Social Feature-based Multi-path Routing in Delay Tolerant Networks

Jie Wu and Yunsheng Wang
Department of Computer and Information Sciences
Temple University, Philadelphia, PA 19122

***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. This approach is motivated from several social contact networks, such as the Infocom 2006 trace, 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 multi-path routing. In social feature extraction, we use entropy to extract the m most informative social features to create a feature space (F-space): $(F_1, F_2, ..., F_m)$, 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 multi-path 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 and spray-and-focus routing.**

***Index Terms*—Closeness, delay tolerant networks, entropy, hypercubes, multi-path routing, social features.**

I. Introduction

Delay tolerant networks (DTNs) are characterized by intermittent connectivity and limited network capacity. There exist several different application scenarios: connectivity of developing countries [1], vehicular DTN road communications [2, 3], and social contact networks [4]. In social contact networks, where nodes (individuals) move around and interact at each contact based on their common interests, social features play an important role.

Several social-behavior-based DTN routing schemes have been proposed recently [2, 5–9]. 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 several social contact networks where *people come in contact with each other more frequently if they have more social features in common*.

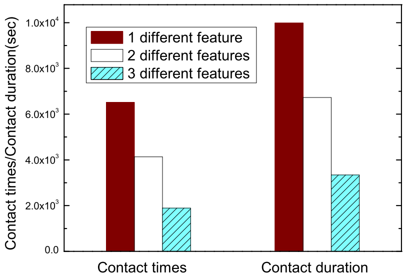

Fig. 1. Comparison of the contacts in the Infocom 2006 trace.

In Fig. 1, we show the difference in contacts when various features differ in the Infocom 2006 conference trace [10]-collected in a period of 337,417 seconds. We can see that the total contact times and contact duration reduce when the social feature difference between two individuals increases. The individuals with only one different feature have about 36.5% more contact times and 32.6% longer contact durations than the individuals with two different features. In [11], Mei et al. found that individuals with similar social features tend to contact more often in DTNs. Hence, we believe that designing a new routing protocol by considering the social features of individuals can improve the performance of DTN routing.

One of the main advantages of using features for routing guidance is its avoidance of state information collection. In addition, *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 (a node in a DTN) is represented by a vector of $(F_1, F_2, ..., F_m)$, 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*, 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 multi-path 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 introduce 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 feature space (i.e., a hypercube node).

To perform efficient multi-path routing, node-disjoint routing is used to which a hypercube-based parallel feature matching process is applied. Feature differences are resolved step-by-step 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 multi-path 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 [8] and spray-and-focus [9], 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 nodes in DTNs) and $M = 2^m$ (number of nodes in the F-space): (1) $M << N$ (i.e., $M = o(N)$), (2) $M = N(M = \Theta(N))$, and (3) $M >> N(M = O(N))$.

The major contributions of our work are as follows:

- We convert the DTN 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 multi-path routing schemes under the hypercube structure: node-disjoint-based and delegation-based.
- We extend multi-path routing to general hypercubes and cube-connected-cubes (CCCs).
- We evaluate the proposed scheme in both synthetic and real traces. The simulation results show the competitive performance of multi-path routing in DTNs.

The remainder of this paper is organized as follows Section II shows the preliminary work. Section III presents the social feature extraction process. Section IV describes two multi-path routing schemes: node-disjoint-based and delegation-based. Section V analyzes these protocols. Section VI discusses two extensions with general hypercubes and cube-connected-cubes (CCCs). Section VII reviews the related work. Section VIII focuses on the simulation and evaluation. We summarize the work in Section IX.

II. Preliminaries

A. Objectives

The objective of this paper is to develop an efficient multi-path routing scheme based on hypercube social feature matching in DTNs. 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 arrival time of a packet; (3) *number of forwardings*: the average number of forwardings of each packet. Efficient routing entails a high delivery rate and low latency with an acceptable number of forwardings and a limited budget in terms of the number of copies of the packet.

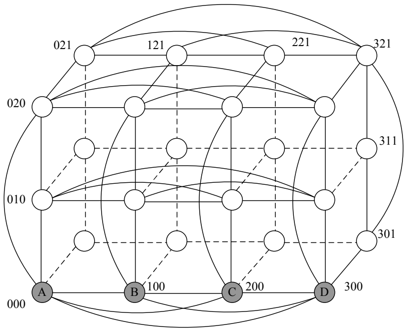

Fig. 2. A 3-dimensional hypercube.

B. 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 a 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 nodes. Fig. 2 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. 2, 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. 2, two groups have a connection if they differ in exactly one feature.

C. 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 node in a hypercube. More specifically, the F-space $(F_1, F_2, ..., F_m)$ is mapped into an m-dimensional hypercube (or simply m-D cube), which consists of $n_1 \times n_2 \times \ldots \times n_m$ nodes. Two nodes, $A = (a_1, a_2, \ldots, a_m)$ and $B = (b_1, b_2, \ldots, b_m)$, 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 and 1. In a binary cube, 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 [15, 16]. These paths are composed of k shortest paths of length k and $m-k$ non-shortest paths of length $k+2$.

In binary cube routing, the relative address of the current node and destination is calculated through XOR on two

TABLE I
NOTATION.

Variable	Description
N	Number of individuals in DTNs
m/m'	Number of key/total features
M	Number of nodes in the F-space, where $M = 2^m$
H_{AB}	Feature distance between A and B
$E(F_i)$	Entropy of feature F_i

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. We will extend this routing scheme by adding shortcuts for fast feature matching in multi-path routing.

D. Delegation Forwarding

In delegation forwarding [13], each node has its estimated distance to the destination which 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 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]. In this paper, we use the feature distance as the quality value of the node to a given destination.

III. FEATURE EXTRACTION

The individuals are characterized by a high dimensional feature profile. However, usually only a small subset of features is important. We use the feature extraction method from data mining [17, 18] to obtain key features.

There are N individuals with m' features, which are denoted as $F_1, F_2, \ldots, F_{m'}$. The goal of our social feature extraction 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:

$$E(F_j) = -\sum_{i=1}^{n_i} p(x_i) log_2 p(x_i), \qquad (j = 1, 2, \ldots, m') \quad (1)$$

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

Table II shows the entropy of each social feature that we obtained from the Infocom 2006 trace [10]: $m = 6$ most informative features out of $m' = 10$ total features.

IV. MULTI-PATH ROUTING

We present a novel *social feature-based multi-path routing* scheme with the objective to reach 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.

TABLE II
ENTROPY OF THE SOCIAL FEATURES IN THE INFOCOM 2006 TRACE.

Social Feature	Entropy
$Affiliation$	4.64
$City$	4.45
$Nationality$	4.11
$Language$	4.11
$Country$	3.59
$Position$	1.37

We propose two special multi-path 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 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 VI. 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 groups to groups 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.

A. Node-Disjoint-based Routing

Initially, the source has m copies of the packet to the destination in k feature distances. As we discussed in Section II-C, there are k shortest paths of length k and $m - k$ non-shortest 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, $C^0, C^1, ..., C^{k-1}$, will create k node-disjoint shortest paths from C:

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

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

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

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

```
1: if $B$ and $D$ are the same group then
2:     Forward the packet to $D$.
3: else
4:     case $i \in d$: $d = d - \{i\}$ and send $\big(C^i, 0\big)$ 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.
7: end if
```

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 $B$.
6: end if
```

shortest path or 1 for a non-shortest 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 vectors, d and d'. d is initialized as $\{1, 2, ..., k\}$, which are different features between the source and destination. d' is $\{k, k+1, ..., m\}$.

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

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 difference 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).

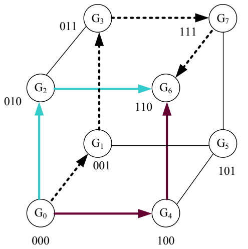

Fig. 3. An example of node-disjoint-based routing with $S = G_0$ and $D = G_6$. Solid directed paths are the shortest paths, and dashed directed paths represent the non-shortest paths.

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 of the packet to $B$.
5:     $L_A \leftarrow L_B$
6: end if
```

B. Delegation-based Routing

Delegation-based routing forwards the copies of a packet only to the individual with 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. 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, level value (L_A), the highest level that A has met so far, is the same as Q_A.

In Algorithm 3, when A, with c copies of the packet, meets another individual 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.

V. Analysis

A. 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 non-shortcut paths are generated based on results in [15, 16], which are k node-disjoint paths of length

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

TABLE III
COMPARISON OF CONTACT FREQUENCY WITH DIFFERENT FEATURE DISTANCE IN THE INFOCOM 2006 TRACE.

Path	Frequency	
(0000, 1000)	$p_1 = 0.196$	P_{11}
(1000, 1100)	$p_2 = 0.183$	P_{22}
(1100, 1110)	$p_3 = 0.192$	P_{33}
(1110, 1111)	$p_4 = 0.188$	P_{44}
(0000, 1100)	$p_{12} = 0.040$	P_{12}
(1000, 1110)	$p_{23} = 0.039$	P_{23}
(1100, 1111)	$p_{34} = 0.041$	P_{34}
(0000, 1110)	$p_{123} = 0.019$	P_{13}
(1000, 1111)	$p_{234} = 0.018$	P_{24}
(0000, 1111)	$p_{1234} = 0.01$	P_{14}
(0000, 1000, 1100)	$p_1p_2 \approx 0.036$	$P_{1..2}$
(1000, 1100, 1110)	$p_2p_3 \approx 0.035$	$P_{2..3}$
(1100, 1110, 1111)	$p_3p_4 \approx 0.036$	$P_{3..4}$
(0000, 1000, 1100, 1110)	$p_1p_2p_3 \approx 0.007$	$P_{1..3}$
(1000, 1100, 1110, 1111)	$p_2p_3p_4 \approx 0.007$	$P_{2..4}$
(0000, 1000, 1100, 1110, 1111)	$p_1p_2p_3p_4 \approx 0.0013$	$P_{1..4}$
(0000, 1000, 1100, 1110)	$p_1p_2p_3 \approx 0.007$	$P'_{1..3}$
(0000, 1100, 1110)	$p_{12}p_3 \approx 0.008$	
(0000, 1000, 1110)	$p_1p_{23} \approx 0.008$	
(0000, 1110)	$p_{123} = 0.019$	
(1000, 1100, 1110, 1111)	$p_2p_3p_4 \approx 0.007$	$P'_{2..4}$
(1000, 1110, 1111)	$p_{23}p_4 \approx 0.007$	
(1000, 1100, 1111)	$p_2p_{34} \approx 0.008$	
(1000, 1111)	$p_{234} = 0.018$	
(0000, 1000, 1100, 1110, 1111)	$p_1p_2p_3p_4 \approx 0.0013$	$P'_{1..4}$
(0000, 1000, 1100, 1111)	$p_1p_2p_{34} \approx 0.0015$	
(0000, 1000, 1110, 1111)	$p_1p_{23}p_4 \approx 0.0014$	
(0000, 1100, 1110, 1111)	$p_{12}p_3p_4 \approx 0.0014$	
(0000, 1000, 1111)	$p_1p_{234} \approx 0.0035$	
(0000, 1110, 1111)	$p_{123}p_4 \approx 0.0036$	
(0000, 1100, 1111)	$p_{12}p_{34} \approx 0.0016$	
(0000, 1111)	$p_{1234} = 0.01$	

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 we discussed in Section IV-A. The non-shortest path can be $(G_0, G_1, G_3, G_7, G_6)$. These three paths are node-disjoint. The shortcuts from the non-shortest 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).

B. Contact Frequency

We use the classic probability theory to draw some observations. We assume that the contact probability is time-independent [19]. We use contact numbers in the most recent time window to estimate contact probability (or precisely, frequency)[2]. More specifically, node S has $p_1, p_2, \ldots, p_m$ contact frequencies to its m neighbors along m dimensions that match the destination features in an m-D cube. $p_{12\ldots k}$ is denoted as the contact frequency between an individual from S and any individual in group D that matches destination features, where S and D differ in k features $1, 2, ..., k$. 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 $\langle 1, 2, ..., k\rangle$.

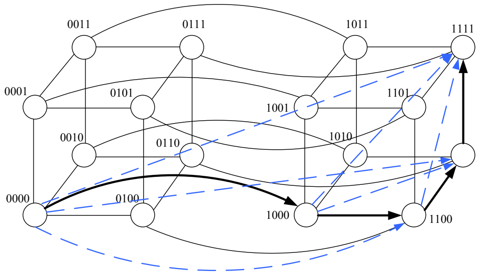

Fig. 4. An example of the composite path from 0000 to 1111, where dashed directed lines are shortcuts.

We conduct an experiment and obtain four social features with the highest entropy values (affiliation, city, nationality, language) from the Infocom 2006 trace to create a 4-D cube, as shown in Fig. 4. 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 includes dimensions $i, i+1, ..., k$. B to D has dimensions $k+1, k+2, ..., j$. Hence, S to D spans dimensions $i, i+1, ..., j$. 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 $P'_{i..j}$, called *composite frequency*, as the frequency of a path from source S to destination D in the following dimension sequence $\langle i, i+1, ..., j\rangle$, including 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 following the dimension sequence. In Fig. 5, we denote the composite frequency from S to D as $P'_{S..D}$. 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} *shortcut frequency*. The shortcut frequency from S to D is denoted as P_{SD}. The *direct frequency*, $P_{i..j} = p_i p_{i+1} \ldots p_j$, corresponds to a direct path in our routing process from S to D, which is denoted as $P_{S..D}$.

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

Proof: Without the loss of generality, we assume the source as S and the destination as D. The corresponding coordinate sequence is $\langle i, i+1, ...j\rangle$, 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. $P'_{k+1..j}$ 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 III records shortcut, direct, and composite frequencies. Destination 1111 is generic and includes all nodes as possible destinations. The binary cube is constructed based on the destination-based partition that 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 III, we have the following two observations that relate to virtual and regular directed triangles:

Observation 1: $P'_{S..D} < P'_{S..B}$, $P'_{S..D} < P'_{B..D}$, and $P'_{S..D} > P'_{S..B} P'_{B..D}$. This means that the composite fre-

[2] Although contact duration is also important, results in [20] and Figure 1 show that there are high correlation coefficients of duration and frequency in many traces; we simply consider only frequency in this paper.

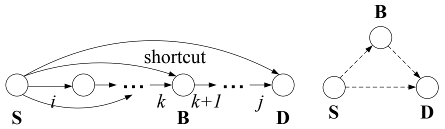

Fig. 5. An illustration of contact frquency.

quency 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.

Based on Observation 2, we can use induction to prove that $P_{SD} > P_{S..D}$, 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 delivery in terms of a smaller number of forwardings and a shorter delivery time. For given source S and destination D, we conjecture that direct frequency $P_{S..D}$ is a lower bound of shortcut frequency P_{SD}, and composite frequency $P'_{S..D}$ is an upper bound of P_{SD}. In our synthetic trace simulation, we will use these two bounds to generate shortcuts.

VI. Extensions

A. General Hypercubes

In the previous sections, we discussed multi-path 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 dark color, where $*$ is a wild card for 0, 1, 2, and 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 be in contact in the near future (i.e., a low contact frequency). In Fig. 2, we assume that A holds a packet to 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 2-hop contact history at each node, without resorting to global contact information.

In order to control the hop-count, we can modify the 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 has a contact with B that 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 IV-A), which will hold the packet until meeting with the destination at a particular dimension.

B. Cube-Connected-Cubes (CCCs)

When the initial number of copies of the packet is less than the number of dimensions, we can use the cube-connected-cubes (CCCs) 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 compared to the basic scheme.

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

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

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 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. Hence, $I(X;Y)$ 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*).

VII. Related Work

The simplest DTN routing scheme is flooding or epidemic routing [21]. To control the copies of the packet, Lee et al. introduce 2-hop routing [22], where the source gives a copy to relay nodes, each of which holds the packet until it contacts the destination. In [8, 9], two multi-copy 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 multi-hop unless the current node has one copy left. Spray-and-focus goes further to allow multi-hop even when there is one copy. Multi-hop is based on a quality metric, like delegation forwarding [13]. Our approach differs in that the split is proportional to the quality of two encountered nodes.

Many social-behavior-based approaches are proposed [2, 5–7, 13, 14] which 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 DTN, although some predictive models [19]

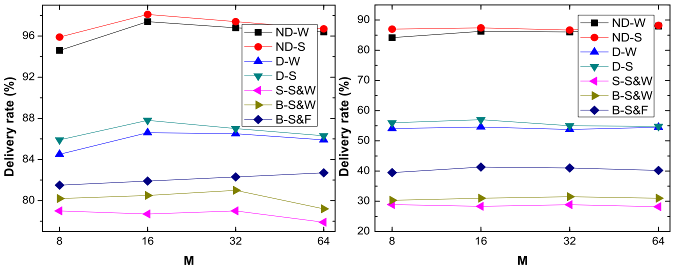

Fig. 6. Comparing the delivery rate in the real trace: (left: L): 20 packets and (right: R): 100 packets.

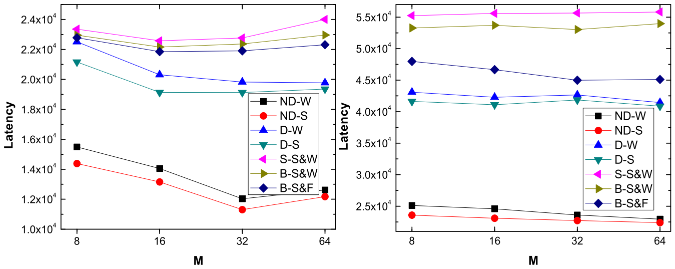

Fig. 7. Comparing the latency in the real trace: (L): 20 packets and (R): 100 packets.

can be applied. In this paper, we introduce the internal feature, which does not incur any cost in state information collection, to guide the routing process. Our approach does require a simple pre-processing in terms of feature selection. Among existing feature selection methods [12, 17, 18], Shannon entropy and mutual information based information theoretic filter (ITF) [12] have received significant attention.

The applications of hypercubes have been initially studied in parallel and distributed computing [15, 16]. There have been some recent works on hypercube routing in wireless networks [23–25]. Our approach utilizes the advantage of hypercube properties so that multiple paths are guaranteed to be node-disjoint. Note that such a property is absent in existing DTN multi-path routing protocols.

VIII. Simulation

We compare the performance of the proposed multi-path routing scheme with several existing ones, including spray-and-wait and spray-and-focus, in Matlab, using both real and synthetic traces. The simulation is grouped into the following categories. (1) Varying node density: we show that multi-path routing is robust under different node density conditions. (2) The importance of the non-shortest path in node-disjoint-based routing: we illustrate the impact of the non-shortest path in node-disjoint multiple path routing. (3) Comparing the extensions: we compare different methods in two extensions: general hypercubes and cube-connected-cubes (CCCs).

A. *Simulation Methods and Setting*

We implement and compare seven routing schemes in the simulation. 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.

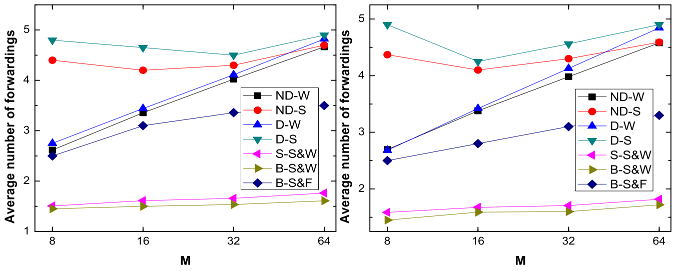

Fig. 8. Comparing the number of forwardings in the real trace: (L): 20 packets; (R): 100 packets.

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) **Source spray-and-wait** (**S-S&W**): *Spray* phase: the source forwards copies to the first m distinct nodes 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.

6) **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.

7) **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.

1) Real trace: we use the Infocom 2006 trace [5, 10] in our simulation. This 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. Firstly, 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 six social features from the original dataset: nationality, language, affiliation, position, city, and country.

2) 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 that is more than one feature distance away from A, the shortcut frequency P_{AB} is randomly selected between its lower bound ($P_{A..B}$, direct frequency) and its upper bound ($P^{'}_{A..B}$, 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.

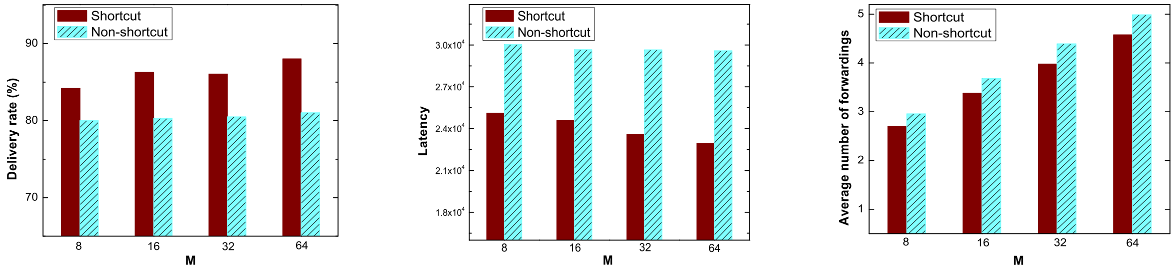

Fig. 9. Comparing *shortcut* and *non-shortcut* with 100 packets in the real trace: (L): delivery rate, ($Center$): latency, and (R): number of forwardings.

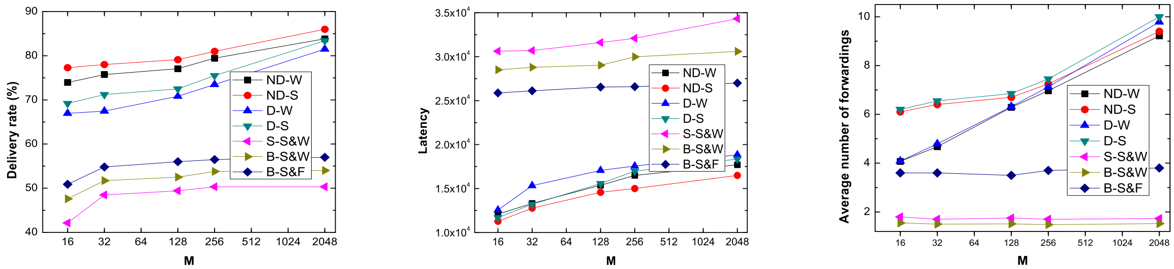

Fig. 10. Comparing with 100 packets in the synthetic trace: (L): delivery rate, ($Center$): latency, and (R): number of forwardings.

B. Simulation Results

1) Varying node density: in this section, we compare the performance of multi-path routing with spray-and-wait and spray-and-focus with varying node densities. In the real trace, as we selected six social features, we set the number of nodes ($M = 2^m$) in the F-space to 8, 16, 32, 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 5 time slots.

From Figs. 6, 7, and 8, we can see that node-disjoint-based routing has the highest delivery rate and the lowest latency among all in the real trace. Delegation-based routing performs better than 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 5.5% compared to delegation-based routing. out of all multi-path routing schemes, spray-at-destination increases the delivery rate by 2% and reduces the latency by 5% 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%, cut latency by 15%, and reduce the number of forwardings by 8%, as seen in Fig. 9.

As the results in the 20 and 100 packets conditions show the same trend, we only report results for the 100 packets condition in the synthetic trace. Node-disjoint-based routing has the highest delivery rate and lowest latency in Fig. 10. Multi-path routing increases the average number of forwardings compared with spray-and-wait and spray-and-focus. Using shortcuts improves the overall performance in node-disjoint-based routing.

2) Non-shortest path: in this section, our simulation demonstrates the importance of the non-shortest 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 non-shortest paths that reach the destination before any shortest path also increases in Fig. 11. This is expected as there are $m - k$ non-shortest paths and k shortest paths in a given m-D cube.

3) Extensions: in general hypercubes, we compare the three methods that were discussed in Section VI-A: *general forward*, *2-hop general forward*, and *general wait*. In Fig. 12, 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 VI-B: *entropy-based braiding*, *random braiding*, and *parallel path routing* under a given number of copies. We set six social features under both traces. In Fig. 13, we can see that entropy-based braiding has the best performance in terms of the delivery rate and the latency. The performance decreases noticeably when using parallel path routing.

C. Summary of Simulation

Our simulation concludes that although multi-path routing increases the number of forwardings compared with spray-and-wait and spray-and-focus, 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 non-shortest path also plays an important role, especially when the number of node-disjoint paths is 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.

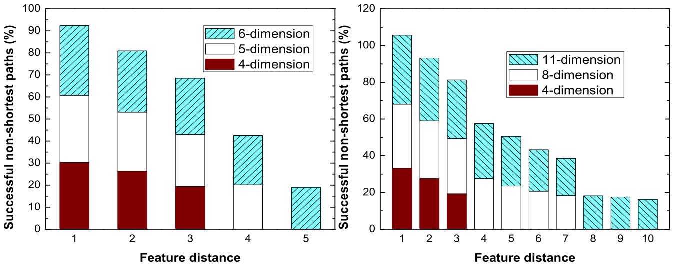

Fig. 11. Comparing in the percentage of the successful non-shortest path: (L): real trace and (R): synthetic trace.

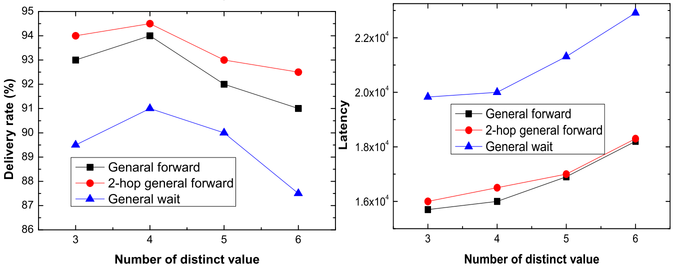

Fig. 12. Comparing in general hypercubes in the real trace: (L): delivery rate and (R): latency.

Simulation results of two extensions show that the proposed multi-path routing scheme can also achieve a competitive performance in general hypercubes, which is more practical in reality. Multi-path 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 multi-path routing can be effective under different conditions in DTN routing.

IX. Conclusion

In this paper, we proposed a social feature-based multi-path routing scheme in DTNs. Our scheme has two parts: social feature extraction and multi-path routing. We used entropy to extract the most informative social features to build an m-dimensional hypercube. In multi-path 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 multi-copy model, which uses a feature-distance-based metric as the copy forwarding decision metric. Trace-driven simulation results showed that our proposed multi-path routing scheme performs better than both spray-and-wait and spray-and-focus. We believe that the social features will play an important role in routing in social contact networks. Our future work will include more experiments on different social network traces to validate our observations. We also plan to study more sophisticated routing schemes in general hypercubes.

References

[1] A. Pentland, R. Fletcher, and A. Hasson, “Daknet: rethinking connectivity in developing nations,” *Computer*, vol. 37, no. 1, pp. 78–83, 2004.

[2] J. Burgess, B. Gallagher, D. Jensen, and B. N. Levine, “Maxprop: Routing for vehicle-based disruption-tolerant networks,” in *Proc. of IEEE INFOCOM*, 2006.

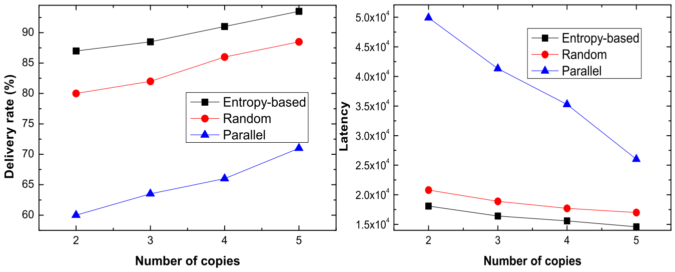

Fig. 13. Comparing in CCCs in the real trace: (L): delivery rate and (R): latency.

[3] J. Ott and D. Kutscher, “Drive-thru Internet: IEEE 802.11b for “automobile” users,” in *Proc. of IEEE INFOCOM*, 2004.

[4] J. Scott, J. Crowcroft, P. Hui, and C. Diot, “Haggle: a networking architecture designed around mobile users,” in *Proc. of WONS*, 2006.

[5] A. Chaintreau, P. Hui, J. Crowcroft, C. Diot, R. Gass, and J. Scott, “Impact of human mobility on opportunistic forwarding algorithms,” *IEEE Transactions on Mobile Computing*, vol. 6, pp. 606–620, 2007.

[6] J. Wu and Y. Wang, “A non-replication multicasting scheme in delay tolerant networks,” in *Proc. of IEEE MASS*, 2010.

[7] G. S. Thakur, A. Helmy, and W.-J. Hsu, “Similarity analysis and modeling in mobile societies: the missing link,” in *Proc. of ACM CHANTS*, 2010.

[8] T. Spyropoulos, K. Psounis, and C. S. Raghavendra, “Spray and wait: an efficient routing scheme for intermittently connected mobile networks,” in *Proc. of ACM WDTN*, 2005.

[9] T. Spyropoulos, K. Psounis, and C. S. Raghavendra, “Spray and focus: Efficient mobility-assisted routing for heterogeneous and correlated mobility,” in *Proc. of IEEE PERCOMW*, 2007.

[10] J. Scott, R. Gass, J. Crowcroft, P. Hui, C. Diot, and A. Chaintreau, “CRAWDAD trace cambridge/haggle/imote/infocom2006 (v. 2009-05-29).” Downloaded from http://crawdad.cs.dartmouth.edu/ cambridge/haggle/imote/infocom2006, May 2009.

[11] A. Mei, G. Morabito, P. Santi, and J. Stefa, “Social-aware stateless forwarding in pocket switched networks,” in *Proc. of IEEE Infocom*, 2011.

[12] C. Shannon, N. Petigara, and S. Seshasai, “A mathematical theory of communication,” *Bell System Technical Journal*, vol. 27, pp. 379–423, 1948.

[13] V. Erramilli, M. Crovella, A. Chaintreau, and C. Diot, “Delegation forwarding,” in *Proc. of ACM MobiHoc*, 2008.

[14] Y. Wang, X. Li, and J. Wu, “Multicasting in delay tolerant networks: Delegation forwarding,” in *Proc. of IEEE GLOBECOM*, 2010.

[15] J. Wu, *Distributed System Design*. CRC Press, 1998.

[16] Y. Saad and M. Schultz, “Topological properties of hypercubes,” *IEEE Transactions on Computers*, vol. 37, no. 7, pp. 867 –872, 1988.

[17] A. Jain and D. Zongker, “Feature selection: Evaluation, application, and small sample performance,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 19, pp. 153–158, 1997.

[18] I. Guyon and A. Elisseeff, “An introduction to variable and feature selection,” *J. Mach. Learn. Res.*, vol. 3, pp. 1157–1182, 2003.

[19] M. C. Gonzalez, C. A. Hidalgo, and A.-L. Barabasi, “Understanding individual human mobility patterns,” *Nature*, vol. 453, no. 7196, pp. 779–782, 2008.

[20] T. Hossmann, T. Spyropoulos, and F. Legendre, “Putting contacts into context: Mobility modeling beyond inter-contact times,” in *Proc. of ACM MobiHoc*, 2011.

[21] A. Vahdat and D. Becker, “Epidemic routing for partially connected ad hoc networks,” *Technical Report, Dept. of Computer Science, Duke University*, 2000.

[22] U. Lee, S. Y. Oh, K.-W. Lee, and M. Gerla, “Relaycast: Scalable multicast routing in delay tolerant networks,” in *Proc. of IEEE ICNP*, 2008.

[23] R. Manoharan and P. Thambidurai, “Hypercube based team multicast routing protocol for mobile ad hoc networks,” in *Proc. of ICIT*, 2006.

[24] C.-T. Chang, C.-Y. Chang, and J.-P. Sheu, “Bluecube: constructing a hypercube parallel computing and communication environment over bluetooth radio system,” in *Proc. of ICPP*, 2003.

[25] H. Huo, W. Shen, Y. Xu, and H. Zhang, “Virtual hypercube routing in wireless sensor networks for health care systems,” in *Proc. of ICFIN*, 2009.