routing, motion cycle, scalability, simulation.

1 INTRODUCTION

DELAY-TOLERANT networks (DTNs) are occasionally connected networks that may suffer from frequent partitions. Representative DTNs include sensor-based networks using scheduled intermittent connectivity, terrestrial wireless networks that cannot ordinarily maintain end-to-end connectivity, satellite networks with moderate delays and periodic connectivity, and underwater acoustic networks with moderate delays and frequent interruptions due to environmental factors. Our interests focus on a special kind of DTN where the mobility patterns of the nodes are roughly cyclic, hence the name Cyclic Mobile Networks (CMNs). Snapshots of a CMN over a period of 6 minutes are shown in Fig. 1, where nodes 1, 3, and 4 have circular trajectories, and nodes 2, 5, and 6 are static nodes. The motion cycle of node 1 is 2 minutes and that of nodes 3 and 4 is 3 minutes. The network has a motion cycle of 6 minutes since the same snapshot of the network recurs once every 6 minutes. For a message that is in node 1 and is heading for node 6 at minute 0, the optimal routing scheme will send it to node 3 immediately. Then, the routing scheme allows node 3 to store the message until minute 2 when node 3 gets the opportunity to forward the message to node 6.

The DTN Research Group (DTNRG) has designed a complete architecture to support various protocols in DTNs [2]. A DTN can be described abstractly using a graph. Each edge in this graph contains a set of contacts. A *contact* is a period of time during which two neighboring nodes can communicate with each other. Several types of contacts are defined in [2]. Among them are the *persistent contacts* between

the persistently connected nodes and the *predicted contacts* that are intermittently available. Their availability is predicted based on the history of previously observed contacts or some other information. A persistent contact can be between two static nodes (e.g., between nodes 5 and 6 in Fig. 1), between one static and one mobile one (e.g., nodes 1 and 2), or between two mobile nodes (e.g., nodes 3 and 4), whereas a predicted contact can be either between one static and one mobile node (e.g., nodes 3 and 6) or between two mobile nodes (e.g., nodes 1 and 3). All predicted contacts in Fig. 1 repeat with frequencies of at least once every 6 minutes. Fig. 2a is the graph model of the physical network in Fig. 1. In this figure, a thick line represents a persistent contact, and a thin line represents a predicted contact.

Routing in DTNs poses some unique challenges compared to a conventional data network due to the uncertainty and time-varying nature of network connectivity. The additional complexity of the time dimension significantly complicates the routing decision. Centralized routing approaches based on deterministic mobility include [9] and [13]. Ferry-based routing includes those with deterministic [18], [22] and semideterministic [20] mobility. Flooding-based approaches for networks with general mobility include [7] and [19]. Probability-based routing without a message delivery guarantee for semideterministic mobility models include [6], [10], [11], and [1]. Like many of the previous works, this paper focuses on deterministic mobility where the mobility of the nodes is predetermined, and therefore, the contacts are predictable.

On the other hand, many hierarchical routing approaches [3], [5], [14], and [16] have been proposed in MANETs, where the (virtual) hierarchical network constructed by multilevel clustering is used as a compressed topology abstraction for routing. Based on our previous work [12], we adapt the hierarchical network, which was previously only applicable in relatively static and connected networks, to our DTN model for CMNs. The challenge is to effectively represent the time-space information in the

• The authors are with the Department of Computer Science and Engineering, Florida Atlantic University, Boca Raton, FL 33431. E-mail: gzccong@gmail.com, jie@cse.fau.edu.

Manuscript received 1 May 2008; revised 10 Sept. 2008; accepted 18 Sept. 2008; published online 29 Sept. 2008.

Recommended for acceptance by P. Srimani.

For information on obtaining reprints of this article, please send e-mail to: tpds@computer.org, and reference IEEECS Log Number TPDS-2008-05-0161. Digital Object Identifier no. 10.1109/TPDS.2008.218.

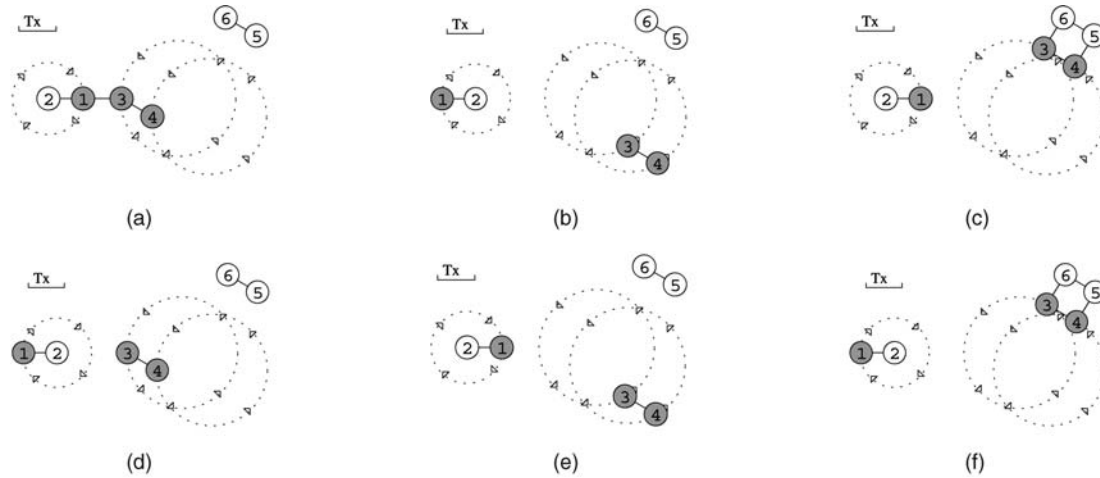


Fig. 1. A CMN in its motion cycle of 6 minutes. (a), (b), (c), (d), (e), and (f) Snapshots of the network at the minute mark. The transmission range T_x is shown in the figure. (a) Minute 0. (b) Minute 1. (c) Minute 2. (d) Minute 3. (e) Minute 4. (f) Minute 5.

hierarchical structure. There is no previous work in DTNs that uses the hierarchical network to compress the time-space information and use it as a time-space topology abstraction in routing. We propose the first routing protocol in DTNs with cyclic and deterministic mobility that guarantees both scalability and delivery.

Since it is difficult, if not impossible, for a scalable and delivery-guaranteed algorithm to obtain the necessary information in order to make a routing decision in a network of completely random movement, we focus on CMNs, which are DTNs with cyclic nodal mobilities. We first propose a simplified DTN model in which each node is either static or has a strict repetitive motion. We name our proposed routing protocol in this model the *DTN Hierarchical Routing* (DHR). We address the challenge of efficient information aggregation and compression in the time-space domain while maintaining critical information for efficient routing. We then relax the assumption of strict repetitive motion in the model by allowing a certain degree of uncertainty in nodal mobility. Accordingly, a heuristic is added to DHR to handle the uncertainty.

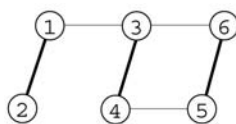
We achieve scalability in our model by using three combined methods to compress the time-space topology information. First, we construct a hierarchical network by performing multilevel clustering. We aggregate related contact information to the links in the hierarchical network. In order to route, each node only needs to have partial global topology information (consisting of the contact

information stored in some of the links in the hierarchical network) whose size is $O(\log(N))$ in the network of size N . The selection of each cluster head becomes more important here. In general, a cluster head should be situated at the “center” of the cluster to provide a better network topology approximation in terms of average delay.

Two other methods are used to compress the contact information aggregated into the links in the hierarchical network. The first one is to reduce the complexity in the space domain through a redundant contact information removal (CIR) method, which removes the contact information in a link that contributes little to the link’s underlying time-variant shortest path. The second method is to reduce the complexity in the time domain by stopping contact information aggregation in the links above a certain level, i.e., to remove the time dimension of the time-space information in the upper levels of the hierarchical network. Experimental results show that both of these compression methods have limited effects on routing performance. To accommodate uncertainty in the occurrence of the predicted contacts, we extend our model and algorithms by considering contact probability in the estimation of the end-to-end delays.

The contributions of our work are summarized as follows:

- We present a simplified DTN model for CMNs and with which multilevel clustering can be performed to build a hierarchical network.



(a)

Node	Contacts $(n_i, n_j, T_{ij}, t_s, t_d)$
1	$(1, 2, -, -, -), (1, 3, 6, 0, 0.1)$
2	$(2, 1, -, -, -)$
3	$(3, 4, -, -, -), (3, 1, 6, 0, 0.1), (3, 6, 3, 2, 0.1)$
4	$(4, 3, -, -, -), (4, 5, 3, 2, 0.1)$
5	$(5, 6, -, -, -), (5, 4, 3, 2, 0.1)$
6	$(6, 5, -, -, -), (6, 3, 3, 2, 0.1)$

(b)

Fig. 2. (a) The simplified DTN model for the DTN in Fig. 1. (b) Contacts of the nodes.

- Based on this model, we build the first hierarchical network in DTNs in which the time-varying nature of the physical topology is reflected by the time-variant information (the contact information) maintained by the links in the hierarchical network.
- We develop compression algorithms that reduce the amount of information in the hierarchical links without much loss or significant impact on the routing performance.
- We devise a hierarchical routing algorithm based on our hierarchical network. Simulation results support that in our DHR algorithm, the size of the per-node information is small and scalable, and the routing performance approximates that of the optimal time-space Dijkstra algorithm.
- We increase the robusticity of DHR by an extension, DHR+, which considers the uncertainty in predicted contact information. Simulation results show that the routing performance of DHR+ approximates that of flooding in a network with a moderate level of uncertainty.

The rest of the paper is organized as follows: A simplified DTN model is given in Section 2. Section 3 presents a framework of multilevel clustering and hierarchical routing. Section 4 proposes the DHR routing algorithm in our DTN model, as well as other routing information compression methods. Section 5 models uncertainty in node mobility based on our DTN model and proposes the extended DHR (DHR+). Section 6 presents an analysis on the routing performance and the scalability of our algorithm. Simulations and results are displayed in Section 7. Section 8 discusses related works. Finally, Section 9 concludes the paper.

2 A SIMPLIFIED DTN MODEL FOR CYCLIC MOBILE NETWORKS

We model a CMN as a time-varying graph where vertices are nodes and edges are sets of (representative) contacts. Our model allows only static nodes and nodes with strict repetitive motions. A strict repetitive motion means that the node moves on a predetermined trajectory repetitively and the position of the node is a function of time in each repetition. Although this assumption might be quite strong in some situations, it is not unrealistic in many other cases since the motion of most real objects (such as buses [21] and airplanes) have repetitive patterns, and their positions at any particular time can be roughly estimated. Example networks to which such a model can be directly adopted are satellite networks where nodes have accurate periodic connectivity, and underwater acoustic networks where events that trigger the aberrance of the nodes from their fixed trajectory are rare. Later, we will extend this model to cases with uncertainty in contact information.

We define the *motion cycle* T_i of a mobile node n_i having a strict repetitive motion as a period of time such that if the node is at any point p at time t , then it is at p at time $t + k \times T_i$ for any integer k . The *relative motion cycle* $T_{S'}$ of a set of mobile nodes $S' = \{n_1, n_2, \dots, n_k\}$ having strict repetitive motions is a period of time such that if

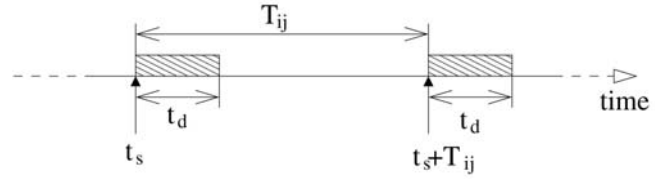


Fig. 3. A predicted contact along the time axis with mobility cycle T_{ij} , contact starting time t_s , and contact duration t_d .

n_1, n_2, \dots, n_k are at points p_1, p_2, \dots, p_k , respectively, at any time t , then they are at p_1, p_2, \dots, p_k , respectively, at time $t + k \times T_{S'}$. The relative motion cycle $T_{S'}$ of a set S' of nodes (including static nodes and nodes with strict repetitive motions) is equal to $T_{S'}$ if $S' \subseteq S$ ($S' \neq \emptyset$) and S' is the set of all mobile nodes in S .

For simplicity, the $T_{S'}$ of $S' = \{n_i, n_j\}$ is denoted by T_{ij} . It is obvious that if T_i and T_j are the motion cycles of n_i and n_j , respectively, then T_{ij} equals the *least common multiple* (LCM) of T_i and T_j . A similar result holds when S' has more than two elements. In Fig. 1, for example, the motion cycles T_1 and T_3 of nodes 1 and 3 are 2 minutes and 3 minutes, respectively, and thus, the relative motion cycle T_{13} of nodes 1 and 3 is $LCM(2, 3) = 6$ minutes. CMNs are common in the real world. The motion cycle of most objects are based on human-defined or natural cycles of time such as hour, day, week, etc., and thus, $T_{S'}$ is usually small.

Once the relative motion cycle T_{ij} of nodes n_i and n_j is known, the set of contacts C occurring within any period of time equal to T_{ij} can be used to represent all the other contacts. In this paper, a predicted contact is represented by a tuple $(n_i, n_j, T_{ij}, t_s, t_d)$, and a persistent contact is represented by $(n_i, n_j, -, -, -)$, where t_s and t_d are the starting time and the duration (which depends on the speed and transmission range) of the contact. Fig. 3 shows a predicted contact along the time axis. For simplicity, we construct a representative set C of contacts for each pair of nodes n_i and n_j within the time period $(0, T_{ij})$. That is, $0 \leq t_s < T_{ij}$. The representative contacts for the nodes in Fig. 1, whose model is shown in Fig. 2a, are given in Fig. 2b. A uniform contact duration of 0.1 second is set to simplify the discussion.

With the representative contacts for a set V of nodes in the network, one can use the optimal time-space Dijkstra algorithm such as the ones in [9] and [13] to calculate the shortest path between a pair of nodes in V . Since our routing is unicast and there are usually multiple paths with the same minimal delay, we design an extended time-space Dijkstra algorithm that first finds all paths with the minimal delay and then chooses one among those paths with the least hop count randomly. This algorithm is used in DHR to route with local contact information in some levels of the hierarchical network. It is also used to compute the optimal routing solution on the global topology that is the basis upon which the routing performance of DHR is evaluated.

3 MULTILEVEL CLUSTERING AND HIERARCHICAL ROUTING

This section reviews a multilevel clustering and hierarchical routing algorithm (which is similar to [3], [5], [14], and [16])

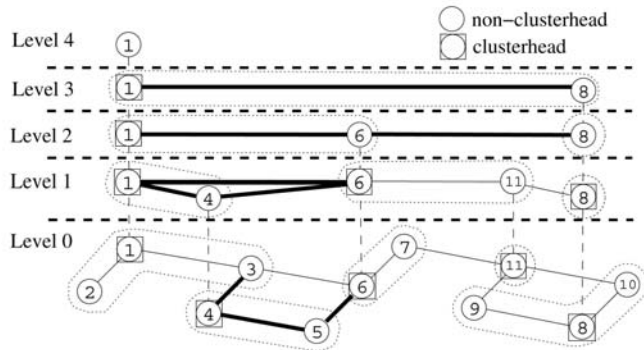


Fig. 4. Lowest ID hierarchical clustering.

in static networks on which our proposed algorithm in Section 3.2 is based. Organizing a network into a hierarchical structure could make the management of routing tables scalable. Clustering offers such a structure, and it suits networks with relatively large numbers of nodes. Multilevel clustering, which is clustering applied recursively over cluster heads, is feasible and effective in large networks.

3.1 Multilevel Clustering

Clustering is conducted first by electing cluster heads. Then, non-cluster heads choose clusters to join and become members. Clustering algorithms can be broadly divided into two types: one in which each cluster head can only be selected by itself (called *clustering algorithm* [4]) and the other in which each cluster head can be selected by itself or one of its neighbors (such an algorithm is called *core algorithm* [17]). In the clustering algorithm, a node selects itself as a cluster head if it has the highest priority among its unclustered neighbors. A non-cluster head joins the cluster of a cluster head that has the highest priority among the node's neighboring cluster heads. In the core algorithm, each node selects a node in its neighborhood including itself with the highest priority as a cluster head and then joins that node's cluster. The main difference between these two is whether cluster heads could be neighbors (as in core) or not (as in cluster). Also, the cluster algorithm runs sequential rounds, while the core algorithm needs one round to complete. In our work, we use the cluster algorithm. There are many ways to define node priorities. The lowest ID algorithm [17] is widely used. In the lowest ID cluster algorithm, the lower a node's ID, the higher its priority is. Fig. 4 shows the hierarchical network as the result of the multilevel clustering using the lowest ID algorithm on the physical network shown on level 0 of the hierarchy.

The hierarchical network is a logical tree of nodes in a homogeneous network, where nodes and links above level 0 in the hierarchy are conceptual (in contrast to the physical nodes and links in level 0). A node in level $k+1$ corresponds to a cluster (or a cluster head) in level k , and the ID of the level- $(k+1)$ node is the same as that of the cluster head in the level- k cluster. We use subscripts to denote the level to which a node belongs. In Fig. 4, we use 4_1 to denote that the node with label 4 is in level 1, which represents the cluster in level 0 consisting of 4_0 and 5_0 . The *hierarchy address* of a node is a sequence of IDs of the node

and its clusters in all levels. For instance, the hierarchy address of the physical node 5 is $(1_4, 1_3, 1_2, 4_1, 5_0)$, where 4_1 is the cluster of 5_0 in level 1, and 1_2 is the cluster of 4_1 (and thus of 5_0) in level 2.

Each cluster head needs to know all of its members and its adjacent cluster heads. If there is a link between any two nodes of two clusters on level k , then there is a link between the nodes representing these clusters in level $k+1$. For instance, in Fig. 4, there is a link between 4_1 and 6_1 since there is a link between 5_0 and 6_0 . We call 5_0 a *gateway* and 6_0 a *remote gateway* from cluster 4_1 to cluster 6_1 . Each level- $(k+1)$ link is associated with a delay that is equal to the delay of the shortest path between the corresponding level- k cluster heads. For instance, if the shortest path between 4_0 and 6_0 is $\{4_0, 5_0, 6_0\}$ and the delay of the link between 4_0 and 5_0 is three and that between 5_0 and 6_0 is two, then the delay of the level-1 link between 4_1 and 6_1 is five.

3.2 Hierarchical Routing

Hierarchical routing facilitates the hierarchical network as a topology abstraction and may not generate a shortest path. The advantage of hierarchical routing is that the routing traffic load is scalable (logarithmic) for localized traffic patterns.

Hierarchical routing requires that the source has the hierarchy address of the destination. If only the ID of the destination is available, the source can resort to using a location service [3]. Hierarchical routing is a hop-by-hop routing. Before any routing, each node in the network needs to obtain the topology information of its clusters in all levels. For instance, in Fig. 4, node 5 has the topology information consisting of all the thick links. Note that it includes the gateway links, such as links $(5_0, 6_0)$ and $(4_1, 6_1)$.

Each node makes its forwarding decision via the following steps:

1. Find the highest level k where the source s and the destination d do not have a common cluster.
2. Define the intermediate source s' and the intermediate destination d' , which are the level- k clusters of s and d , respectively.
3. Use the time-space Dijkstra algorithm to find the next hop n' on the shortest path from s' to d' based on the level- k topology information of s .
4. If $k=0$, n' is the forwarding decision of s ; otherwise, go back to step 3 with a new $k=k-1$, a new d' being the remote gateway from s' to n' , and a new s' being the node on level k (new k), which is either s or a cluster of s .

This routing algorithm releases the cluster heads' burden of relaying every message forwarded to other clusters.

As an example, we show the process of node 5 making its forwarding decision for the destination, node 9, in Fig. 4. The highest level where nodes 5 and 9 do not have a common cluster head is $k=3$, and thus, $s'=1_3$, $d'=8_3$, and the resulting $n'=8_3$. Since $k=3$, we continue with $k=2$, where $s'=1_2$, $d'=6_2$, and the resulting $n'=6_2$. Again, we continue with $k=1$, where $s'=4_1$, $d'=6_1$ (assuming that path $\{4_1, 6_1\}$ has a smaller delay than $\{4_1, 1_1, 6_1\}$), and the resulting $n'=6_1$. Finally, we set $k=0$; then, $s'=5_0$, $d'=6_0$ (assuming path $\{5_0, 6_0\}$ has a smaller delay than

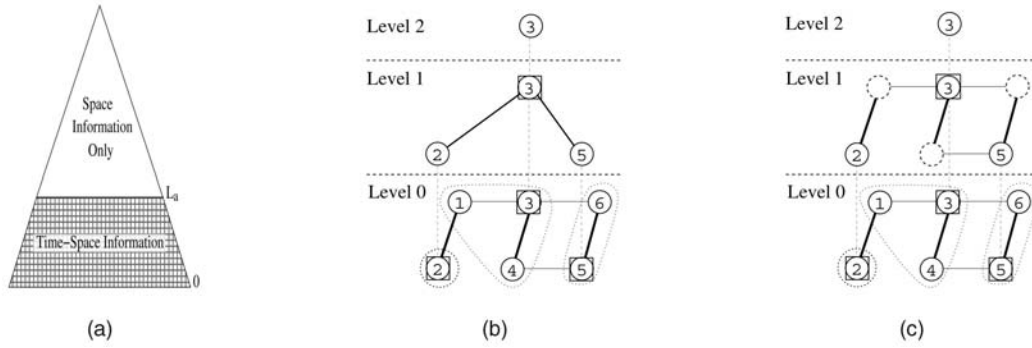


Fig. 5. (a) A hierarchical network with aggregated contact information given an aggregation level. (b) $L_a = 0$. (c) $L_a = 1$.

$\{5_0, 4_0, 3_0, 6_0\}$), and the resulting $n' = 6_0$. Thus, node 5's decision is to forward the message to node 6.

4 MULTILEVEL CLUSTERING AND HIERARCHICAL ROUTING IN DTNS

As mentioned in Section 3, the information in a level- $(k+1)$ link in a static hierarchical network contains a time-invariant delay that is simply the sum of the delays of the level- k links on a shortest path between the level- k cluster heads. The challenge in multilevel clustering in our DTN model is that the delay information in the links is time variant. A method needs to be proposed to aggregate the time-varying information to the links in the hierarchical network from the level that is immediately below them.

4.1 Multilevel Clustering in DTNs

Each node on level 0 of the hierarchical network corresponds to a physical node, and there is a link between two level-0 nodes if there are contacts between their physical nodes. To allow the links in each level in the hierarchical network to have time-variant delay information, we store in each link a set of contacts. Contacts are stored in the five-element tuple format presented in Section 2. A level-0 link stores all contacts between the two end nodes. A level- $(k+1)$ link aggregates all contacts in the level- k links that form paths between the corresponding level- k cluster heads, which includes those between cluster heads and gateways and those between gateways.

For example, in Fig. 5c, nodes 3_0 and 5_0 are cluster heads, and the level-1 link between 3_1 and 5_1 contains the contact information contained in all of the level-0 links between 3_0 and 4_0 , between 5_0 and 6_0 , between 3_0 and 6_0 , and between 4_0 and 5_0 .

The size of the aggregated contact information in the links increases exponentially as their level increases. To ensure scalability, the aggregation should stop at a certain level, which we call the *aggregation level* L_a . The links above level L_a maintain time-invariant delays just as in the static hierarchical network, which is illustrated in Fig. 5a.

Figs. 5b and 5c show the results of two multilevel clusterings on the same network with different values for L_a (0 and 1). One difference is shown in the link between nodes 3_1 and 5_1 . In Fig. 5b, we use a thick line to represent a time-invariant delay in the link since $L_a < 1$. In Fig. 5c, we move the four links from level 0 to level 1 to illustrate that

all related contact information is aggregated to the link from the level immediately below.

The delay on the links above level L_a is calculated in the same way as in the static hierarchical network presented in Section 4. To calculate the time-invariant delay on level $L_a + 1$, each link on level L_a also needs to have a time-invariant delay. We use the *weighted average delay*, which is the expectation (in statistics) of the time-variant delays of the shortest paths between the two end nodes of a level- L_a link. In our DTN model, the weighted average delay can be calculated with the set of shortest paths within a period of time equal to the LCM of the cycles of the contacts stored in the link.

We will now present two examples to show the calculation of the weighted average delay. First, we calculate $D(1_0, 3_0)$ in Fig. 5c, which is a hierarchical network of the DTN in Fig. 1. The link between 1_0 and 3_0 contains only one contact $(1, 3, 6, 0, 0.1)$, as given in Table 2b. Here, we assume that the transmission delay (including queuing delay and radio transmission duration) is 0.01. During the first 0.1 minute when the contact is active, the delay is 0.01 (the transmission delay). In the remaining 5.9 minutes, the delay is $(6 - t + 0.01)$, where $(6 - t)$ is the waiting time for the next contact. Thus

$$D(1_0, 3_0) = \frac{1}{6} \left[\int_0^{0.01} 0.1 dt + \int_{0.1}^6 (6 - t + 0.01) dt \right] \approx 3.$$

Second, we calculate $D(3_1, 5_1)$ in Fig. 5c. The link between 3_1 and 5_1 contains four contacts, and the LCM of their cycles is 3 minutes. In Fig. 1, we see that in the first 3 minutes, before the occurrence of contact between nodes 3 and 6, path $\{3, 4, 5\}$ has the minimal delay $\max(3 - t + 0.01, 0.02)$; during the contact (0.1 minute, except the last 0.01 minute), path $\{3, 6, 5\}$ has the minimal delay of 0.02. Therefore

$$D(3_1, 5_1) = \frac{1}{3} \left[\int_0^{0.09} 0.02 dt + \int_{0.09}^{3.0} \max(3 - t + 0.01, 0.02) dt \right] \approx 1.5.$$

4.2 Cluster Head Selection in DTNs

The selection of each cluster head becomes more important here. In general, a cluster head should be situated at the

TABLE 1
Data in the Example Hierarchical Clustering of Fig. 5c

$D(i, j)$	$i = 1_0$	$i = 2_0$	$i = 3_0$	$i = 4_0$	$i = 5_0$	$i = 6_0$
$j = 1_0$	-	0.1	3	-	-	-
$j = 2_0$	0.1	-	-	-	-	-
$j = 3_0$	3	-	-	0.1	-	1.5
$j = 4_0$	-	-	0.1	-	1.5	-
$j = 5_0$	-	-	-	1.5	-	0.1
$j = 6_0$	-	-	1.5	-	0.1	-

(a)

Level 0 Priority	Absolute $Pr(n)$	Relative $Pr_j(n)$					
		$j = 1_0$	$j = 2_0$	$j = 3_0$	$j = 4_0$	$j = 5_0$	$j = 6_0$
$n = 1_0$	1	-	10	3.33	-	-	-
$n = 2_0$	0	0	-	-	-	-	-
$n = 3_0$	3	1	-	-	30	-	2
$n = 4_0$	1	-	-	10	-	0.67	-
$n = 5_0$	1	-	-	-	0.67	-	10
$n = 6_0$	1	-	-	0.67	-	10	-

(b)

$D(i, j)$	$i = 2_1$	$i = 3_1$	$i = 5_1$
$j = 2_1$	-	3.1	-
$j = 3_1$	3	-	1.5
$j = 5_1$	-	1.5	-

(c)

Level 1 Priority	Absolute $Pr(n)$	Relative $Pr_j(n)$		
		$j = 2_1$	$j = 3_1$	$j = 5_1$
$n = 2_1$	0	-	0	-
$n = 3_1$	1	0.33	-	0.67
$n = 5_1$	0	-	0	-

(d)

(a) Level-0 average delay. (b) Level-0 node priority. (c) Level-1 average delay. (d) Level-1 node priority.

“center” of the cluster to provide a better network topology approximation in terms of average delay. The cluster head selection is reflected in the calculation of the node priorities in the multilevel clustering, as shown below.

The multilevel clustering algorithm we use in our DTN model uses two priorities for each node: the *absolute priority* $Pr(n)$ and the *relative priority* $Pr_i(n)$:

$$Pr(n) = \sum_{i,j \in N(n), i \neq j} \frac{\frac{1}{D(i,n)+D(n,j)}}{\frac{1}{D(i,j)} + \frac{1}{D(i,n)+D(n,j)}}$$

Here, $N(n)$ is the set of the neighbors of node n , and $D(i, j)$ is the weighted average delay between nodes i and j . Note that each term inside the summation has a value in $(0, \frac{1}{2}]$ (since $D(i, j) \leq D(i, n) + D(n, j)$). The criteria for cluster

TABLE 2
Contacts in the DTN Model of Fig. 6d

Node	Contacts $(n_i, n_j, T_{ij}, t_s, t_d)$
1	(1, 4, 6, 0, 0.1), (1, 5, 6, 5, 0.1), (1, 7, 6, 2, 0.9)
2	(2, 4, 6, 4, 0.1), (2, 5, 6, 3, 0.1), (2, 7, 6, 0, 0.9)
3	(3, 7, -, -, -)
4	(4, 6, 2, 0.3, 0.1), (4, 1, 6, 0, 0.1), (4, 2, 6, 4, 0.1)
5	(5, 6, -, -, -), (5, 1, 6, 5, 0.1), (5, 2, 6, 3, 0.1)
6	(6, 4, 2, 0.3, 0.1), (6, 5, -, -, -), (6, 8, -, -, -)
7	(7, 3, -, -, -), (7, 1, 6, 2, 0.9), (7, 2, 6, 0, 0.9)
8	(8, 6, -, -, -)

head selection reflected by the absolute priority are that 1) a cluster head is on or close to the shortest path between any two neighbors (which is reflected by the value of the term inside the summation), and 2) a cluster head has a large node degree (which is reflected by the summation):

$$Pr_i(n) = \frac{Pr(n)}{D(i, n)}$$

Our rationale is that a cluster head should 1) have a large betweenness centrality (the number of shortest paths between the other nodes on which the current node is located) and 2) be popular (having many contacts).

The clustering algorithm is simply presented as follows, which is executed whenever the topology of a node’s 1-hop neighbors changes: 1) Each node calculates its absolute priority, 2) a node without a neighboring cluster head declares itself a cluster head if it has the largest absolute priority among its neighbors that have not joined any cluster, and 3) a node having neighboring cluster heads chooses one with the largest relative priority and joins that cluster.

We show the priorities of the nodes and the weighed average delays of the links used in the multilevel clustering of Fig. 5c in Table 1.

4.3 Contact Information Compression

In this section, we address the challenge of efficient information aggregation and compression in the time-space domain while maintaining critical information for efficient routing. Whenever the multilevel clustering is completed,

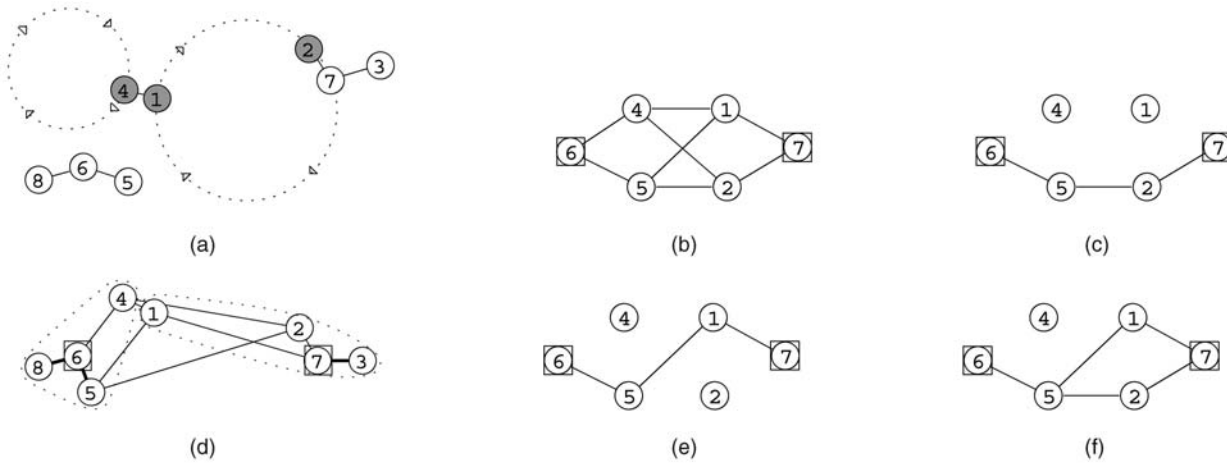


Fig. 6. Another physical network at time 0 (a) and its model (d). The aggregated contact information compression process (b), (c), (e), and (f) on the link between 6_1 and 7_1 . (a) Physical network. (b) All contacts. (c) Path 1. (d) DTN model. (e) Path 2. (f) Result.

we apply two methods to compress the aggregated contact information in the hierarchical links. The first method focuses on compression in the space domain, while the second one addresses compression in the time domain.

The first compression method removes part of the contact information from the links. For a hierarchy link on level L ($0 \leq L \leq L_a$), a contact is meaningful to the calculation of the underlying shortest path of the link only if it is on a shortest path at a particular time. This method is illustrated by the following example. The physical network in this example is shown in Fig. 6a, its model is shown in Fig. 6d, and the contacts in the network model are shown in Table 2. Node 4 has a motion cycle of 2 minutes, and the motion cycles of nodes 1 and 2 are both 6 minutes.

The hierarchical clustering process runs on the network and the result of the level-0 cluster is shown in Fig. 6d. As rearranged in Fig. 6b, the level-0 cluster heads are 6_0 and 7_0 , 4_0 and 5_0 are the gateways from 6_1 to 7_1 , and 1_0 and 2_0 are the gateways from 7_1 to 6_1 . Contacts on all of the possible paths between 6_0 and 7_0 through these gateways are aggregated to the level-1 link between 6_1 and 7_1 (Fig. 6b).

Since all the predicted contacts shown in Table 2 have a common period of 6 minutes, we will examine the paths from 6 to 7 with minimal delays in the first 6 minutes. It can be observed in Fig. 6a and Table 2 that the path with the minimal delay between minute 3 and minute 5 is $\{6, 5, 2, 7\}$, as shown in Fig. 6c, and the path with the minimal delay in the rest of the time is $\{6, 5, 1, 7\}$, as shown in Fig. 6e. After removing all the contacts that are not on these two paths, the result of the compression is shown in Fig. 6f.

The first compression method is implemented as follows: Suppose (u, v) is a level- k link, T is the LCM of the motion cycles of the contacts stored by (u, v) , $\{t_i\}$ is the set of time slots that is a partition of T such that the set of shortest paths starts at any time within t_i and, between u and v , does not change, and $\{S_{ij}\}$ is the set of contacts on the j th shortest path starting within t_i . A Boolean function E is constructed in the following steps. First, construct terms E_{ij} with the names of all contacts in $\{S_{ij}\}$ as literals. Second, construct disjunctive normal forms (a disjunction of terms) E_i with all E_{ij} s. Finally, construct E , which is a conjunction of all E_i s. Here, a set of

contacts satisfies E_i if they form a shortest path starting within t_i , and a set R of contacts satisfying E can be used to calculate the delay of the shortest paths between u and v at any time. This compression algorithm reduces the size of R to reduce the storage and communication overhead while retaining most of the important contacts for the links. This compression is not lossless since a contact in a shortest path between two cluster heads on level $k+1$ might not be on any shortest path between cluster heads on level k .

An intelligent improvement is to choose the smallest set of contacts whose shortest paths' variance from the actual shortest paths is below a particular threshold. It should be able to further reduce the size of contact information when, for instance, 20 percent of the contacts form 80 percent of the shortest paths.

The second method, which has already been presented in the previous section, compresses the time-space information by removing the time-dimension information in the links above level L_a . The analytical study and simulation in later sections show that its impact on the routing performance is slight when L_a is moderate. An intelligent improvement is to decide whether to compress according to the volume of the contact information and the variance (in statistics) of the time-variant delays of the link instead of based on a constant L_a .

4.4 DTN Hierarchical Routing (DHR)

Our proposed DHR is quite straightforward after the hierarchical network has been built. DHR is also a hop-by-hop routing. Each node makes its forwarding decision in two phases. The first phase runs only when the highest level L_c , on which the cluster of the current node and that of the destination are different, is greater than L_a . In this phase, the static hierarchical routing, as presented in Section 3.2, runs on the levels (from L_c to L_a+1) where delay on the links are time invariant. The result of this phase is an intermediate destination d' (see Section 3.2) of level L_a .

When $L_c \leq L_a$, the first phase does not run, and d' is set to the cluster head of the destination in level L_c . The second phase, which we called *aggregation routing*, has the following steps: 1) All contact information in the links from level 0 to level L_c that are stored by the current node

are aggregated to form a graph G . That is, G includes all contact information for the links in the clusters of the current node from level 0 to level L_c in the hierarchical network. 2) The optimal time-space Dijkstra algorithm is then performed on G to find a shortest path p from the current node to d' . The first hop on p is the current node's forwarding decision.

5 DHR WITH PROBABILISTIC CONTACTS

In the previous sections, we have assumed the simplified DTN model for cyclic mobile space where node trajectories can be precisely described by functions from time to position. In practical situations, a routing protocol should be able to tolerate a moderate level of uncertainty in the cyclical trajectories of the nodes. In DTNs such as bus and airplane networks, the nonpreciseness in node trajectories might add uncertainty regarding the occurrence of the predicted contacts and consequently degrade the routing performance of DHR, as well as other deterministic routing algorithms. In this section, we use probabilistic contacts to model the uncertainty of contact information. We also extend DHR under probabilistic contact information to enhance its routing performance in CMNs with uncertainty.

5.1 Probabilistic Contacts

The uncertainty in the node trajectories with repetitive mobility may be caused by many factors in practical situations. Take a bus for example: the time a bus stops at a bus station might change depending on the number of passengers, and the speed of the bus would be affected by the traffic conditions. The uncertainty can be represented by the probability of the actual occurrence of the predicted contact. We use an empirical method to estimate the probability of occurrence p_o of a predicted contact by a function of the duration t_d of the contact as the following:

$$p_o = 1 - e^{-\lambda t_d}, \quad (1)$$

where λ is called the contact uncertainty parameter, which is a positive value representing the level of uncertainty in the contact information for a network. A network with a larger level of uncertainty in its contact information has a smaller λ .

To estimate the relation between the uncertainty of contact information and the duration of the contact, we conduct the following simulation. We create some turbulence on the nodes' mobilities at a certain time. In a discrete-event simulator, where a node's trajectory is a series of waypoints, we create turbulence by moving each node forward or backward a few waypoints (16-64 in our simulation). We generate networks with 70 mobile nodes and 30 static nodes using the method found in Section 7. We run the simulation for a period of time (24 seconds) such that all contact information is collected by the nodes. We then store the contact information as set S_1 , remove the contact information from the nodes, create a turbulence for each node, run the simulation for another 24 seconds, and store the contact information as another set S_2 . For each contact c_1 in S_1 , if there is a contact c_2 in S_2 such that the two nodes in the contact are the same for c_1 and c_2 and c_1 and c_2

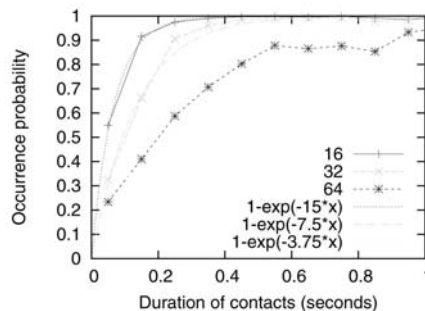


Fig. 7. Estimating uncertainty in contact information.

overlaps in time, we regard that contact c_1 reoccurs after the turbulence of the nodes' mobilities.

We perform this simulation in 30 different randomly generated networks (see Section 7). The certainty of a contact information is the probability of the contact's reoccurrence after the turbulence made in the node mobilities. The turbulences of the nodes are uniformly distributed with the maximum numbers of waypoints ranging from 16 to 64, respectively, in different rounds of simulations. The total average number of waypoints on the trajectories of the nodes is 500. The result is shown in Fig. 7, where the uncertainty (one minus the occurrence probabilities of the contact information) is plotted in four curves for different maximum numbers of waypoints in the turbulence. Each curve shows the certainties under different contact durations. Estimation functions with different λ s are also shown in the same figure. We found that our estimation functions with different λ values 15, 7.5, and 3.75 approximates the curves best, respectively.

5.2 Estimating Delay with Uncertainty

We need to estimate the delay of a path that consists of probabilistic contacts. This estimation method will be used in the calculation of the time-variant delay in the multilevel clustering and the routing process that use probabilistic contact information. We assume that the probability of occurrence of each contact is estimated by (1). In practice, given a network that has run for a substantial period of time, the probability of each contact can be calculated from the statistics of the historical contacts. Note that we do not intend to provide a very accurate estimation of the expected delay but rather to provide a simple solution¹ to rank the available paths for each node to get the preferential next hop for each message.

The estimated waiting time $D_p(c, t)$ for a reoccurring contact c at time t is given as follows: First, let $t_s(c, t)$ be the first occurrence of c after time t . We have

$$t_s(c, t) = \min_{k=0}^{\infty} \{t_s(c) + T(c) \times k | t_s(c) + T(c) \times k > t\},$$

where $T(c)$ and $t_s(c) = t_s(c, 0)$ are the cyclic period and the starting time of contact c , respectively. Letting D be estimated waiting time for c immediately before time $t_s(c, t)$, we have

1. Better but more complicated solutions can be devised similar to the solutions to the Expected Minimum Delay (EMD) problem.

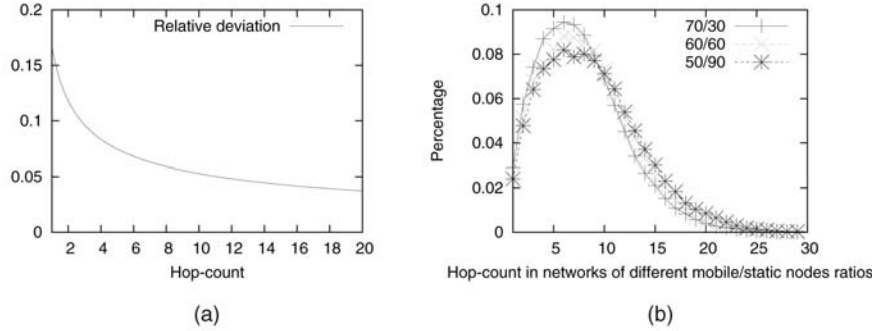


Fig. 8. (a) The inaccuracy of the time-invariant delay represented by the relative deviation $\frac{\sigma_n}{\mu_n}$. (b) Percentage of routes of different lengths.

$$\begin{aligned} D &= p_o(c) \times 0 + (1 - p_o(c)) \times (T(c) + D) \Rightarrow D \\ &= \frac{1 - p_o(c)}{p_o(c)} T(c). \end{aligned}$$

Therefore, the estimated waiting time is calculated by

$$D_p(c, t) = (c.t_s(t) - t) + D = (t_s(c, t) - t) + \frac{1 - p_o(c)}{p_o(c)} T(c).$$

At time t , the estimated delay $D'_p(p_n, t)$ of a path $p_n = c_1, c_2, \dots, c_n$, which consists of contacts c_1, c_2, \dots, c_n , is estimated by

$$D'_p(p_n, t) = D'_p(p_{n-1}, t) + D_p(c_n, D'(p_{n-1}, t)), \quad (2)$$

where $D'(p_{n-1}, t)$ is the estimated starting time of c_{n-1} given that the message is in the first node at time t . Assuming that $p_o(c_i) = 1$ for $1 \leq i \leq n$, $D'(p_{n-1}, t)$ can be approximated by

$$D'(p_n, t) = D'(p_{n-1}, t) + t_s(c_n, D'(p_{n-1}, t)).$$

We use $D_1(p_{n-1}, t)$ instead of $D'_p(p_{n-1}, t)$ in (2) to calculate the estimated waiting time of c_n because 1) the latter is just an estimated waiting time and not the actual waiting time of the previous $n - 1$ contacts and therefore is not appropriate as the starting time of the n th contact, and 2) the estimation given by (2) can be quite accurate if n is not large and $p_o(c_i)$ s are large enough (which is likely since the best paths will prefer larger cumulative contact probabilities, which are likely to have large $p_o(c_i)$ s).

5.3 Multilevel Clustering and Routing with Probabilistic Contacts

Multilevel clustering with probabilistic contact information is basically the same as multilevel clustering with accurate contact information except that the weighted average delay is replaced by the estimated delay calculated as described in the last section, which effectively excludes the relatively unreliable contacts from the hierarchical network.

Routing performance is expected to degrade as the level of uncertainty increases. The reasons are the following: 1) effectively fewer contacts are available as $p_o(c_i)$ decreases, 2) historical contact information is less reliable as $p_o(c_i)$ decreases, and 3) the estimated delay from (2) is less accurate as $p_o(c_i)$ decreases. Fortunately, our simulation results in Section 7 show that DHR still has satisfactory routing performance with moderate degree of uncertainty in contact information.

6 ANALYSIS

6.1 Routing Performance

In this section, we will analyze the possible impact of removing the time dimension at higher levels (above L_a). We estimate the inaccuracy of using the weighted average delay to represent the delay of a time-variant link with the *relative deviation* $\frac{\sigma_n}{\mu_n}$, where μ_n is the expectation of the delay of an n -hop link (i.e., the weighted average delay), and σ_n is the standard deviation of the delays of the shortest paths.

For simplicity, we only calculate $\frac{\sigma_n}{\mu_n}$ in a simplified setting where the waiting time for a contact in each hop is in an independent identical uniform distribution. Thus, $\mu_n = n \times \mu_1$, and $\sigma_n^2 = n \times \sigma_1^2$, where μ_1 and σ_1 are the expectation and the variance of the 1-hop delay. We have $\frac{\sigma_n}{\mu_n} = \frac{\sigma_1}{\mu_1} / \sqrt{n}$, which is plotted in Fig. 8a.

Since the hierarchical network provides partial time-variant shortest path information when $L_a > 0$, its performance should be lower bounded by the results in Fig. 8a.

6.2 Overhead

This section briefly analyzes the overhead in DHR, i.e., the overhead in maintaining the hierarchy and the overhead in the routing process.

The contact information of each node is sent to its cluster head, which aggregates it to form a cluster topology and broadcasts the topology to the nodes in the cluster. In any broadcast algorithm, each node sends the message at most once, so the communication overhead will not exceed that of the storage overhead. Under level L_a (a constant), where the contact information is aggregated, the worst case contact information is $O(C^{L_a})$, where C is the maximal cluster size. For each cluster above L_a , where links are time invariant, the cluster topology information size is $O(C)$. The number of levels in a hierarchical network is $O(\log(N))$, where N is the network size. Thus, the average communication and storage overhead for maintaining the hierarchical network is $O(\log(N) \cdot C^{L_a})$. The diameter of the network is $O(\sqrt{N})$. The routing overhead is on the average $O(\sqrt{N})$.

Compared with the $O(N)$ communication overhead of Dijkstra's algorithm for a CMN, DHR is more scalable.

7 SIMULATION AND RESULTS

We developed a stand-alone discrete-event simulator to evaluate our protocol. This simulator only implements the network layers, and it makes simple assumptions regarding

TABLE 3
Simulation Parameters

Parameter	Value
Field size	$5,000 \times 5,000(m^2)$
Transmission range	150(m)
Transmission delay	100(ms)
Number of static nodes	30/60/90
Number of mobile nodes	70/60/50
Total number of nodes	100/120/140
Percentage of mobile nodes	70/50/35
Region size	600/800/1200(m)
Motion cycle	675/900/1350(s)
Aggregation level (L_a)	0-2
Contact uncertainty parameter (λ)	15/7.5/3.75

lower layers. For instance, it assumes infinite bandwidth and nodes having infinite buffers. We compare the performance of DHR against the optimal time-space Dijkstra algorithm in terms of the delay and hop count of the resulting paths.

7.1 Simulation Settings

In our simulation, we generate connected (in the DTN sense) networks containing both mobile nodes and static nodes in a $5,000 \times 5,000 m^2$ field. We generate three groups of 30 networks with different ratios of mobile nodes and static nodes, as shown in Table 3. For instance, the first group contains 70 mobile nodes and 30 static nodes (mobile nodes account for 70 percent of the total nodes). All nodes have a uniform transmission range of 150 m. An example of the simulation network is shown in Fig. 9, where the trajectories of the mobile nodes are shown by dotted lines.

The trajectory of a mobile node is generated as follows: 1) a square region of a given size is placed at a random position in the network, 2) three to seven points are sprayed at random positions inside the region, and 3) a trajectory is formed by starting from the first point, traveling to each point once, and finally coming back to the first point. Here, we use the nearest neighbor algorithm in the traveling salesman problem to generate a short trajectory. The length of the sides for the square regions is 600 minutes, 800 minutes, or 1,200 minutes. The nodes' motion cycles corresponding to these square regions are 675 seconds, 900 seconds, and 1,350 seconds, respectively. Thus, the relative motion cycle of the network is 2,700 seconds. The placement of the static nodes in the networks is random. Table 3 shows the critical simulation settings. We only use time-space connected networks in our simulations.

For a source-destination pair, we refer to the routes resulting from the optimal time-space Dijkstra algorithm as *Dijkstra routes* and those of the DHR as *DHR routes*. In different experiments, we let L_a range from 0 to 2. After a hierarchical network is built, we route messages from every node in the network to 30 other randomly selected nodes. The delay and hop count for all the source-destination pairs are then averaged and grouped by the hop count of their Dijkstra route. The results are also averaged over the 30 networks for each group with the same mobile/static node ratio.

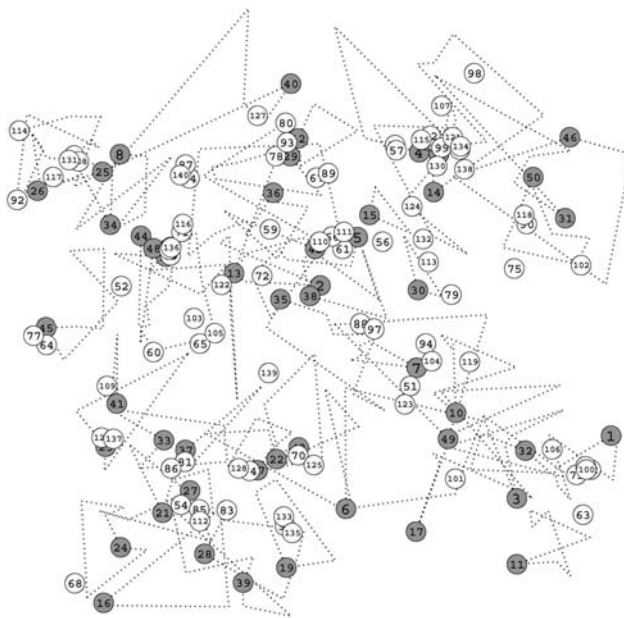


Fig. 9. An example of the simulated network.

7.2 Results in the Simplified DTN Model

First, we show the results of DHR routing using only the compression that removes contacts above level L_a . Fig. 8b shows the distribution of the hop count of Dijkstra routes in the three network groups. Thus, we only show the simulation results where the hop count is between 1 and 13 hops because only the routing performance within this range is accurate (with a large volume of data) and representative (with more than 90 percent of the total paths).

Figs. 10a, 10b, and 10c show the lengths of DHR routes in terms of hop count. The x -axis is the length of Dijkstra routes of the same source-destination pairs. DHR routes with different L_a s are all very close to Dijkstra routes in hop count, and a larger L_a gives a smaller hop count. Note that Dijkstra routes are not always shorter than DHR routes since they are first optimized in delay and then in hop count.

Figs. 10d, 10e, and 10f present the same data as Figs. 10a, 10b, and 10c from a different angle where the hop counts of the DHR routes are divided by those of Dijkstra routes. The hop count of the DHR routes becomes smaller than that of Dijkstra routes when the hop count of Dijkstra routes is 13.

Figs. 11a, 11b, and 11c compare the DHR routes with Dijkstra routes in terms of delay. DHR always performs worse than the optimal time-space Dijkstra algorithm, and DHR with a larger L_a performs better. The figure shows that the DHR routes of the three groups of networks (different in static/mobile node ratio) have similar average delays.

Figs. 11d, 11e, and 11f show the delay ratio of the DHR routes to Dijkstra routes. The ratio is greater than and closer to 1.0 as the hop count increases. DHR paths have less than 20 percent additional delay than Dijkstra paths in most cases when $L_a = 1$ or $L_a = 2$, while they have more than 40 percent additional delay in all cases when $L_a = 0$. The performance of DHR improves as the source-destination distance increases, which is consistent with our previous analytical results, shown in Fig. 8a.

We also conduct simulations by applying the CIR method. Due to space limitation, only the delay ratios are

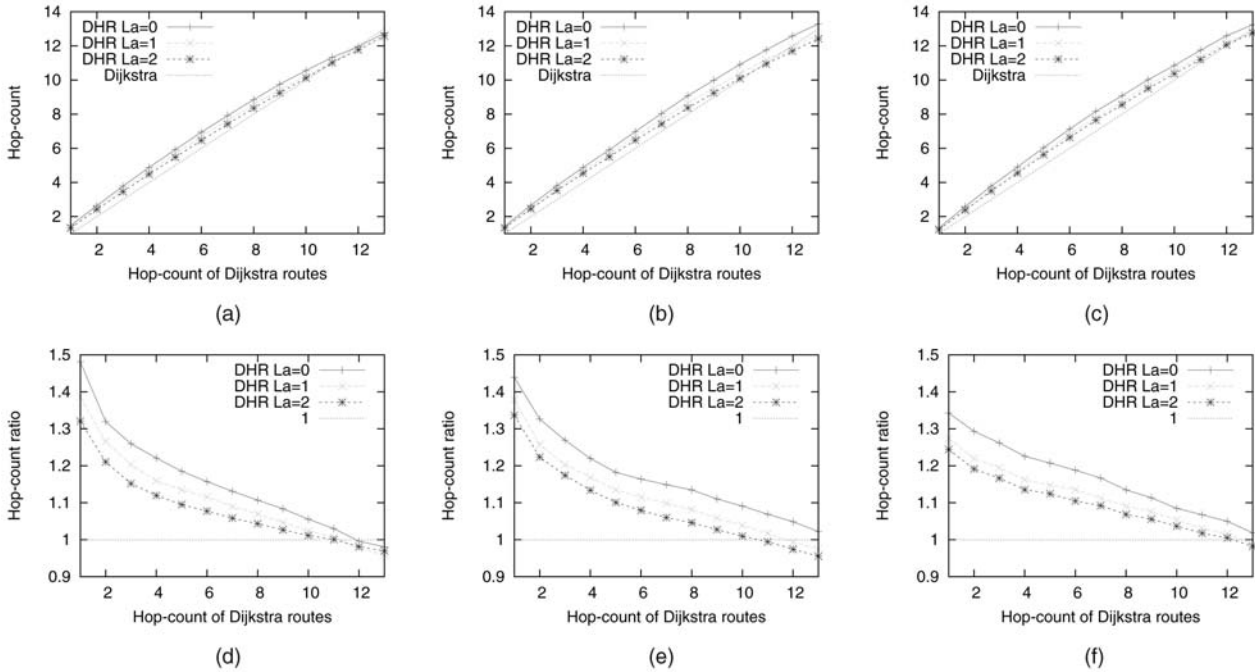


Fig. 10. The hop count and the ratio of the hop count of DHR (with different L_a s) routes compared to those of Dijkstra routes in networks of different numbers of mobile nodes and static nodes. (a) 70 mobile and 30 static nodes. (b) 60 mobile and 60 static nodes. (c) 50 mobile and 90 static nodes. (d) 70 mobile and 30 static nodes. (e) 60 mobile and 60 static nodes. (f) 50 mobile and 90 static nodes.

shown. Fig. 12 shows the delay ratio of DHR with and without CIR. Fig. 13 shows the per-node storage overhead in the hierarchical network. The overhead is calculated in terms of the ratio of the number of contacts stored in each node to the total number of contacts in the network. In Section 6, we discussed that the communication overhead is proportional to the storage overhead.

Fig. 13 shows that the per-node contact information is only 1 percent-3 percent of the total information size in the network when using CIR, which is a substantial saving when compared to the 10 percent of the total size when CIR is not used. This provides a trade-off between scalability and routing performance.

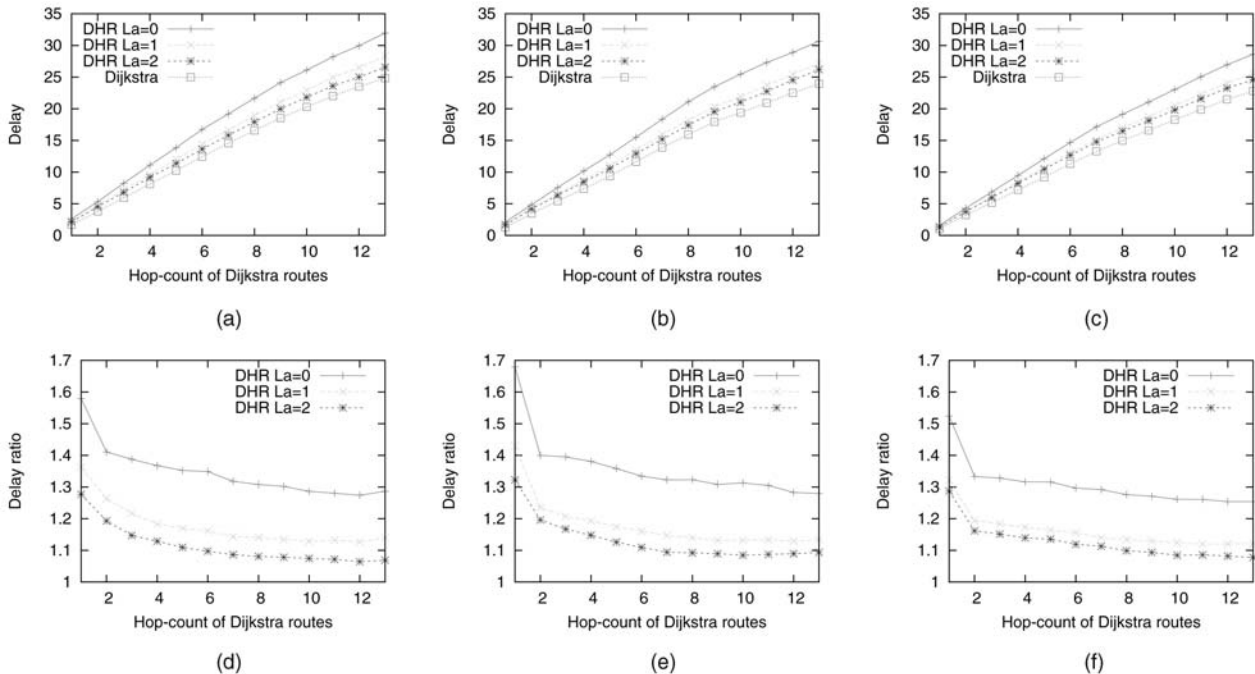


Fig. 11. The delay and the ratio of the delay of the DHR (with different L_a s) routes compared to those of Dijkstra routes in networks of different numbers of mobile nodes and static nodes. (a) 70 mobile and 30 static nodes. (b) 60 mobile and 60 static nodes. (c) 50 mobile and 90 static nodes. (d) 70 mobile and 30 static nodes. (e) 60 mobile and 60 static nodes. (f) 50 mobile and 90 static nodes.

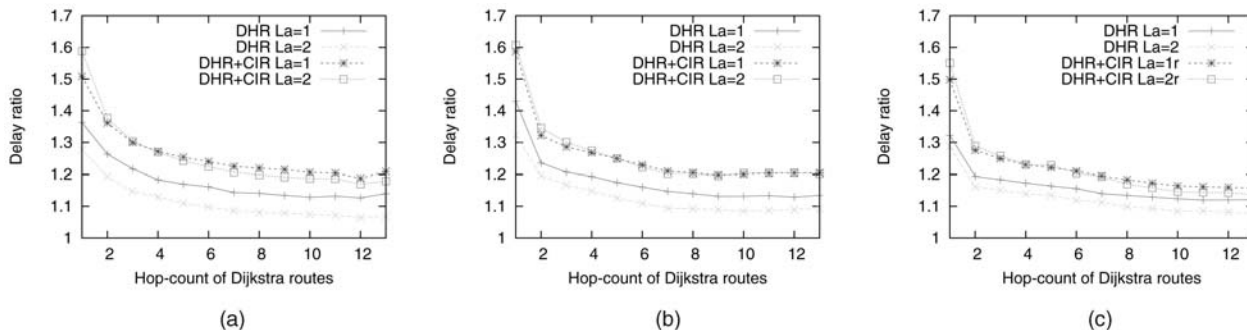


Fig. 12. The ratio of the delay of the DHR (with and without CIR) routes to that of Dijkstra routes in networks of different numbers of mobile nodes and static nodes. (a) 70 mobile and 30 static nodes. (b) 60 mobile and 60 static nodes. (c) 50 mobile and 90 static nodes.

7.3 Results in the DTN Model with Uncertainty

We use the DTN model with uncertain contact information as presented in Section 5, where each contact information c is associated with a probability of occurrence $p_o(c)$. For each possible occurrence of a contact, we determine whether the contact occurs randomly with the probability being $p_o(c)$. Once determined, we store this information in a map with the key being $c.n_i$, $c.n_j$, and $c.t_s$. This makes the actual forwarding opportunity consistent for the routing process of all messages. All of the networks that are simulated in this section feature 70 mobile nodes and 30 static nodes. The multilevel clustering is performed with $L_a = 2$ without CIR.

We use three contact probability parameters ($\lambda = 15, 7.5, 3.75$, respectively) in the estimation of the probability of occurrence $p_o(c)$ with (2). Note that a smaller λ means a larger level of uncertainty. We use a flooding-based routing algorithm to compare the performance of DHR and DHR+. The routing performance in terms of delay will be the same as the previously used time-space Dijkstra algorithm.

Figs. 14a, 14b, and 14c show the routing performance of DHR and DHR+ under different levels of uncertainty in contact information. As shown in the figures, the routing performance in terms of delay of all protocols degrades as λ increases. Also, the delay increases almost linearly as the source-destination distance increases. The routing performance of DHR+ is better than that of DHR because the latter considers the uncertainty of the contact information.

Figs. 14d, 14e, and 14f show the same results as Figs. 14a, 14b, and 14c, but they are shown in terms of the ratio of the delays of DHR and DHR+ over that of Flooding. The comparison is more clear from these figures: DHR has a significant increment over DRH (up to 25 percent when

$\lambda = 15$ and up to 40 percent when $\lambda = 7.5$). However, when the level of uncertainty in contact information is high, DHR+ does not show a significant improvement over DHR for the estimated delay, as (2) becomes inaccurate when $p_o(c_i)$ s are small and n is large, as we discussed in Section 5. Note that the delay ratio does not decrease as the source-destination distance increases, as DHR does in the DTN model with accurate contact information. This is probably due to the weakening in the delay estimation as n increases. Overall, the simulation results show a very satisfactory delay ratio between DHR/DHR+ and the optimal result from the Flooding algorithm.

7.4 Summary of Simulation

To sum up, the simulation results show that our scalable algorithm, DHR, has satisfactory performance that is comparable to the optimal result from the optimal time-space Dijkstra algorithm or the Flooding algorithm for networks combined with mobile nodes and static nodes in different ratios and networks in different levels of uncertain contact information, respectively. DHR with a larger aggregation level and less contact information compressed performs better than that with a smaller aggregation level in terms of both hop count and delay. Routing performance improves as the source-destination distance in terms of hop count increases, which is consistent with our theoretical analysis, as shown in Fig. 8a. While having good routing performance, the storage and communication overhead of running DHR is far smaller than the overhead required to run the optimal time-space Dijkstra algorithm. DHR maintains good routing performance when the level of uncertainty in contact information is moderate, and the extension of DHR has better performance than DHR in networks with probabilistic contacts.

8 RELATED WORKS

In [9], Jain et al. exhaustively formulated the DTN routing problem under different degrees of knowledge about the network. Specifically, Dijkstra's algorithm is used to calculate the shortest path with contacts being predictable, as well as a linear programming approach that calculates the optimal routes across the network with additional knowledge of the global traffic demand. In [13], Merugu et al. proposed a time-space routing algorithm that is similar in spirit to Dijkstra's algorithm in [9]. In [15], Ramanathan et al.

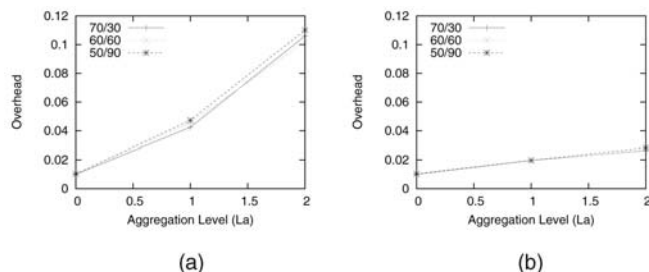


Fig. 13. Storage/communication overhead with/without CIR. (a) Without CIR. (b) With redundant CIR.

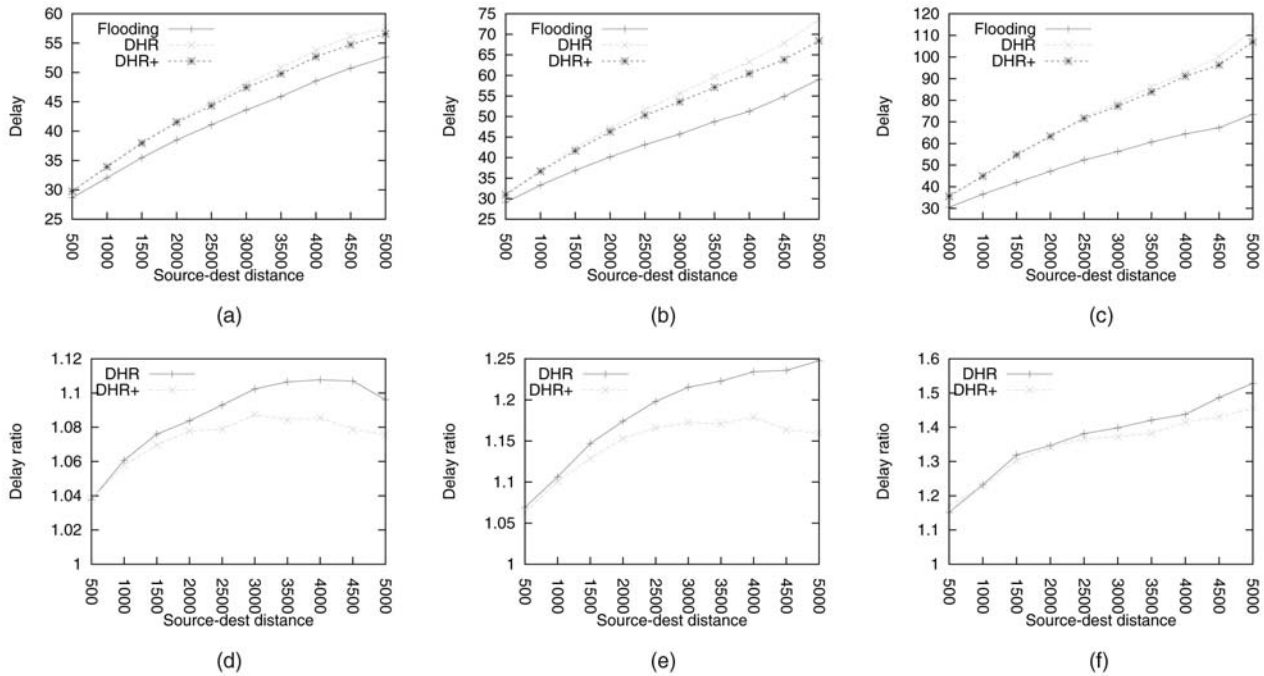


Fig. 14. The delay and ratio of DHR route and DHR+ route compared with that of Flooding in DTNs with different contact uncertainty levels λ . (a) $\lambda = 15$. (b) $\lambda = 7.5$. (c) $\lambda = 3.75$. (d) $\lambda = 15$. (e) $\lambda = 7.5$. (f) $\lambda = 3.75$.

present several routing algorithms in time-space graphs in terms of connectivity and routing capability.

Among the approaches in deterministic routing, Tariq et al. in [18] and Zhao et al. in [22] exploited the deterministic trajectory of mobile nodes to help deliver data, improve data delivery performance, and reduce energy consumption in nodes. In [20], Wu et al. used the semideterministic trajectory of mobile nodes to achieve deterministic results of several routing schemes. However, such a trajectory is selected from a set of predefined hierarchical structured routes.

Epidemic routing by Vahdat and Becker [19] is a random movement and flooding-based algorithm, where all nodes are mobile and have infinite buffers. When a node has a message to send, it propagates the message to all nodes through contacts. Epidemic routing is extended by Haas et al. in Gossip [7], where each message is flooded to the neighbor nodes with probability p .

In the probability-based routing by Lindgren et al. in [11], each node maintains the delivery predictability (based on historical contacts) to all known destinations and uses them to make routing decisions. In [10], Leguay et al. presented a routing scheme for DTNs that uses a high-dimensional euclidean space constructed upon the nodes' motion patterns. The axis in the euclidean space could be contacts or particular locations in the network. This algorithm might not deliver messages in some situations and requires global knowledge such as the locations of the access points. Similarly, in Solar [6], Ghosh et al. used a set of hubs for probability-based routing in semideterministic mobility modeling of DTNs. In [8], Hui et al. identify the logical topology using statistical information, i.e., two specific aspects of society (community and centrality), for

weighting edges, and then, such logical topology is used for the aid of routing.

Much work has been done on multilevel clustering and hierarchical routing in MANETs. In multilevel clustering approaches such as DART [5], L+ [3], MMWN [16], and WHIRL [14], certain nodes are elected as cluster heads. These cluster heads in turn select higher level cluster heads, up to some desired level. A node's address is defined as a sequence of cluster head identifiers, one per level, allowing the size of routing tables to be logarithmic in the size of the network. One problem with explicit cluster heads is that routing through cluster heads creates traffic bottlenecks. In L+, this issue is partially solved by allowing nearby nodes to route packets instead of the cluster head.

9 CONCLUSION

In this paper, we have proposed DHR in a simplified DTN model where nodes have strict repetitive motions. We constructed a hierarchical network that provides a compressed time-space topology abstraction of the network in our DTN model. Two aggregated contact information compression algorithms are presented for better scalability. Simulation results showed that the performance of DHR approximates that of the optimal time-space Dijkstra algorithm in terms of delay and hop count in networks of different ratios of mobile and static nodes. Simulation results are consistent with the theoretical analysis in that the performance of DHR increases as the source-destination distance increases, and the routing performance degrades slightly as more contact information in the hierarchical network is compressed. Our future work will focus on improving the contact information compression algorithms. Two such improvements have been suggested in Section 4.3. The expected shortest path (ESP) problem has been solved

for probabilistic graphs where constant delays are associated with the links. In our future work, we will also extend the ESP to probabilistic graphs whose links are represented by time-variant contact information, and we expect the routing performance of DHR to improve further when we have a more accurate estimation for the delay of the shortest paths.

ACKNOWLEDGMENTS

This work was supported in part by US National Science Foundation (NSF) Grants CNS 0422762, CNS 0434533, CNS 0531410, and CNS 0626240.

REFERENCES

- [1] U. Acer, S. Kalyanaraman, and A. Abouzeid, "Weak State Routing for Large Scale Dynamic Networks," *Proc. ACM MobiCom*, 2007.
- [2] V. Cerf, S. Burleigh, A. Hooke, L. Torgerson, R. Durst, K. Scott, K. Fall, and H. Weiss, *Delay-Tolerant Network Architecture*, Internet Draft: draft-irrf-dtnrg-arch.txt, DTN Research Group, 2006.
- [3] B. Chen and R. Morris, "L+: Scalable Landmark Routing and Address Lookup for Multi-Hop Wireless Network," Technical Report 837, Mass. Inst. of Technology LCS, Mar. 2002.
- [4] A. Ephremides, J.E. Wieselthier, and D.J. Baker, "A Design Concept for Reliable Mobile Radio Networks with Frequency Hopping Signaling," *Proc. IEEE '87*, vol. 75, no. 1, pp. 56-73, Jan. 1987.
- [5] J. Eriksson, M. Faloutsos, and S. Krishnamurthy, "Scalable Ad Hoc Routing: The Case for Dynamic Addressing," *Proc. IEEE INFOCOM*, 2004.
- [6] J. Ghosh, S.J. Philip, and C. Qiao, "Sociological Orbit Aware Location Approximation and Routing (SOLAR) in MANET," *Proc. ACM MobiHoc*, 2005.
- [7] J. Haas, J.Y. Halpern, and L. Li, "Gossip-Based Ad Hoc Routing," *Proc. IEEE INFOCOM*, 2002.
- [8] P. Hui, J. Crowcroft, and E. Yoneki, "Bubble Rap: Social-Based Forwarding in Delay Tolerant Networks," *Proc. ACM MobiHoc*, 2008.
- [9] S. Jain, K. Fall, and R. Patra, "Routing in a Delay Tolerant Network," *Proc. ACM SIGCOMM*, 2004.
- [10] J. Leguay, T. Friedman, and V. Conan, "DTN Routing in a Mobility Pattern Space," *Proc. ACM SIGCOMM Workshop Delay-Tolerant Networking*, 2005.
- [11] A. Lindgren, A. Doria, and O. Schelen, "Probabilistic Routing in Intermittently Connected Networks," *Lecture Notes on Computer Science*, vol. 3126, pp. 239-254, Aug. 2004.
- [12] C. Liu and J. Wu, "Scalable Routing in Delay Tolerant Networks," *Proc. ACM MobiHoc*, 2007.
- [13] S. Merugu, M. Ammar, and E. Zegura, "Routing in Space and Time in Network with Predictable Mobility," Technical Report GIT-CC-04-07, College of Computing, Georgia Tech, 2004.
- [14] G. Pei, M. Gerla, X. Hong, and C. Chiang, "A Wireless Hierarchical Routing Protocol with Group Mobility," *Proc. IEEE Wireless Comm. and Networking Conf. (WCNC)*, 1999.
- [15] R. Ramanathan, P. Basu, and R. Krishnan, "Towards a Formalism for Routing in Challenged Networks," *Proc. ACM MobiCom Workshop Challenged Networks (CHANTS)*, 2007.
- [16] R. Ramanathan and M. Steenstrup, "Hierarchically Organized, Multihop Mobile Wireless Networks for Quality of Service Support," *Mobile Networks and Applications*, vol. 3, no. 1, pp. 101-119, 1998.
- [17] P. Sinha, R. Sivakumar, and V. Bharghavan, "Enhancing Ad Hoc Routing with Dynamic Virtual Infrastructures," *Proc. IEEE INFOCOM*, 2001.
- [18] M.M.B. Tariq, M. Ammar, and E. Zegura, "Message Ferry Route Design for Sparse Ad Hoc Networks with Mobile Nodes," *Proc. ACM MobiHoc*, 2005.
- [19] A. Vahdat and D. Becker, "Epidemic Routing for Partially-Connected Ad Hoc Networks," technical report, Duke Univ., 2000.
- [20] J. Wu, S. Yang, and F. Dai, "Logarithmic Store-Carry-Forward Routing in Mobile Ad Hoc Networks," *IEEE Trans. Parallel and Distributed Systems*, vol. 18, no. 6, June 2007.
- [21] 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," *Towards a Formalism for Routing in Challenged Networks*, 2007.
- [22] W. Zhao, M. Ammar, and E. Zegura, "A Message Ferrying Approach for Data Delivery in Sparse Mobile Ad Hoc Networks," *Proc. ACM MobiHoc*, 2004.



Cong Liu received the BS degree in microelectronics from South China University of Technology in 2002, the MS degree in computer software and theory from Sun Yat-sen (Zhongshan) University in 2005, and the PhD degree from the Department of Computer Science and Engineering, Florida Atlantic University in 2009. He is affiliated with Shenzhen Institute of Advanced Technology, Chinese Academy of Sciences, and the Department of Computer Science, Sun Yat-sen (Zhongshan) University. He has served as a reviewer for many conferences and journal papers. His main research interests include routing in mobile ad hoc networks (MANETs) and delay-tolerant networks (DTNs), deep packet inspection, and transaction processing.



Jie Wu is chairman and professor in the Department of Computer and Information Sciences, Temple University. He has published more than 400 papers in various journals and conference proceedings. His research interests are in the areas of wireless networks and mobile computing, routing protocols, fault-tolerant computing, and interconnection networks. He was on the editorial board of *IEEE Transactions on Parallel and Distributed Systems* and was a coguest editor of *Computer* and the *Journal of Parallel and Distributed Computing*. He served as a program cochair for MASS 2004, the program vice chair for ICDCS 2008, and the program vice chair for ICPP 2000. He was also the general chair for MASS 2006 and is the general chair for IPDPS 2008. He is the author of the text *Distributed System Design* (CRC Press). He was also the recipient of the 1996-1997, 2001-2002, and 2006-2007 Researcher of the Year Award at Florida Atlantic University. He has served as an IEEE Computer Society distinguished visitor and is the chairman of the IEEE Technical Committee on Distributed Processing (TCDP). Dr. Wu served as a program director at the National Science Foundation (2007-2008). He is a fellow of the IEEE.

► For more information on this or any other computing topic, please visit our Digital Library at www.computer.org/publications/dlib.