Hypercube-Based Multipath Social Feature Routing in Human Contact Networks

Jie Wu, *Fellow*, *IEEE*, and Yunsheng Wang

Abstract—Most routing protocols for delay tolerant networks resort to the sufficient state information, including trajectory and contact information, to ensure routing efficiency. However, state information tends to be dynamic and hard to obtain without a global and/or long-term collection process. In this paper, we use the internal *social features* of each node in the network to perform the routing process. In this way, feature-based routing converts a routing problem in a highly mobile and unstructured contact space to a static and structured feature space. This approach is motivated from several human contact networks, such as the Infocom 2006 trace and MIT reality mining data, where people contact each other more frequently if they have more social features in common. Our approach includes two unique processes: social feature extraction and multipath routing. In social feature extraction, we use entropy to extract the m most informative social features to create a feature space (F-space): $(F_1, F_2, \ldots, F_m)$, where F_i corresponds to a feature. The routing method then becomes a hypercube-based feature matching process, where the routing process is a step-by-step feature difference resolving process. We offer two special multipath routing schemes: node-disjoint-based routing and delegation-based routing. Extensive simulations on both real and synthetic traces are conducted in comparison with several existing approaches, including spray-and-wait routing, spray-and-focus routing, and social-aware routing based on betweenness centrality and similarity. In addition, the effectiveness of multipath routing is evaluated and compared to that of single-path routing.

Index Terms—Closeness, delay tolerant networks, entropy, human contact networks, hypercubes, multipath routing, social features

1 INTRODUCTION

Delay tolerant networks (DTNs) are characterized by intermittent connectivity and limited network capacity. There exist several different applications: connectivity of developing countries [1], vehicular communications [2], and *human contact networks* (HCNs) [3], [4]. In HCNs, such as pocket social networks [5], where individuals move around and interact at each contact point based on their common interests, social features play an important role. HCNs are usually used for scenarios where the network infrastructure is not available, and establishing it within a short time is infeasible.

Several social-aware routing schemes have been proposed recently [2], [6], [7], [8]. Most of these approaches consider the trajectory and/or the contact history of mobile nodes. However, most state information is dynamic and hard to obtain without a global and/or long-term collection process. In this paper, we use the internal features of a node (an individual) for routing guidance. These features include nationality, affiliation, speaking language, and so on. This approach is motivated from the fact that *people come in contact with each other more frequently if they have more social features in common*.

In Table 1, we show the difference in contacts when various features differ in the Infocom 2006 conference trace [9] and MIT reality mining data [10]. We can see that the total number of contacts decreases when the social feature difference between two individuals increases. Mei et al. [11] found that individuals with similar social features tend to contact each other more often in HCNs. Hence, we believe that designing a new routing protocol by considering the social features of individuals can improve routing performance.

One advantage of using features for routing guidance is its avoidance of state information collection. *Feature-based routing converts a routing problem in a highly mobile and unstructured contact space (M-space) to a static and structured feature space (F-space)*. More specifically, each individual is represented by a vector of $(F_1, F_2, \ldots, F_m)$, where each feature F_i has n_i distinct values for $i = 1, 2, \ldots, m$. In this way, the F-space contains $\prod_{i=1}^{m} n_i$ nodes. Structurally, these nodes form an *m-dimensional hypercube*, as shown in Fig. 1, in which two nodes are connected if and only if they differ in one feature. When $n_i = 2$ for all i, it is called a *binary hypercube*.

Although the initial idea of feature-based routing was proposed earlier in [11], our approach provides a systematic way of multipath routing in the F-space by taking advantage of the structural property of hypercubes. We start by giving a model for representing the social features of each individual, and introducing a method to measure the social similarities between individuals. Generally, each individual has many social features; however, some features are more important than others for routing purposes. Hence, in the *social feature extraction* process, we use Shannon entropy [12] to select m key social features. After that, individuals can be partitioned into different groups, each of which corresponds to a position in a feature space (i.e., a hypercube node).

To perform efficient multipath routing, node-disjoint routing is used to which a hypercube-based parallel feature

- *J. Wu is with the Department of Computer and Information Sciences, Temple University, 324 Wachman Hall, 1805 N. Broad St., Philadelphia, PA 19122. E-mail: jiewu@temple.edu.*
- *Y. Wang is with the Department of Computer Science, Kettering University, 1700 University Avenue, Flint, MI 48504. E-mail: ywang@kettering.edu.*

Manuscript received 5 Jan. 2012; revised 6 Aug. 2012; accepted 9 Aug. 2012; published online 28 Aug. 2012.
For information on obtaining reprints of this article, please send e-mail to: tc@computer.org, and reference IEEECS Log Number: TC-2012-01-0013.
Digital Object Identifier no. 10.1109/TC.2012.209.

TABLE 1
Contacts in *Infocom 2006* and *MIT Reality Mining*

Feature distance	0	1	2	3	4	5
Infocom Contacts	53,500	6,720	4,390	2,070	1,420	970
MIT Contacts	213,500	92,300	50,100	28,200	17,500	9,700

matching process is applied. Feature differences are resolved in a step-by-step manner until the destination is reached. We also propose a feature matching shortcut algorithm for fast searching, which also ensures node-disjointness. Another way to achieve efficient multipath routing is to extend delegation forwarding [13], [14]. In delegation forwarding, a copy is made to a newly encountered node if this node is "closer" to the destination than the current node. Here, we use feature closeness as a forwarding metric, and apply a feature-distance-based metric for copy redistribution.

In the simulation, we compare node-disjoint-based routing and delegation-based routing with spray-and-wait [15], spray-and-focus [16], and social-aware routing schemes, (such as SimBet [17]) based on betweenness centrality and similarity, both in synthetic and real traces. To evaluate the impact of node density on the routing performance, we examine three cases in terms of the relative order between N (number of individuals in HCNs) and $M = 2^m$ (number of group nodes in the F-space): 1) $M \ll N (i.e., M = o(N))$, 2) $M = N (M = \Theta(N))$, and 3) $M \gg N (M = O(N))$.

The major contributions of our work are as follows:

- We convert the HCN routing problem from the mobile contact space into the social feature space, and use entropy to extract the most informative features to create a hypercube.
- We present two efficient multipath routing schemes under the hypercube structure: node-disjoint based and delegation based.
- We extend multipath routing to general hypercubes and cube-connected-cubes (CCCs).
- We compare the cost effectiveness of multipath routing and single-path routing based on different metrics for path construction.
- We include social-aware routing for comparison, which uses betweenness centrality and similarity to make the forwarding decision.
- We evaluate the proposed scheme in both synthetic and real traces (INFOCOM 2006 and MIT reality mining). The simulation results show the competitive performance of multipath routing.

In summary, our approach makes use of social features for efficient routing without resorting to external information, such as contact information.

2 Preliminaries

2.1 Objectives

The objective of this paper is to develop an efficient multipath routing scheme based on hypercube social feature matching in HCNs. Three performance metrics are used to measure the performance: 1) *delivery rate*: the average delivery ratio of the routing packet; 2) *latency*: the average duration between the generation time and the arrival time of a packet; and 3) *number of forwardings*: the average number of forwardings of each packet before the destination is reached.

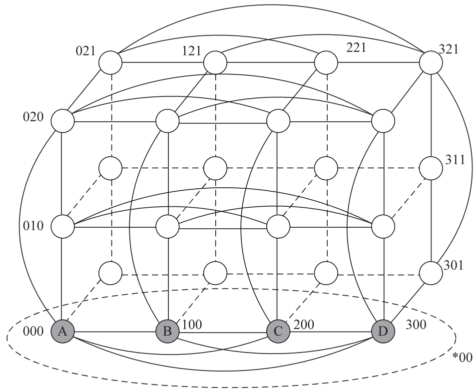


Fig. 1. A three-dimensional hypercube.

Efficient routing entails a high delivery rate, low latency, an acceptable number of forwardings, and a limited number of copies of the packet.

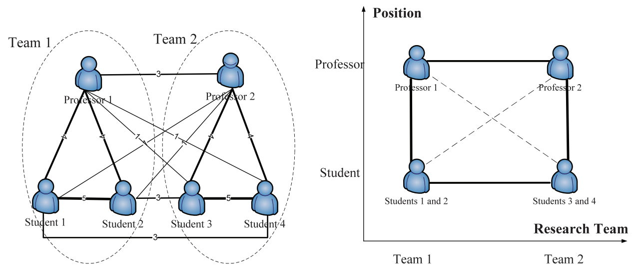
2.2 Social Features

Assume that there are N individuals in the system. Each individual can be represented by a social feature profile, a representation of her/his social features within a *feature space*, also called the F-space. The social features represent either physical features, such as gender, or logical ones, such as membership in a social group.

In this paper, we convert the mobile and unstructured contact space (M-space) with N individuals into a static and structured feature space (F-space) with M groups (or simply nodes), each consisting of a set of individuals in M-space with the same features. Therefore, there is no one-to-one correspondence between an individual in the M-space and a group (node) in the F-space.

Fig. 1 represents a $4 \times 3 \times 2$ F-space. It consists of 24 groups. In this example, there are three different social features in the F-space, represented by four, three, or two distinct values, respectively. In the F-space in Fig. 1, dimension 1 (the left most position) corresponds to *city*, with four distinct values: New York (0), London (1), Paris (2), and Shanghai (3); dimension 2 (the second left most position) shows *position*, with three distinct values: professor (0), researcher (1), and student (2); dimension 3 represents *gender* with two distinct values: male (0) and female (1). In Fig. 1, two groups have a connection if they differ in exactly one feature.

Fig. 2 shows an example of transforming an M-space into an F-space. Professor 1 has two PhD students, 1 and 2, and professor 2 has two PhD students, 3 and 4. Every week, each individual's contact time is shown in Fig. 2a. The professors contact their own students more frequently than another research team's students. The students tend to contact students on their own team more frequently than students on a team different than their own. We can convert the M-space into F-space as shown in Fig. 2b; these six individuals are divided into four groups and form a 2×2 F-space based on their social features. If student 1 wants to discuss an idea with professor 2, student 1 only has one chance to meet with professor 2 directly each week. In



(a) (b)

Fig. 2. An example of transforming from M-space to F-space.

Fig. 2b, there are two indirect paths from student 1 to professor 2. Student 1 can contact his supervisor (professor 1) first, then professor 1 can contact professor 2; or he can contact one of professor 2's students, then that student can contact professor 2. In this way, the contact opportunities significantly increase. The dashed lines in Fig. 2b are shortcuts, which we will discuss in detail later.

2.3 Hypercubes and Hypercube Routing

Given the above definition of the feature space, we can represent the social feature profile for a group of users as a group node (or node for short) in a hypercube. More specifically, the F-space $(F_1, F_2, \dots, F_m)$ is mapped into an m-dimensional hypercube (or simply m-D cube), which consists of $n_1 \times n_2 \times \cdots \times n_m$ nodes. Two nodes, $A = (a_1, a_2, \dots, a_m)$ and $B = (b_1, b_2, \dots, b_m)$, in an m-D cube are connected if and only if they differ in exactly one dimension (say i, such that $a_i \neq b_i$). To express the virtual similarity between individuals in a cube, we use the feature distance to measure the closeness between two individuals.

The binary hypercube is a special cube in which each feature has a binary value: 0 or 1. In an m-D binary cube, there are $M = 2^m$ groups (nodes). The feature distance between two individuals, A and B, is denoted as H_{AB}, which is the Hamming distance between A and B. We assume that source S has a packet for destination D, with feature distance k, in an m-D binary cube. There are exactly m node-disjoint paths from S to D based on the hypercube property [18], [19]. These paths are composed of k shortest paths of length k, and $m - k$ nonshortest paths of length $k + 2$.

In binary cube routing, the relative address of the current node and the destination is calculated through using XOR on two addresses and is sent, along with the packet, to the next node. The relative distance is updated at each step until it becomes zero at the destination group. We will extend this routing scheme by adding shortcuts for fast feature matching in multipath routing.

2.4 Delegation Forwarding

In delegation forwarding [13], each node has its estimated distance to the destination that is measured by quality (Q). Initially, the quality level (L) of each node is equal to its Q. A packet holder only forwards the packet to a node with a higher quality than its own level. In addition, the packet holder raises its own level to the quality of the higher quality node. This means that a node will duplicate and forward a packet only if it encounters another node whose quality value is higher than any node met by the packet so far. It is shown that the expected cost of delegation forwarding in an N-node network is $O(\sqrt{N})$, compared to $O(N)$ in the naïve scheme of forwarding to any higher quality node [13]. Chen et al. [14] enhanced the performance of delegation forwarding through a probability model. In this paper, we use the feature distance as the quality value of the node to a given destination.

TABLE 2
Entropy of the Social Features in Real Traces

Infocom Feature	Entropy	MIT Feature	Entropy
$Affiliation$	4.64	*Neighborhood*	4.13
$City$	4.45	*Daily commute*	4.05
$Nationality$	4.11	*Hangouts*	3.86
$Language$	4.11	*Working hour*	3.54
$Country$	3.59	*Affiliation*	2.56
$Position$	1.37	*Research group*	1.53

3 Feature Extraction

The individuals are characterized by a high dimensional feature profile. However, usually only a small subset of features is important, extracted by the feature extraction method from data mining [20], [21].

There are N individuals with m' features, which are denoted as $F_1, F_2, \dots, F_{m'}$. The goal of our social feature extraction process is to extract the *most informative subset* (MIS) with $m (< m')$ key features. We use Shannon entropy [12], which quantifies the expected value of the information contained in the feature, to select the key features:

$$E(F_i) = -\sum_{k=1}^{n_i} p(x_k) log_2 p(x_k), \quad (i = 1, 2, \dots, m'), \tag{1}$$

where $E(F_i)$ denotes the entropy of the feature F_i, and p denotes the probability mass function of F_i. $\{x_1, \dots, x_{n_i}\}$ are the possible values of F_i. The entropy considers not only the number of possible values, but also the distribution of their frequencies.

Table 2 shows the entropy of each social feature that we obtained from the Infocom 2006 trace [9]: $m = 6$ most informative features out of $m' = 10$ total features. It also lists $m = 6$ most informative features from the MIT reality mining data.

4 Multipath Routing

We present a novel *social feature-based multipath routing* scheme with the objective of reaching the destination quickly, while maximizing the delivery rate. The constraint is the number of copies of the packet. The main objective is to distribute the copies of the packet in a cost-effective way.

We propose two special multipath routing schemes: *node-disjoint based*, where the copies are distributed to multiple node-disjoint paths to resolve the feature difference between the source and destination, and *delegation-based*, where the dissemination of copies is based on the feature distance to the destination.

We use the features of the destination to partition nodes into groups. This approach is called *destination-based partitioning*. At each dimension (i.e., feature), we separate

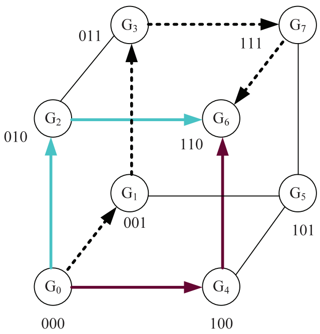


Fig. 3. An example of node-disjoint-based routing with $S = G_0$ and $D = G_6$. Solid directed paths are the shortest paths, and dashed directed paths represent the nonshortest paths.

nodes based on whether they have the same features as the one at the destination or not. In this way, a general cube is "compressed" into a binary cube, even though each feature may have many different values. We will discuss another approach that uses general cubes directly in Section 6. Our routing scheme focuses on the group level, i.e., a node in a cube. Note that each group has many individuals who have the same partially matched features as the destination. The routing packet is forwarded from group to group until it reaches the destination group—the group where the destination is located. The packet can then be forwarded once more to the destination, which is in the same group.

4.1 Node-Disjoint-Based Routing

The source has m copies of the packet to send to the destination in k feature distances. In an m-D binary cube, there are k shortest paths of length k and $m-k$ nonshortest paths of length $k+2$, which are all node-disjoint.

Suppose that the source and destination differ in k dimensions $\{1, 2, ..., k\}$, denoted as a set C. $C^0 : \langle 1, 2, ..., k\rangle$ is defined as the *coordinate sequence* (or *sequence*) from a given C. C^0 determines how a path is constructed based on the resolution order of dimension differences given in C^0. C^i is defined as i circular left shifts of C^0. In fact, C^0 can be any permutation of C. Then, k sequences, $C^0, C^1, \ldots, C^{k-1}$, will create k node-disjoint shortest paths from C:

- *Path* 1 generated by C^0: $\langle 1, 2, 3, \ldots, k\rangle$;
- *Path* 2 generated by C^1: $\langle 2, 3, 4, \ldots, k, 1\rangle$;
- *Path* 3 generated by C^2: $\langle 3, 4, 5, \ldots, k, 1, 2\rangle$;

  ......
- *Path* k generated by C^{k-1}: $\langle k, 1, 2, \ldots, k-2, k-1\rangle$.

Here, the path generated from source S by sequence C^0 follows a matching process along dimension 1, dimension 2, and so on. In Fig. 3, from node G_0 with sequence $\langle 1, 2\rangle$, the path is (G_0, G_4, G_6). In hypercube routing, the coordinate sequence of a path is sent along with the packet. After a successful forwarding along dimension i, dimension i will be deleted from the sequence. Clearly, the sequence becomes an empty sequence upon reaching the destination.

In Algorithm 1, for the source node, the source sends $(seq, mode)$ to a matching neighbor, where *mode* is 0 for a shortest path or 1 for a nonshortest path. *seq* is the result of a circular left shift of C^0 for $mode = 0$. D is the destination. The routing packet is not included in the notation for simplicity. The source also maintains two sets, d and d'. d is initialized as $\{1, 2, \ldots, k\}$, which are different features between the source and destination. d' is $\{k, k+1, \ldots, m\}$.

Algorithm 1. Node-Disjoint-based Routing: source node contacts D or neighbor B in dimension i.

1: **if** B and D are in the same group **then**
2: Forward the packet to D.
3: **else**
4: **case** $i \in d$: $d = d/\{i\}$ and send $(C^i, 0)$ to B.
5: **case** $i \in d'$: $d' = d'/\{i\}$ and send $(C\|i, 1)$ to B.
6: **case** $i \notin d \cup d'$: do nothing.

More specifically, when the source meets a neighbor with a feature difference in $i \in d$, which represents a dimension in a shortest path, the source sends C^i with $mode = 0$ (which represents a strict coordinate sequence in C^i) and removes i from d. If $i \in d'$, which represents a dimension in a nonshortest path, the source sends sequence $C^0\|i$ with $mode = 1$ (which represents any permutation of C followed by i) and removes i from d'. If $i \notin d \cup d'$, no action is needed, as is shown in step 6, where the encountered node comes from a dimension to which a copy has been sent earlier.

In Algorithm 2, for a nonsource node, source routing is used when the routing path is determined by the packet header *seq*. Step 4 represents shortest-path routing, where a strict coordinate sequence order is followed through extracting the first dimension in C'. Step 5 corresponds to nonshortest path routing, where any permutation of dimension differences can be used. In Fig. 3, the nonshortest path can be either $(G_0, G_1, G_3, G_7, G_6)$, as shown in the figure, or $(G_0, G_1, G_5, G_7, G_6)$.

Algorithm 2. Node-Disjoint-based Routing: non-source node contacts D or neighbor B in i with $(seq : C', mode : m)$.

1: **if** B and D are in the same group **then**
2: Forward the packet to D.
3: **else**
4: **case** $m = 0 \wedge i = first(C')$: send $(C'/\{i\}, 0)$ to B.
5: **case** $m = 1 \wedge i \in C'$: send $(C'/\{i\}, 1)$ to B.

We also propose the *feature matching shortcut* for fast searching. In traditional hypercube routing, each forwarding can only correct one dimension at a time. When a packet holder meets another individual who is more than one feature distance away and is closer to the destination, the packet will not be forwarded to that individual. Here, we allow a controlled jump to a group that is more than one feature distance away, while still ensuring node-disjointness. Such a controlled jump is called a *shortcut*, which is a *prefix*[1] of the coordination sequence. In Fig. 3, G_0 can forward a copy of the packet directly to G_6 as a shortcut for path (G_0, G_4, G_6).

4.2 Delegation-Based Routing

Delegation-based routing forwards the copies of a packet only to the individual that has a smaller feature distance to the destination. The number of copies to be forwarded is proportional to the feature distance to the destination.

In delegation-based routing, shown in Algorithm 3, there are two values to determine packet forwarding: *quality value* and *level value*. We use feature distance as the quality value.

1. Subsequence $\langle 1, 2, \ldots, k'\rangle$ is a *prefix* of $\langle 1, 2, \ldots, k\rangle$, where $k' \leq k$.

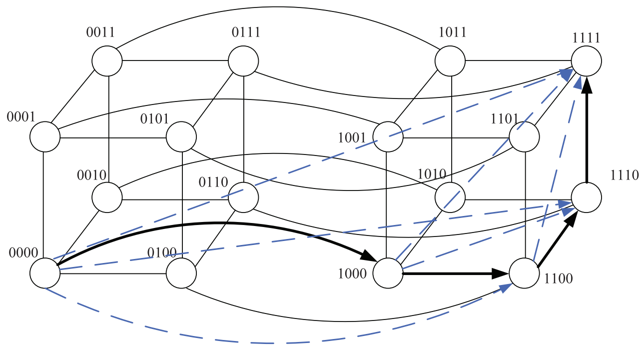


Fig. 4. An example of the composite path from 0000 to 1111, where dashed directed lines are shortcuts.

The quality value (Q_{AD}) of individual A with destination D is inversely proportional to the feature distance between A and D; that is, $Q_{AD} = 1/H_{AD}$. We simply use Q_A to represent Q_{AD}. When H_{AD} is 0, we set Q_A to $+\infty$. Initially, the level value (L_A), the highest level that A has met so far, is the same as Q_A.

Algorithm 3. Delegation-based Routing.
1: /* Individual A meets B, A has a packet with c copies and B has no copy for destination D. */
2: Initialize $L_A \leftarrow Q_A$.
3: **if** $L_B > L_A$ **then**
4: Forward $\lceil (1 - L_A/L_B) \cdot c \rceil$ copies to B
5: $L_A \leftarrow L_B$.

In Algorithm 3, when A, with c copies of the packet, meets B, who has no copy but has a higher quality level L_B (note that $L_B = Q_B$ in this case) than A's level L_A, A will forward $\lceil (1 - L_A/L_B) \cdot c \rceil$ copies of the packet to B and update its level value to L_B.

5 Analysis

5.1 Node-Disjointness

The multiple paths in hypercube routing are node-disjoint. The benefit of node-disjointness is that it guarantees that the multiple paths will not cross each other, except at destination D, to increase the efficiency of the routing. In this section, we prove that by including shortcuts, these paths still remain node-disjoint.

Theorem 1. *In node-disjoint-based routing, the multiple paths with shortcuts are still node-disjoint paths.*

Proof. The nonshortcut paths are generated based on results in [18], [19], which are k node-disjoint paths of length k, and $m - k$ node-disjoint paths of length $k + 2$. All of these paths are generated through coordinate sequences starting from the source. Because each shortcut is a prefix of a coordinate sequence, all resultant paths still remain node-disjoint. □

As shown in Fig. 3, one individual in G_0 has a packet for another individual in G_6. The shortest paths are (G_0, G_2, G_6) and (G_0, G_4, G_6), which follow the coordinate sequences that we discussed in Section 4.1. The nonshortest path can be $(G_0, G_1, G_3, G_7, G_6)$. These three paths are node-disjoint. The shortcuts from the nonshortest path are (G_0, G_6), (G_0, G_7), (G_0, G_3), (G_1, G_6), (G_1, G_7), and (G_3, G_6).

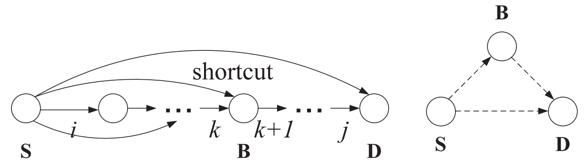


Fig. 5. An illustration of contact frequency.

5.2 Contact Frequency

We use the classic probability theory to make some observations. We assume that the contact probability is time-independent [22]. We use contact numbers from the most recent time window to estimate contact probability (or precisely, frequency).[2] More specifically, an individual in group S has $p_1, p_2, \ldots, p_m$ average contact frequencies with its m neighbors along m dimensions that match the destination features in an m-D binary cube. $p_{1,2,\ldots,k}$ is denoted as the average contact frequency between an individual from S and any individual in group D that has matching destination features, where S and D differ in k features $1, 2, \ldots, k$. Note that this frequency is not symmetric (i.e., the frequency from S to D is not the same as from D to S). For simplicity, when we consider a path, its coordinate sequence is a consecutive ascending sequence, such as $\langle 1, 2, \ldots, k \rangle$.

We conduct an experiment and obtain four social features with the highest entropy values (affiliation, city, nationality, and language) from the Infocom 2006 trace to create a 4D binary cube, as shown in Fig. 4. For the MIT reality mining data, we perform the same process. There are $2^4 = 16$ groups in the cube. The source 0000 here represents a general source. If the destination has a different feature value than the source in a dimension, the corresponding bit is set to 1. In Fig. 4, we use destination 1111 to illustrate.

From Fig. 5, we can consider a *virtual directed triangle* with three nodes: S, B, and D. S to B include dimensions $i, i + 1, \ldots, k$. B to D have dimensions $k + 1, k + 2, \ldots, j$. Hence, S to D span dimensions $i, i + 1, \ldots, j$. When $j = i + 1$, it corresponds to a *regular directed triangle* with A and B (and B and D) differing in exactly one bit position.

We define $P'_{i,\ldots,j}$, called *composite frequency*, as the frequency of a path from source S to destination D in the following dimension sequence $\langle i, i + 1, \ldots, j \rangle$: this includes all possible shortcuts. The corresponding path is called a *composite path*, as shown in Fig. 4 from 0000 to 1111. This is the summation of the frequencies of all possible paths that follow the dimension sequence. In Fig. 5, we denote the composite frequency from S to D as $P'_{S,\ldots,D}$. P_{ij} represents the frequency of a shortcut from dimension i to dimension j, which is equal to $p_{i(i+1)\ldots j}$. We call P_{ij} the *shortcut frequency*. The shortcut frequency from S to D is denoted as P_{SD}. The *direct frequency*, $P_{i,\ldots,j} = p_i p_{i+1} \ldots p_j$, corresponds to a direct path in our routing process from S to D, which is denoted as $P_{S\ldots D}$.

Theorem 2. $P'_{i\ldots j} = \sum_{k=i}^{j} P_{ik} P'_{k+1\ldots j}$, *where* $i < j$, *and* $i \leq k \leq j$. $P'_{i\ldots i} = P_{ii} = p_i$.

2. Although the contact duration is also important, results in [23] and Fig. 1 show that there are high correlation coefficients of duration and frequency in many traces; we simply consider only frequency in this paper.

TABLE 3
Comparison of Contact Frequency with Different Feature Distance in the Infocom 2006 (MIT Reality Mining) Trace

Path	Frequency	
(0000, 1000)	$p_1 = 0.196(0.178)$	P_{11}
(1000, 1100)	$p_2 = 0.183(0.185)$	P_{22}
(1100, 1110)	$p_3 = 0.192(0.183)$	P_{33}
(1110, 1111)	$p_4 = 0.188(0.19)$	P_{44}
(0000, 1100)	$p_{12} = 0.04(0.03)$	P_{12}
(1000, 1110)	$p_{23} = 0.039(0.035)$	P_{23}
(1100, 1111)	$p_{34} = 0.041(0.038)$	P_{34}
(0000, 1110)	$p_{123} = 0.019(0.007)$	P_{13}
(1000, 1111)	$p_{234} = 0.018(0.008)$	P_{24}
(0000, 1111)	$p_{1234} = 0.01(0.003)$	P_{14}
(0000, 1000, 1100)	$p_1p_2 \approx 0.036(0.033)$	$P_{1..2}$
(1000, 1100, 1110)	$p_2p_3 \approx 0.035(0.034)$	$P_{2..3}$
(1100, 1110, 1111)	$p_3p_4 \approx 0.036(0.035)$	$P_{3..4}$
(0000, 1000, 1100, 1110)	$p_1p_2p_3 \approx 0.007(0.006)$	$P_{1..3}$
(1000, 1100, 1110, 1111)	$p_2p_3p_4 \approx 0.007(0.0064)$	$P_{2..4}$
(0000, 1000, 1100, 1110, 1111)	$p_1p_2p_3p_4 \approx 0.0013(0.0011)$	$P_{1..4}$
(0000, 1000, 1100, 1110)	$p_1p_2p_3 \approx 0.007(0.0066)$	$P'_{1..3}$
(0000, 1100, 1110)	$p_{12}p_3 \approx 0.008(0.0055)$	
(0000, 1000, 1110)	$p_1p_{23} \approx 0.008(0.0062)$	
(0000, 1110)	$p_{123} = 0.019(0.007)$	
(1000, 1100, 1110, 1111)	$p_2p_3p_4 \approx 0.007(0.0064)$	$P'_{2..4}$
(1000, 1110, 1111)	$p_{23}p_4 \approx 0.007(0.0067)$	
(1000, 1100, 1111)	$p_2p_{34} \approx 0.008(0.007)$	
(1000, 1111)	$p_{234} = 0.018(0.008)$	
(0000, 1000, 1100, 1110, 1111)	$p_1p_2p_3p_4 \approx 0.0013(0.0011)$	$P'_{1..4}$
(0000, 1000, 1100, 1111)	$p_1p_2p_{34} \approx 0.0015(0.0013)$	
(0000, 1000, 1110, 1111)	$p_1p_{23}p_4 \approx 0.0014(0.0012)$	
(0000, 1100, 1110, 1111)	$p_{12}p_3p_4 \approx 0.0014(0.001)$	
(0000, 1000, 1111)	$p_1p_{234} \approx 0.0035(0.0014)$	
(0000, 1110, 1111)	$p_{123}p_4 \approx 0.0036(0.0013)$	
(0000, 1100, 1111)	$p_{12}p_{34} \approx 0.0016(0.0011)$	
(0000, 1111)	$p_{1234} = 0.01(0.003)$	

Proof. Without loss of generality, we assume the source as S and the destination as D. The corresponding coordinate sequence is $\langle i, i+1 \dots j\rangle$, as shown in Fig. 5. From Fig. 5, each P_{ik} ($i \leq k \leq j$) corresponds to a *prefix shortcut* from S to D. $P'_{k+1,\dots,j}$ corresponds to the composite frequency of the remaining path to destination D. A simple summation of these paths enumerates each possible path from S to D. □

Table 3 records the shortcut, direct, and composite frequencies. Destination 1111 is generic and includes all nodes as possible destinations. The binary cube is constructed based on destination-based partitioning which was discussed earlier. Here, we consider path $(0000, 1000, 1100, 1110, 1111)$, which is one of the shortest paths from the source to the destination. From Table 3, we have the following two observations that relate to virtual and regular directed triangles.

Observation 1. $P'_{S\dots D} < P'_{S\dots B}$, $P'_{S\dots D} < P'_{B\dots D}$, and $P'_{S\dots D} > P'_{S\dots B}P'_{B\dots D}$. This means that the composite frequency in the hypotenuse is smaller than each side of the triangle and is larger than the product of the composite frequencies of two sides.

Observation 2. $P_{SD} < P_{SB}$, $P_{SD} < P_{BD}$, $P_{SD} > P_{SB}P_{BD}$. This means that the shortcut frequency in the hypotenuse is smaller than each side, and is larger than the product of the composite frequency of two sides.

Observation 1 holds for both Infocom 2006 and MIT reality mining. Observation 2 holds for Infocom 2006 and most of the time for MIT reality mining. Based on Observation 2, we can use induction to show that $P_{SD} > P_{S\dots D}$, which means that the shortcut frequency is larger than the direct frequency for any path.

From the above observations, we can use shortcuts for fast deliveries in terms of a smaller number of forwardings and a shorter delivery time. For a given source S and destination D, we conjecture that the direct frequency $P_{S\dots D}$ is a lower bound of shortcut frequency P_{SD}, and the composite frequency $P'_{S\dots D}$ is an upper bound of P_{SD}. In our synthetic trace simulation, we will use these two bounds to generate shortcuts.

6 Extensions

6.1 General Hypercubes

In the previous sections, we discussed multipath routing in a binary cube. We can extend this routing scheme to the general cube, with multiple distinct values in each dimension without compression. We can extend the basic scheme by treating all nodes that differ in a particular feature as a *clique*, i.e., a complete subgraph. Fig. 2 shows a clique $*00$ in a dark color, where $*$ is a wild card for 0, 1, 2, or 3. The corresponding nodes in the clique are A, B, C, and D, respectively.

Although each pair of nodes in the clique is directly connected, they may not come in contact with each other in the near future (i.e., a low contact frequency). In Fig. 1, we assume that A holds a packet for D that has the same value as the destination address in that dimension (D is called a *destination at a dimension*). If node A meets another node that has a higher contact frequency to node D, forwarding is allowed. This is the same idea as delegation forwarding, but is used within one dimension. We call this approach *general forward*. The contact frequency is calculated locally, based on the 2-hop contact history at each node, without resorting to global contact information.

To control the hop-count, we can modify general forward to allow the packet to be forwarded twice at most in each dimension. We call this approach *2-hop general forward*. In this way, we can control the total number of forwardings. In the above example, when A comes in contact with B, who has a higher contact frequency with D, A will forward the packet to B. Then, B will hold the packet until it meets D.

In our simulation, we compare general forward and 2-hop general forward with *general wait* (i.e., routing schemes in binary hypercubes in Section 4.1), which will hold the packet until meeting with the destination at a particular dimension.

6.2 Cube-Connected-Cubes

When the initial number of copies of the packet is less than the number of dimensions, we can use cube-connected-cubes to enhance the performance.

We assume that there are only c copies of the packet in the source, and the destination is k feature distances away with $k > c$. Here, we assume that k is divisible by c. In CCCs, k dimensions are partitioned into c groups, each of which includes k/c dimensions. To offer a good partition for braiding relevant features into the same group, first we pick the highest entropy feature F_i and select k/c largest values of *mutual information* $I(F_i; F_j)$ [12] (details in the next paragraph) as the most relevant features to be braided with F_i. Here, F_j is an unselected feature. Then, we repeat the same process with the remaining features to create c groups of braided features. In CCCs with k dimensions, an inner (k/c)-dimensional cube can be considered as a node of an outer c-dimensional cube. Inside the inner cube, there are $2^{k/c}$ paths, and the outer cube has c node-disjoint paths for c copies. Therefore, CCCs explore more paths when compared to the basic scheme.

The mutual information of two feature variables, X and Y, can be defined as

$$I(X;Y) = \sum_{y \in Y} \sum_{x \in X} p(x,y) log\left(\frac{p(x,y)}{p(x)p(y)}\right), \tag{2}$$

where $p(x, y)$ is the *joint probability distribution function* of X and Y, and $p(x)$ and $p(y)$ are the marginal probability distribution functions of X and Y, respectively. $p(x, y)$ is equal to the product of $p(x)/p(y)$ and the conditional probability $p(y|x)/p(x|y)$: $p(x, y) = p(y|x)p(x) = p(x|y)p(y)$ [12]. Mutual information quantifies the dependence between the joint distribution of X and Y. $I(X;Y)$ is larger when features X and Y are more similar.

In the simulation, we compare our method (*entropy-based*) with random feature braiding (*random*) and parallel path routing (*parallel*).

6.3 Shortest Single-Path Routing

To reduce the overhead, we propose a single-path routing scheme. In an m-D binary cube, we assume that each pair of neighboring nodes meet according to their contact frequencies, λ_i for $1 \leq i \leq m \cdot 2^{m-1}$.[3] We assume that the expected forwarding time from one individual to its neighbor is equal to $\frac{1}{\lambda_i}$. Then, we use Dijkstra's algorithm to select the shortest time path from the source to the destination.

We consider the m-D binary cube to be a weighted graph $G = (V, E)$. We assume that the weight is the forwarding time that $w(u, v) \geq 0$ for each edge $(u, v) \in E$. Dijkstra's algorithm maintains a set S of vertices whose final shortest path and time from the source s have already been determined. The algorithm repeatedly selects the vertex $u \in V - S$ that has the minimum shortest time estimation, then adds u to S, and relaxes the edge (u, v). This means that the forwarding time will be adjusted from s to v through u. In Algorithm 4, we use a min-priority queue Q of vertices, keyed by their forwarding time values.

3. There are $m \cdot 2^{m-1}$ edges in an m-D binary cube.

Algorithm 4. Shortest Single-Path Routing.

```
Initialize the source (G, s)
S ← Ø
Q ← V[G]
while Q ≠ Ø do
    u ← extract min(Q)/* minimum forwarding time
    from s to u, u ∈ Q*/
    S ← S ∪ {u}
    Record the path from s to u
    Relax the edge (u, v)/* adjust forwarding time from
    s to v through u*/
```

After using Dijkstra's algorithm, we have the shortest paths from the source to all other vertices. Hence, we can use the shortest path from the source to the destination for routing guidance.

Fig. 6 illustrates the whole execution process of shortest single-path routing. 000 is the source s, and the time from the other vertices to s is infinity initially, as shown in Fig. 6a. In the next step, s will check the forwarding time and compare it to its neighboring vertices: 100, 010, and 001. From Fig. 6b, we can see that path (000, 100) has the shortest time. Hence, vertex 100 will be selected, and path (000, 100) will be recorded. By repeatedly doing the selection process, we will get the shortest forwarding time and shortest paths from the source to other vertices. If we want to send a packet from source 000 to destination 111, path (000, 100, 101, 111) is the shortest path for routing guidance, and the shortest forwarding time is 9.

Although single-path routing can reduce the number of forwardings when compared with multipath routing in Section 4, it may increase the delivery time. We will compare these two routing schemes in our simulation. Note that shortest single-path routing does require global information. Our objective is to show that our node-disjoint multipath routing scheme can compete with the best single-path routing scheme.

7 Related Work

The simplest DTN routing scheme is flooding or epidemic routing [24]. To control the copies of the packet, Grossglauser and Tse [25] introduced 2-hop routing, where the source gives a copy to relay nodes, each of which holds the packet until it contacts the destination. In [15], [16], two multicopy routing schemes, spray-and-wait and spray-and-focus, are proposed. The source spray-and-wait is the same as 2-hop routing. Binary spray always halves the number of copies at each spray; it allows multihop unless the current node has one copy left. Spray-and-focus goes further to allow multihop, even when there is only one copy. Multihop is based on a quality metric like delegation forwarding [13]. Our approach differs in that the split is proportional to the quality of the two encountered nodes.

Many social-aware approaches have been proposed [2], [6], [7], [8]. Daly and Haahr [17] analyzed the social network characterizing information for routing and proposed SimBet routing scheme based on a node's ego-centric betweenness centrality and a node's social similarity. In BUBBLE Rap [26], Hui et al. also used human mobility, in terms of social structures, in the design of forwarding algorithms for pocket

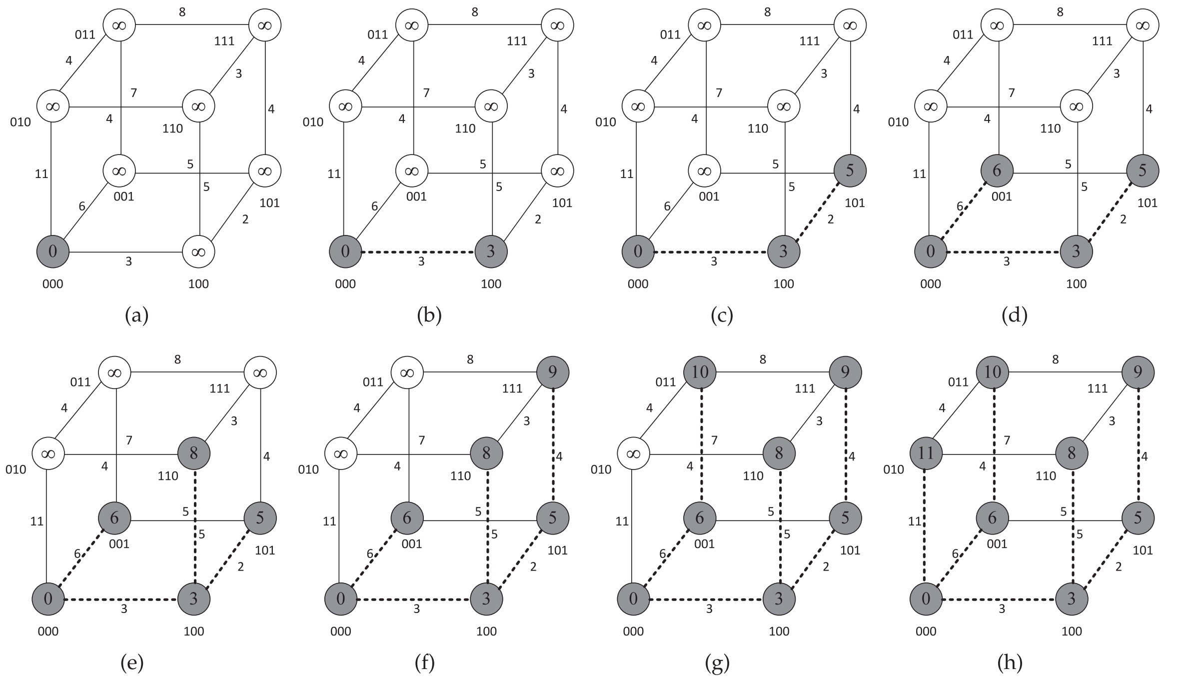


Fig. 6. The execution of shortest single-path routing.

switched networks. In addition, a special destination community, k-clique [27], is used where nodes in the destination community can be selected as relay nodes to speed up the routing process.

Existing social-aware approaches resort to sufficient state information, including trajectory and contact, to ensure routing efficiency. Such information is expensive to obtain, especially in a dynamic network such as a HCN, although some predictive models [22], [28] can be applied. In this paper, the internal feature used does not incur any cost in state information collection, to guide the routing process.

Detecting clusters or communities in large real-world graphs, such as large social or information networks, is a problem of considerable interest. The existing destination community structures, including k-clique [27] and the Girvan-Newman algorithm [29], which form communities based on the high edge betweenness of intercommunity links, all depend on a prior knowledge of the global contact information among nodes. For fair comparisons, we only selected social-aware routing schemes that use local contact information, i.e., ego-centric metrics.

Among existing feature selection methods include [12], [20], [21]. Shannon entropy [12] and the mutual information-based information theoretic filter (ITF) [30] have received significant amounts of attention. However, obtaining mutual information requires global information. Shannon entropy is easy to calculate, thus, we use Shannon entropy to extract the most informative social feature.

The applications of hypercubes have been initially studied in parallel and distributed computing [18], [19]. There have been some recent works on hypercube routing in wireless networks [31], [32], [33]. Our approach utilizes the advantage of hypercube properties so that multiple paths are guaranteed to be node-disjoint, a property that is absent in existing DTN multipath routing protocols.

8 SIMULATION

We compare the performance of the proposed multipath routing scheme with several existing ones, including spray-and-wait and spray-and-focus, in Matlab, using both real and synthetic traces. In our simulation, the 90 percent confidence interval of each result is within 1 percent. However, the marks for confidence intervals make these figures less discernible. Therefore, we decide to leave one curve in a figure (Fig. 7) with confidence intervals and remaining ones without confidence intervals.

The simulation is grouped into four categories:

1. *Varying node density*. We show that multipath routing is robust under different node density conditions.
2. *The importance of the nonshortest path in node-disjoint-based routing*. We illustrate the impact of the nonshortest path in node-disjoint multiple path routing.
3. *Comparing the extensions*. We compare different methods in two extensions: general hypercubes and cube-connected-cubes.

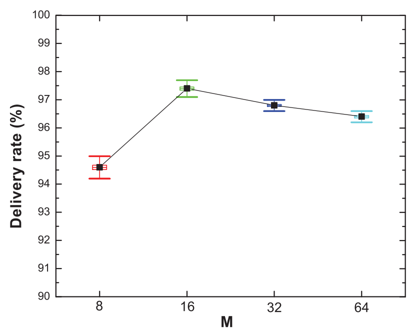


Fig. 7. Confidence level for node-disjoint based with wait-at-destination scheme.

4. *Multipath versus single path*. We compare the performance between node-disjoint-based multiple path routing and shortest single-path routing.

8.1 Different Routing Schemes under Comparison

Our node-disjoint-based scheme dramatically increases the delivery rate compared with the BUBBLE Rap scheme [26] from overall 50 to 80 percent in MIT reality mining data. By comparing with the SimBet scheme [17], our scheme reduces the latency significantly. At the same time, our scheme has similar delivery rate and number of forwarding compared with SimBet. However, these schemes have different settings. To conduct a more fair comparison, we extract several key mechanisms used in these schemes. They are similarity and betweenness as metrics for delegation.

In category 1, we implement and compare seven routing schemes. The first four are our proposed schemes. In all schedules, we consider m copies.

1. *Node-disjoint based with wait-at-destination (ND-W)*. Waiting for the destination after the packet enters the destination group in node-disjoint-based routing.
2. *Node-disjoint-based with spray-at-destination (ND-S)*. Spraying $N/(2M)$ copies into the destination group after the packet enters the group in node-disjoint-based routing.
3. *Delegation-based with wait (D-W)*. Same final step as ND-W in delegation-based routing.
4. *Delegation-based with spray (D-S)*. Same final step as ND-S in delegation-based routing.
5. *Similarity-based delegation (S-D)*. Using the social similarity[4] [17] as the quality level in delegation-based routing.
6. *Betweenness-based delegation (B-D)*. Using the ego-centric betweenness centrality[5] [17] as the quality level in delegation-based routing.
7. *Source spray-and-wait (S-S&W)*. *Spray* phase: the source forwards copies to the first m distinct nodes that it encounters. At the end of the spray, each packet holder has one copy; *Wait* phase: if the destination is not found in the spray phase, the copy carriers wait for the destination.
8. *Binary spray-and-wait (B-S&W)*. *Spray* phase: any node with copies will forward half of the copies to the encountered node with no copy; *Wait* phase: the same as S-S&W.
9. *Binary spray-and-focus (B-S&F)*. *Spray* phase: same as B-S&W; *Focus* phase: if the destination is not found in the spray phase, the copy carriers forward the copy to the encountered node with a smaller feature distance to the destination.

In category 4, we compare node-disjoint-based multiple path routing and shortest single-path routing.

1. *Shortest single-path routing (SSP)*. A shortest path-based single-path routing scheme in hypercube-based DTNs.
2. *Special node-disjoint-based routing (SND)*. A special node-disjoint-based routing scheme in which the coordinate sequence C^0 is scheduled by the shortest single-path route, followed by the sequence generation $C^1, C^2, \ldots, C^{k-1}$, based on C^0, as discussed in Section 4.1.
3. *Random single-path routing (RSP)*. Randomly choose a single-path to guide the routing process.
4. *Random node-disjoint-based routing (RND)*. The coordinate sequence C^0 is randomly chosen, followed by the sequence generation $C^1, C^2, \ldots, C^{k-1}$, based on C^0, as discussed in Section 4.1.

Feature-based forwarding can be combined with existing contact frequencies, such as the ones used in delegation forwarding and social-aware routing for the forwarding decision. To demonstrate the effectiveness of each metric, we compare all metrics individually in the simulation.

8.2 Simulation Methods and Setting

Real trace. We use two real traces, the *Infocom 2006* trace [9] and the *MIT reality mining* data [10], in our simulation.

The Infocom 2006 data set consists of two parts: *contacts* between the iMote devices that are carried by participants and *social features* of the participants, which are the statistics of participants' information from a questionnaire form. First, we discard some participants that do not have social features in their profiles. In this way, we reduce the number of participants to 61. There are 74,981 contacts between these participants over a period of 337,418 time slots in seconds. We extract five social features from the original data set: nationality, language, affiliation, position, and country.

The MIT reality mining data set also consists of *contacts* and *social features*. After discarding some participants with incomplete input information, we reduce the number of participants to 57. There are 411,313 contacts between these participants over a period of 897,921 time slots in seconds. We extract five social features from the original data set: daily commute, research group, affiliation, neighborhood, hangouts, and working hours.

Synthetic trace. We assume the contact frequency between pairwise individuals with only one different feature. That is, a node A has m contact frequencies, $p_1, p_2, \ldots, p_m$, with its m neighbors in the m-D F-space. To estimate the contact frequency of node B, which is more than one feature distance away from A, the shortcut frequency P_{AB} is randomly selected between its lower bound ($P_{A\ldots B}$, direct frequency) and its upper bound ($P'_{A\ldots B}$, composite frequency). In our synthetic trace, we create 128 individuals and 50,000 time slots in seconds. Contacts are randomly selected from these time slots based on selected frequencies.

Setting in practice. In the current version, we require all nodes to share the same feature set. In the extension, we allow for some missing features (similar to do not-care). All relevant features are decided offline.

8.3 Simulation Results

8.3.1 Varying Node Density

We compare the performance of multipath routing with spray-and-wait and spray-and-focus with varying node densities. In both the Infocom 2006 trace and the MIT reality mining trace, as we selected six social features, we set the number of nodes ($M = 2^m$) in the F-space to 8, 16, 32,

4. The similarity presents the number of common neighbors between the current node and the destination node.

5. If each node has local information of neighbors and neighbors' neighbors, ego-centric betweenness centrality measures how much a node connects neighbors that are not directly connected themselves.

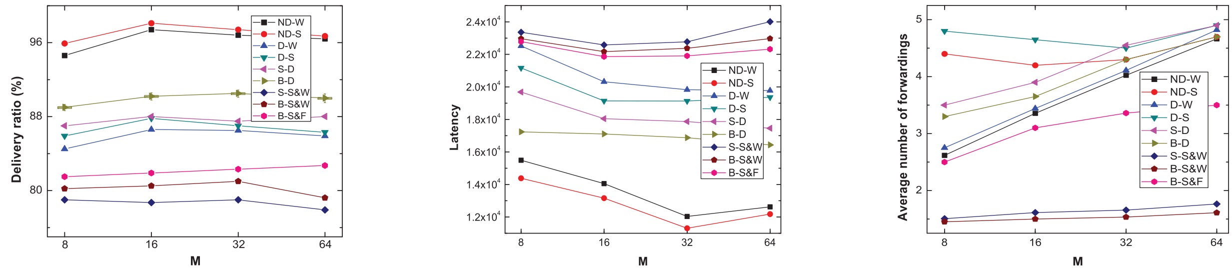


Fig. 8. Comparing with 20 packets in Infocom 2006 trace: (L): delivery rate, ($Center$): latency, and (R): number of forwardings.

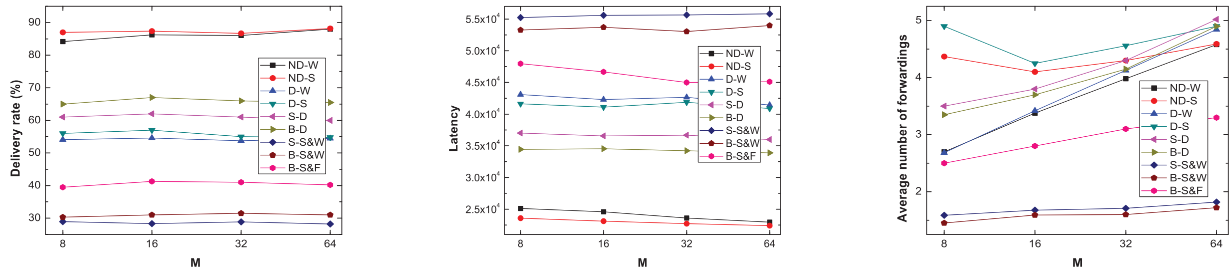


Fig. 9. Comparing with 100 packets in Infocom 2006 trace: (L): delivery rate, ($Center$): latency, and (R): number of forwardings.

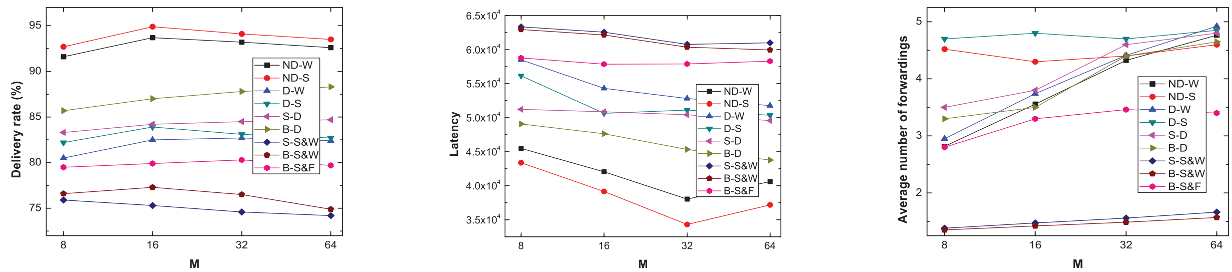


Fig. 10. Comparing with 20 packets in MIT reality mining trace: (L): delivery rate, ($Center$): latency, and (R): number of forwardings.

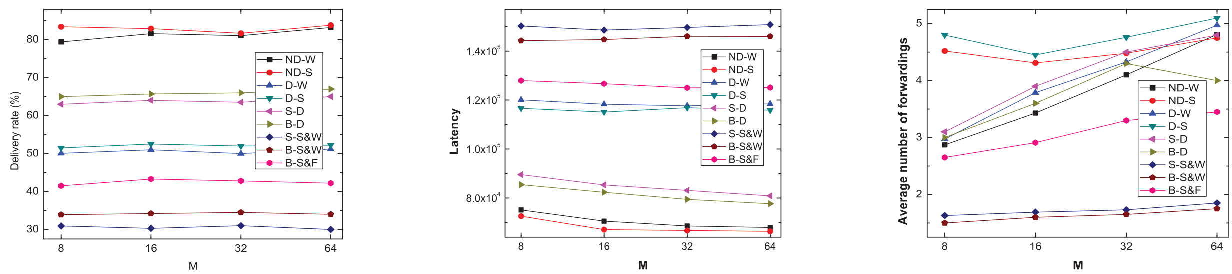


Fig. 11. Comparing with 100 packets in MIT reality mining trace: (L): delivery rate, ($Center$): latency, and (R): number of forwardings.

and 64. In the synthetic trace, we set 16, 32, 128, 256, and 2,048 nodes in the F-space to examine different schemes at a larger scale. We also compare these routing schemes with 20 and 100 packets, which are created at the rate of one packet per five time slots.

From Figs. 8, 9, 10 and 11, we can see that node-disjoint-based routing has the highest delivery rate and the lowest latency among all in the Infocom 2006 and MIT reality mining traces. Delegation-based routing performs better than the spray-and-wait and spray-and-focus schemes in both delivery rate and latency. Binary spray-and-wait has the smallest number of forwardings before reaching the destination, as it does not forward once there is only one copy left. Node-disjoint-based routing can reduce forwardings by about 6 percent, compared to delegation-based routing. Node-disjoint-based routing has a higher delivery rate, shorter latency, and a smaller number of forwardings compared with the similarity-based and betweenness-based forwarding schemes. The betweeness-based delegation forwarding scheme performs better than the similarity-based delegation forwarding scheme. Both of them perform better than the delegation based with wait scheme.

Out of all of the multipath routing schemes, spray-at-destination increases the delivery rate by 3 percent and reduces the latency by 5.7 percent, compared to wait-at-destination. Although, the former will increase the number of forwardings when the node density is high. We also find that using shortcuts can increase the delivery rate by about 5.3 percent, cut latency by 14.6 percent, and reduce the number of forwardings by 7 percent, as seen in Figs. 12 and 13.

As the results in the 20 and 100 packet conditions show the same trend, we only report results for the 100 packet condition in the synthetic trace. Node-disjoint-based routing has the highest delivery rate and the lowest latency in Fig. 14. Multipath routing increases the average

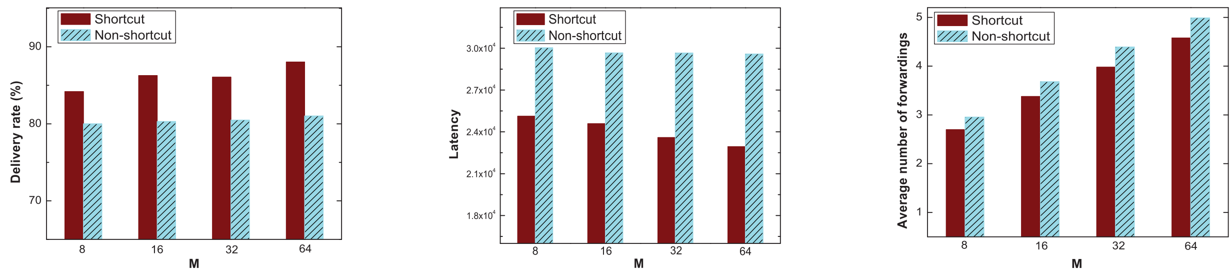


Fig. 12. Comparing *shortcut* and *nonshortcut* with 100 packets in Infocom 2006 trace: (L): delivery rate, ($Center$): latency, and (R): number of forwardings.

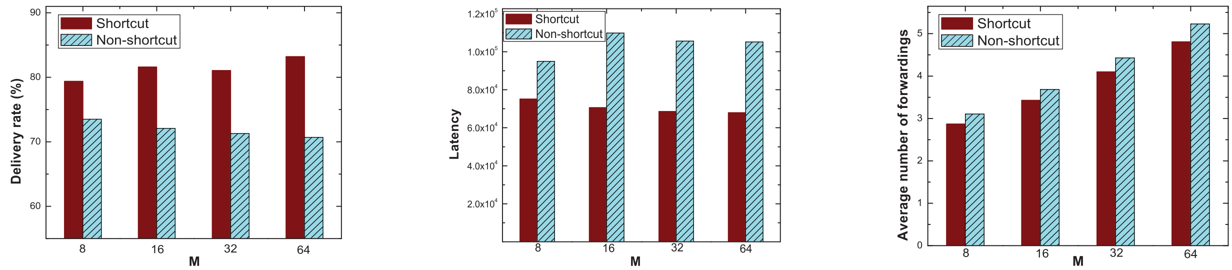


Fig. 13. Comparing *shortcut* and *nonshortcut* with 100 packets in MIT reality mining trace: (L): delivery rate, ($Center$): latency, and (R): number of forwardings.

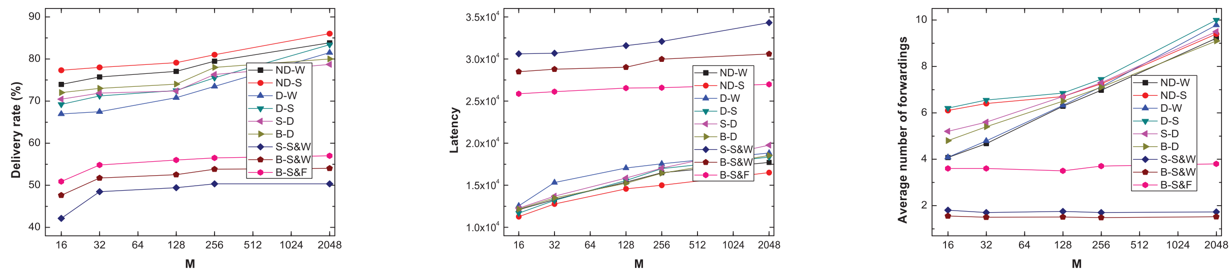


Fig. 14. Comparing with 100 packets in the synthetic trace: (L): delivery rate, ($Center$): latency, and (R): number of forwardings.

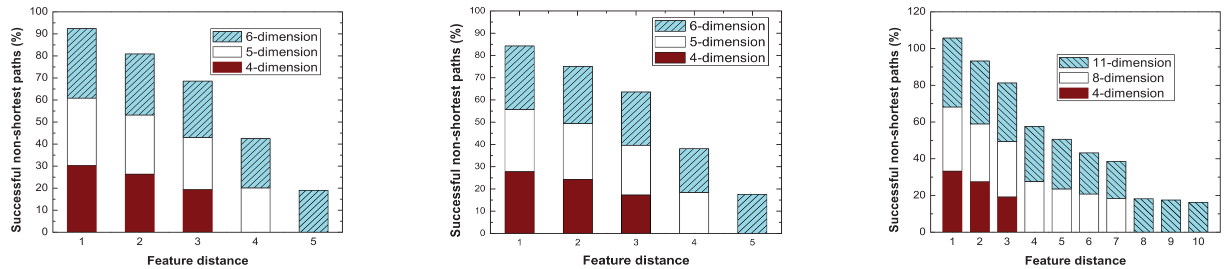


Fig. 15. Comparing in the percentage of the successful nonshortest path: (L): Infocom trace, ($Center$): MIT trace, and (R): synthetic trace.

number of forwardings compared to spray-and-wait and spray-and-focus. Using shortcuts improves the overall performance in node-disjoint-based routing.

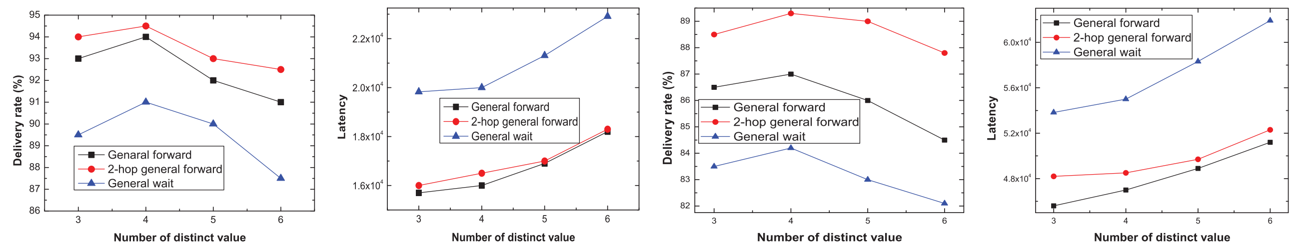
8.3.2 Nonshortest Path

Our simulation demonstrates the importance of the nonshortest paths in node-disjoint-based routing. In the real trace, we set the dimension of the F-space (m) to 4, 5, and 6. In the synthetic trace, it is 4, 8, and 11. We compare the performance under different feature distances (k). Both the real and synthetic traces show that as $m - k$ increases, the percentage of the nonshortest paths that reach the destination before any shortest path also increases in Fig. 15. This is expected, as there are $m - k$ nonshortest paths and k shortest paths in a given m-D binary cube.

8.3.3 Extensions

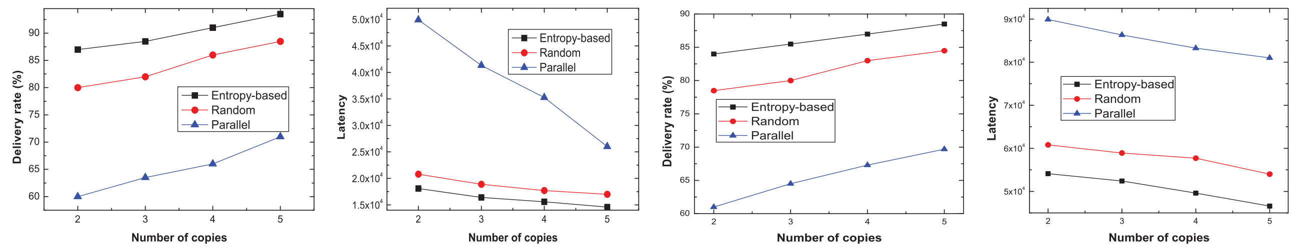
In general hypercubes, we compare the three methods that were discussed in Section 6.1: *general forward*, *2-hop general forward*, and *general wait*, in real traces. In Fig. 16, 2-hop general forward is the best, as it increases the delivery rate, shortens the hop-count, and reduces the latency compared with general forward. Although general wait can further reduce the hop-count, it will, at the same time, reduce the delivery rate and increase the latency significantly.

In CCCs, we compare the three methods that were discussed in Section 6.2: *entropy-based braiding*, *random braiding*, and *parallel path routing*, under a given number of copies in real traces. We set six social features under both traces. In Fig. 17, we can see that entropy-based braiding has the best performance in terms of delivery rate and



(a) Delivery rate (b) Latency (c) Delivery rate (d) Latency

Fig. 16. Comparing in general hypercubes in real traces: *Left two*: Infocom and *Right two*: MIT.



(a) Delivery rate (b) Latency (c) Delivery rate (d) Latency

Fig. 17. Comparing in CCCs in real traces: *Left two*: Infocom and *Right two*: MIT.

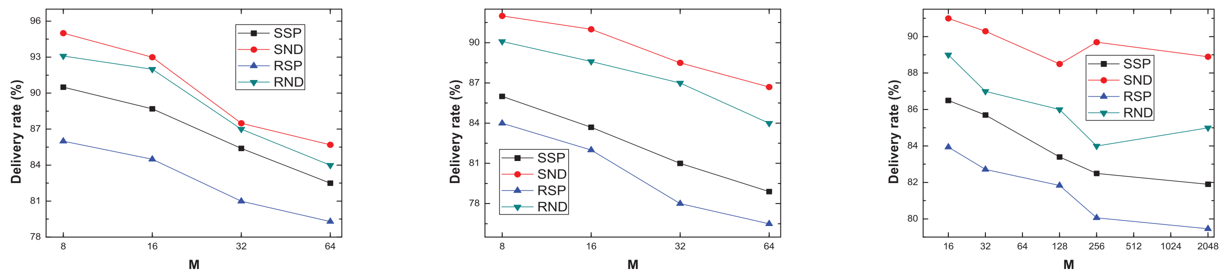


Fig. 18. Comparing the delivery rate between multipath routing and single-path routing: (L): Infocom trace, ($Center$): MIT trace, and (R): synthetic trace.

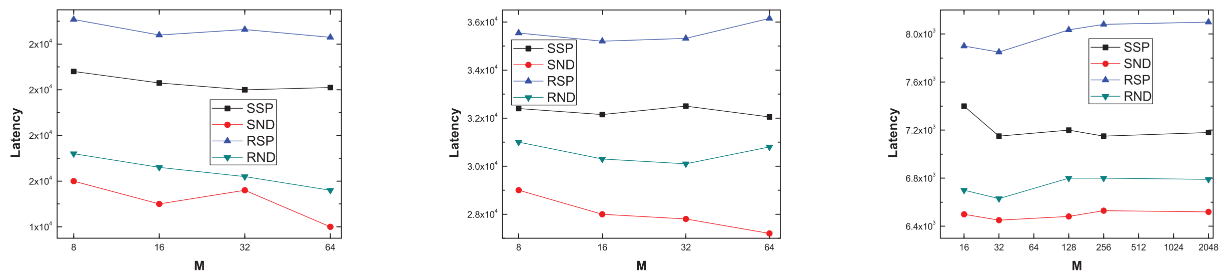


Fig. 19. Comparing the latency between multipath routing and single-path routing: (L): Infocom trace, ($Center$): MIT trace, and (R): synthetic trace.

latency. The performance decreases noticeably when using parallel path routing.

8.3.4 Multipath versus Single Path

In Figs. 18, 19, and 20, we find that, although multipath routing increases the number of forwardings, it can increase the delivery rate and reduce the latency in both the real and synthetic traces. In the Infocom 2006 trace, SND increases the delivery rate by about 4.5 percent and reduces the latency by about 15 percent compared to SSP. RND increases the delivery rate by about 7.6 percent and reduces the latency by about 17 percent compared to RSP. SND performs better than RND. At the same time, SSP is better than RSP. In the MIT reality mining trace and the synthetic trace, the trends are the same as in the Infocom 2006 trace. Hence, we believe that multipath routing that uses multiple parallel paths to search the destination can improve the delivery rate and reduce the latency.

8.4 Summary of Simulation

Our simulation concludes that, although multipath routing increases the number of forwardings compared to spray-and-wait, spray-and-focus, and social-aware routing based on betweenness centrality and similarity, it has a significantly higher delivery rate and reduces the latency, especially under node-disjoint-based routing. Node-disjoint-based routing has multiple node-disjoint paths, which help to improve search efficiency. In node-disjoint-based routing, shortcuts also can increase the delivery rate, lower the latency, and reduce the number of forwardings, at the same time. The nonshortest path also plays an important role, especially when the number of node-disjoint paths is

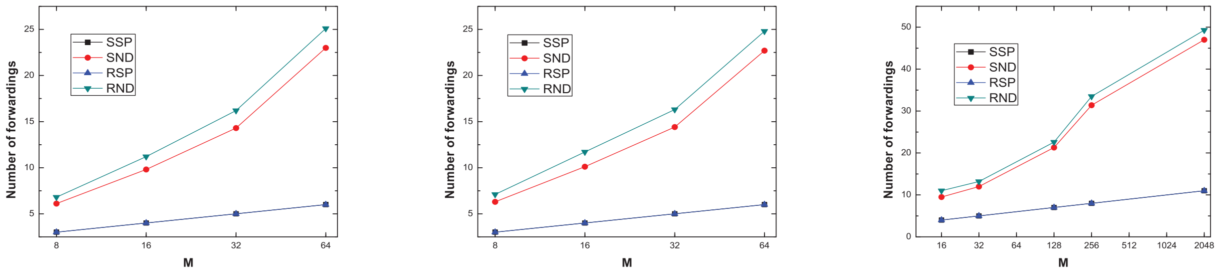


Fig. 20. Comparing the number of forwardings between multipath routing and single-path routing: (L): Infocom trace, ($Center$): MIT trace, and (R): synthetic trace.

limited. When the node density is relatively high, there are more individuals in each group. Therefore, using spray-at-destination seems to be a viable solution to reduce the latency compared with wait-at-destination.

Simulation results of two extensions show that the proposed multipath routing scheme can also achieve a competitive performance in general hypercubes, which is more practical in reality. Multipath routing can still be effective when the number of copies is limited. This can be done through dimension braiding in CCCs. The competitive performances in all of the extensions verify that multipath routing can be effective under different conditions in DTN routing.

By comparing the performance between multipath and single-path routing, we find that multipath routing has multiple paths to search for the destination simultaneously, which performs better than single-path routing. Using Dijkstra's algorithm to choose the shortest path can also increase the quality of the performance.

9 Conclusion

In this paper, we proposed a social feature-based multipath routing scheme for human contact networks, a special form of delay tolerant networks. Our scheme has two parts: social feature extraction and multipath routing. We used entropy to extract the most informative social features, and build an m-dimensional hypercube. In multipath routing, we presented two schemes: node-disjoint based and delegation based. In node-disjoint-based routing, the feature difference between the source and the destination is resolved in a step-by-step fashion during the routing process. Shortcuts were used for fast matching. In delegation-based routing, we extended delegation forwarding into the multicopy model, which uses a feature-distance-based metric as the copy forwarding decision metric. Trace-driven simulation results showed that our proposed multipath routing scheme performs better than spray-and-wait, spray-and-focus, and social-aware routing schemes based on betweenness centrality and similarity.

We believe that the social features will play an important role in routing in human contact networks. Our future work will include more experiments on different social network traces to validate our two observations. We also plan to study more sophisticated routing schemes in general hypercubes. Combinations of feature-based and contact-frequency-based schemes will be also investigated.

Acknowledgments

This research was supported in part by US National Science Foundation (NSF) grants ECCS 1128209, CNS 10655444, CCF 1028167, CNS 0948184, and CCF 0830289, and National Basic Research Program of China (973) no. 2009CB320705.

References

[1] A. Pentland, R. Fletcher, and A. Hasson, "DakNet: Rethinking Connectivity in Developing Nations," *Computer*, vol. 37, no. 1, pp. 78-83, Jan. 2004.
[2] J. Burgess, B. Gallagher, D. Jensen, and B.N. Levine, "MaxProp: Routing for Vehicle-Based Disruption-Tolerant Networks," *Proc. IEEE INFOCOM*, 2006.
[3] J. Scott, J. Crowcroft, P. Hui, and C. Diot, "Haggle: A Networking Architecture Designed around Mobile Users," *Proc. Third Ann. Conf. Wireless On-Demand Network Systems and Services (WONS '06)*, 2006.
[4] K.E. Defrawy, J. Solis, and G. Tsudik, "Leveraging Social Contacts for Message Confidentiality in Delay Tolerant Networks," *Proc. IEEE 33rd Ann. Int'l Computer Software and Applications Conf. (COMPSAC '09)*, pp. 271-279, 2009.
[5] P. Hui, A. Chaintreau, J. Scott, R. Gass, J. Crowcroft, and C. Diot, "Pocket Switched Networks and Human Mobility in Conference Environments," *Proc. ACM SIGCOMM Workshop Delay-Tolerant Networking (WDTN '05)*, pp. 244-251, 2005.
[6] A. Chaintreau, P. Hui, J. Crowcroft, C. Diot, R. Gass, and J. Scott, "Impact of Human Mobility on Opportunistic Forwarding Algorithms," *IEEE Trans. Mobile Computing*, vol. 6, no. 6, pp. 606-620, June 2007.
[7] J. Wu and Y. Wang, "A Non-Replication Multicasting Scheme in Delay Tolerant Networks," *Proc. IEEE Seventh Int'l Conf. Mobile Adhoc and Sensor Systems (MASS)*, 2010.
[8] G.S. Thakur, A. Helmy, and W.-J. Hsu, "Similarity Analysis and Modeling in Mobile Societies: The Missing Link," *Proc. Fifth ACM Workshop Challenged Networks (CHANTS '10)*, 2010.
[9] J. Scott, R. Gass, J. Crowcroft, P. Hui, C. Diot, and A. Chaintreau, "CRAWDAD Trace Cambridge/Haggle/Imote/Infocom2006 (v. 2009-05-29)," 2009.
[10] A.P.N. Eagle and D. Lazer, "Inferring Social Network Structure Using Mobile Phone Data," *Proc. Nat'l Academy of Sciences USA*, vol. 106, no. 36, pp. 15274-15278, Sept. 2009.
[11] A. Mei, G. Morabito, P. Santi, and J. Stefa, "Social-Aware Stateless Forwarding in Pocket Switched Networks," *Proc. IEEE INFOCOM*, 2011.
[12] C. Shannon, N. Petigara, and S. Seshasai, "A Mathematical Theory of Communication," *Bell System Technical J.*, vol. 27, pp. 379-423, 1948.
[13] V. Erramilli, M. Crovella, A. Chaintreau, and C. Diot, "Delegation Forwarding," *Proc. ACM MobiHoc*, 2008.
[14] X. Chen, J. Shen, T. Groves, and J. Wu, "Probability Delegation Forwarding in Delay Tolerant Networks," *Proc. 18th Int'l Conf. Computer Comm. and Networks (ICCCN)*, 2009.
[15] T. Spyropoulos, K. Psounis, and C.S. Raghavendra, "Spray and Wait: an Efficient Routing Scheme for Intermittently Connected Mobile Networks," *Proc. ACM SIGCOMM Workshop Delay-Tolerant Networking (WDTN '05)*, 2005.

[16] T. Spyropoulos, K. Psounis, and C.S. Raghavendra, "Spray and Focus: Efficient Mobility-Assisted Routing for Heterogeneous and Correlated Mobility," *Proc. IEEE Fifth Ann. Int'l Conf. Pervasive Computing and Comm. Workshops (PERCOMW),* 2007.
[17] E.M. Daly and M. Haahr, "Social Network Analysis for Routing in Disconnected Delay-Tolerant MANETs," *Proc. ACM MobiHoc,* pp. 32-40, 2007.
[18] J. Wu, *Distributed System Design.* CRC Press, 1998.
[19] Y. Saad and M. Schultz, "Topological Properties of Hypercubes," *IEEE Trans. Computers,* vol. 37, no. 7, pp. 867-872, July 1988.
[20] A. Jain and D. Zongker, "Feature Selection: Evaluation, Application, and Small Sample Performance," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 19, no. 2, pp. 153-158, Feb. 1997.
[21] I. Guyon and A. Elisseeff, "An Introduction to Variable and Feature Selection," *The J. Machine Learning Research,* vol. 3, pp. 1157-1182, 2003.
[22] M.C. Gonzalez, C.A. Hidalgo, and A.-L. Barabasi, "Understanding Individual Human Mobility Patterns," *Nature,* vol. 453, no. 7196, pp. 779-782, 2008.
[23] T. Hossmann, T. Spyropoulos, and F. Legendre, "Putting Contacts into Context: Mobility Modeling Beyond Inter-Contact Times," *Proc. ACM MobiHoc,* 2011.
[24] A. Vahdat and D. Becker, "Epidemic Routing for Partially Connected Ad Hoc Networks," technical report, Dept. of Computer Science, Duke Univ., 2000.
[25] M. Grossglauser and D. Tse, "Mobility Increases the Capacity of Ad Hoc Wireless Networks," *IEEE/ACM Trans. Networking,* vol. 10, no. 4, pp.477-486, Aug. 2002.
[26] P. Hui, J. Crowcroft, and E. Yoneki, "Bubble Rap: Social-Based Forwarding in Delay Tolerant Networks," *Proc. ACM MobiHoc,* pp. 241-250, 2008.
[27] G. Palla, I. Dernyi, I. Farkas, and T. Vicsek, "Uncovering the Overlapping Community Structure of Complex Networks in Nature and Society," *Nature,* vol. 435, no. 7043, pp. 814-818, 2005.
[28] K. Lee, Y. Kim, S. Chong, I. Rhee, and Y. Yi, "Delay-Capacity Tradeoffs for Mobile Networks with Levy Walks and Levy Flights," *Proc. IEEE INFOCOM,* pp. 3128-3136, 2011.
[29] M. Girvan and M.E.J. Newman, "Community Structure in Social and Biological Networks," *Proc. Nat'l Academy of Sciences USA,* vol. 99, no. 12, pp. 7821-7826, 2002.
[30] F.M. Reza, *An Introduction to Information Theory.* Dover Publications, Inc., 1994.
[31] R. Manoharan and P. Thambidurai, "Hypercube Based Team Multicast Routing Protocol for Mobile Ad Hoc Networks," *Proc. Ninth Int'l Conf. Information Technology (ICIT '06),* 2006.
[32] C.-T. Chang, C.-Y. Chang, and J.-P. Sheu, "Bluecube: Constructing a Hypercube Parallel Computing and Communication Environment over Bluetooth Radio System," *Proc. Int'l Conf. Parallel Processing (ICPP),* 2003.
[33] H. Huo, W. Shen, Y. Xu, and H. Zhang, "Virtual Hypercube Routing in Wireless Sensor Networks for Health Care Systems," *Proc. First Int'l Conf. Future Information Networks (ICFIN),* 2009.

Jie Wu (F'09) is the chair and a Laura H. Carnell professor in the Department of Computer and Information Sciences at Temple University. Before joining Temple University, he was a program director at the US National Science Foundation and a distinguished professor at Florida Atlantic University. His research interests include wireless networks, mobile computing, routing protocols, fault-tolerant computing, and interconnection networks. His publications include more than 600 papers in scholarly journals, conference proceedings, and books. He has served on several editorial boards, including *IEEE Transactions on Computers* and *Journal of Parallel and Distributed Computing*. He was the general cochair for IEEE MASS 2006, IEEE IPDPS2008, and IEEE DCOSS 2009 and was the program cochair for IEEE INFOCOMM 2011. He served as the general chair for IEEE ICDCS 2013. He was an IEEE Computer Society distinguished visitor and the chair for the IEEE Technical Committee on Distributed Processing (TCDP). He received the China Computer Federation (CCF) Overseas Outstanding Achievement Award in 2011. He is currently an ACM distinguished speaker and a fellow of the IEEE.

Yunsheng Wang received the BEng degree in electronic and information engineering from Dalian University of Technology, Dalian, China, in 2007; the MRes degree in telecommunication from University College London, United Kingdom, in 2008; the PhD from Department of Computer and Information Sciences, Temple University, Philadelphia, PA, US, in 2013. He is an assistant professor in the Department of Computer Science in Kettering University, Flint, MI, US. His research interests include various topics in the application and protocols of wireless networks. His current research focuses on the efficient communication in delay tolerant networks and mobile opportunistic social networks.