Evaluating Mobility Pattern Space Routing for DTNs

Jérémie Leguay,*† Timur Friedman,* Vania Conan†
* Université Pierre et Marie Curie, Laboratoire LiP6–CNRS
† Thales Communications

***Abstract*— Because a delay tolerant network (DTN) can often be partitioned, routing is a challenge. However, routing benefits considerably if one can take advantage of knowledge concerning node mobility. This paper addresses this problem with a generic algorithm based on the use of a high-dimensional Euclidean space, that we call MobySpace, constructed upon nodes' mobility patterns. We provide here an analysis and a large scale evaluation of this routing scheme in the context of ambient networking by replaying real mobility traces. The specific MobySpace evaluated is based on the frequency of visits of nodes to each possible location. We show that routing based on MobySpace can achieve good performance compared to that of a number of standard algorithms, especially for nodes that are present in the network a large portion of the time. We determine that the degree of homogeneity of node mobility patterns has a high impact on routing. And finally, we study the ability of nodes to learn their own mobility patterns.**

I. Introduction

This paper addresses the problem of routing in delay tolerant networks (DTNs) [1]. It evaluates a scheme that we proposed in earlier work [2] that turns the problem of DTN routing into a problem of routing in a virtual space defined by the mobility patterns of nodes. The earlier work tested the scheme with an entirely artificial scenario. By driving simulations with real mobility traces, in this paper we validate this routing scheme in the context of ambient networks. This paper also studies a number of important factors, such as the degree of homogeneity in the mobility of nodes, that impact routing performance. Finally, the paper examines the ability of nodes to learn their own mobility, which is important for the feasibility of such a scheme.

In one common DTN scenario, like the one we consider in this paper, nodes are mobile and have wireless networking capabilities. They are able to communicate with each other only when they are within transmission range. The network suffers from frequent connectivity disruptions, making the topology intermittently and partially connected. This means that there is a very low probability that an end-to-end path exists between a given pair of nodes at a given time. End-to-end paths can exist temporarily, or may sometimes never exist, with only partial paths emerging. Due to these disruptions, regular ad-hoc networking approaches to routing and transport do not work, and new solutions must be proposed.

The Delay Tolerant Network Research Group (DTNRG) [3] has proposed an architecture [4] to support messaging that may be used by delay tolerant applications in such a context. The architecture consists mainly of the addition of an overlay, called the bundle layer, above a network's transport layer. Messages transferred in DTNs are called bundles. They are transferred in an atomic fashion between nodes using a transport protocol that ensures node-to-node reliability. These messages can be of any size. Nodes are assumed to have buffers in which they can store the bundles.

Routing is one of the very challenging open issues in DTNs, as mentioned by Jain et al. [5]. Indeed, since the network suffers from connectivity problems, MANET [6] routing algorithms such as OLSR, based on the spreading of control information, or AODV, which is on-demand, fail to achieve routing. Different approaches have to be found.

The problem of routing in DTNs is not trivial. Epidemic routing [7], studied by Vahdat and Becker, is a possible solution when nothing is known about the behavior of nodes. Since it leads to buffer overloads and inefficient use of transmission media, one would prefer to limit bundle duplication and instead use routing heuristics that can take advantage of the context. To move in such a direction, the DTN architecture defines several types of contacts: *scheduled*, *opportunistic*, and *predicted*. *Scheduled* contacts can exist, for instance, between a base station somewhere on earth and a low earth orbiting relay satellite. *Opportunistic* contacts are created simply by the presence of two entities at the same place, in a meeting that was neither scheduled nor predicted. Finally, *predicted* contacts are also not scheduled, but predictions of their existence can be made by analyzing previous observations.

The study presented in this paper relies also on contacts that can be characterized as predicted, but the underlying concept is a more generic abstraction compared to previous work, being able to capture the interesting properties of major mobility patterns for routing.

The main contribution of this paper is the validation of a routing scheme for DTNs that uses the formalism of a high-dimensional Euclidean space based on nodes' mobility patterns. We show the feasibility of this concept through an example in which each dimension represents the frequency with which a node is be found in a particular location. We conduct simulations by replaying mobility traces to analyse the feasibility and comparative performance of such a scheme.

The rest of this paper is structured as follows. Sec. II describes the general concept of the mobility pattern based routing scheme, called the MobySpace. Sec. III presents the specific MobySpace we have considered for the evaluation. Sec. IV presents the simulation results and Sec. V a feasibility study. Sec. VI provides an overview of related work concerning routing in DTNs. Sec. VII concludes the paper, discussing directions for future work.

II. MobySpace: a Mobility Pattern Space

Two people having similar mobility patterns are more likely to meet each other, thus to be able to communicate. Based on this simple principle, our proposition [2] is to use the formalism of a Euclidean virtual space, that we call a *MobySpace*, as a tool to help nodes make routing decisions. These decisions rely on the notion that a node is a good candidate for taking custody of a bundle if it has a mobility pattern similar to that of the bundle's destination. Routing is done by forwarding bundles toward nodes that have mobility patterns that are more and more similar to the mobility pattern of the destination. Since in the MobySpace, the mobility pattern of a node provides its coordinates, called its *MobyPoint*, routing is done by forwarding bundles toward nodes that have their MobyPoint closer and closer to the MobyPoint of the destination. Note that the MobySpace is purely a virtual expression of the mobility patterns, and as such does not express the geographic coordinates of the nodes (GPS or otherwise). It cannot be used for geographic routing.

In this section, we describe manners in which mobility patterns can be characterized and the ways these patterns can be managed by the nodes, and we discuss possible limits and issues surrounding the overall concept.

A. Mobility pattern characterization

Since the mobility pattern of a node provides its coordinates in the MobySpace, the way in which these patterns are characterized determines the way the virtual space is constructed.

The way in which mobility patterns are characterized determines the number and the type of the dimensions of the specific MobySpace. It bears repeating that the MobySpace is not a physical space: each MobyPoint summarizes some characteristics of a node's mobility pattern. Many methods could be employed to describe a mobility pattern, but some requirements must be satisfied. We want mobility patterns to be simple to measure in order to keep them computationally inexpensive and to reduce the overhead associated with exchanging them between nodes. Furthermore, they must be relevant to routing, by helping nodes to take efficient routing decisions.

A mobility pattern could be based, for instance, upon historic information regarding contacts that the node has already had. A recent study [8] by Hui et al. has shown the interest of such mobility patterns. It highlights that contacts between people at the Infocom 2005 conference follow power-laws in terms of their duration. If we want to route a bundle from one node to another, we have an interest in taking the unevenness of the distribution into consideration. Intuitively, it could be very efficient to transmit a bundle to a relay that frequently encounters the destination. A MobySpace based on this kind of pattern would be as follows. Each possible contact is an axis, and the distance along that axis indicates an estimate of the probability of contact. Two nodes that have a similar set of contacts that they see with similar frequencies are close in this space, whereas nodes that have very different sets of contacts, or that see the same contacts but with very different frequencies, are far from each other. It seems reasonable that one would wish to pass a bundle to a node that is as close as possible to the destination in this space, because this should improve the probability that it will eventually reach the destination.

We might wish to consider an alternative space in which there is a more limited number of axes. If nodes' visits to particular locations can be tracked, then the mobility pattern of a node can be described by its visits to these locations. In this scenario, each axis represents a location, and the distance along the axis represents an estimate of the probability of finding a node at that location. We can imagine that nodes that have similar probabilities of visiting a similar set of locations are more likely to encounter each other than nodes that are very different in these respects.

Prior work [5] has demonstrated the interest of capturing temporal information as well. It is well known that network usage patterns follow diurnal and weekly cycles. We could easily imagine two nodes that visit the same locations with the same frequencies, but on different days of the week. This kind of desynchronisation could arise for instance in a campus at the scale of the hour if we consider two users each having a course in the same lecture hall the same day but not at the same time. Even so, it still might make sense to route to one node in order to reach the other, especially if there is a relay node at the commonly visited location. We can imagine ways in which the dimensional representation could capture temporal information as well. For instance, visit patterns could be translated into the frequential domain (by which we mean cyclic frequencies). A node's visits to a location could be represented by a point on a cyclic frequency axis, capturing the dominant cyclic frequency of visits, and a point on a phase axis, as well as a point on the axis already described, that represents the overall frequency (in terms of number of visits) of visiting the location.

The evaluation and the comparison of the different kinds of mobility patterns are kept for further studies. In Secs. III and IV, we test a MobySpace based on the frequency with which nodes find themselves in certain locations

B. Mobility pattern acquisition

A node in the network has to determine its coordinates in the MobySpace, the ones of the nodes it meets, and the ones of the destinations of the bundles it carries, in order to take appropriate routing decisions. Two problems arise: how does a node learn its own mobility pattern, and how does a node learn those of the other nodes?

There are several ways a node can learn its own mobility pattern. First, a node can learn its mobility pattern by observing its environment, e.g., by studying its contacts or its frequency of visits to different locations. If the node requires information about its current position, we can assume that particular tags are attached to each location. Alternatively, we can imagine that nodes are able to interrogate an exiting infrastructure to obtain these patterns. This infrastructure would act as a passive monitoring tool for pattern calculation. The system can be accessible anywhere in a wireless or in a wired fashion or it can be located at certain places.

Similarly, there are several ways that a node can learn the mobility patterns of other nodes. These mobility patterns could be spread in an epidemic fashion. Nodes could also spread just the most significant coordinates of their mobility patterns to reduce buffer occupancy and network resource consumption (an idea that we explore in Sec. V-B). We can also imagine that nodes drop off their mobility patterns in repositories placed at strategic locations, and at the same time they update their knowledge with the content available at the repositories. We leave the study of possible solutions to future work.

C. Mobility pattern usage

As mentioned in the introduction, the mobility pattern of a node determines its coordinates in the MobySpace, i.e., the position of its associated MobyPoint. The basic idea is that bundles are forwarded to nodes having mobility patterns more and more similar to that of the destination. Formally, let U be the set of all nodes and L be the set of all locations. The MobyPoint for a node $k \in U$ is a point in an n-dimensional space, where $n = |L|$. We write $m_k = (c_{1_k}, ..., c_{n_k})$ for the MobyPoint of node k. The distance between two MobyPoints is written $d(m_i,m_j)$.

At a point in time, t, the node k will have a set of directly connected neighbors, which we write as $W_k(t) \subseteq U$. $W_k^+(t) = W_k(t) \cup \{k\}$ is the augmented neighborhood that contains k. MobySpace routing consists of either choosing one of these neighbors to receive the bundle or deciding to keep the bundle. The routing function, which we call f, chooses the neighbor that is closest to the destination, b. The decision for node k when sending a bundle to b is taken by applying the function f:

$$f(W_k^+(t),b) = \begin{cases} b \text{ if } b \subset W_k(t), \text{ else} \\ i \in W_k^+(t) : d(m_i,m_b) = \min_{j \in W_k^+(t)} d(m_j,m_b) \end{cases} \quad (1)$$

The choice of the distance function d used in the routing decision process is important. One straightforward choice is Euclidean distance. Examples of other distance functions can be found in [2]. We leave their comparison to future work.

D. Possible limits and issues

DTN routing in a contact space or a mobility space is based on the assumption that there will be regularities in the contacts that nodes have, or in their choices of locations to visit. There is always the possibility that we may encounter mobility patterns similar to the ones observed with random mobility models. The efficiency of the virtual space as a tool may be limited if nodes change their habits too rapidly.

Some problems could occur even if nodes have well defined mobility patterns, but their existence and nature may depend on the particularities of the space. For instance, in the Euclidean space, a bundle may reach a local maximum if a node has a mobility pattern that is the most similar in the local neighborhood to the destination node's mobility pattern, but is not sufficient for one reason or another to achieve the delivery. In the second type of space, where each dimension represents a location, it can happen if nodes visit similar places, but for timing reasons, such as being on opposite diurnal cycles, they never meet. This kind of user behavior has been observed by Henderson et al. [9] and Hui et al. [8].

The Euclidean spaces that we have discussed here are finite in terms of number of dimensions, but in practice the number of dimensions might be unbounded. This is the case, for instance, in the space we use as a case study in Sec. III. Additional mechanisms must be found to allow this.

Finally, the routing scheme presented here is based on each node forwarding just a single copy of a bundle, which may be a problem in case of node failure or nodes leaving the system for extended periods of time. One may wish to introduce some redundancy into MobySpace routing. For instance, a node can be allowed to transmit a bundle up to T times if, after the first transmission, it meets other nodes having mobility patterns even more similar to that of the destination within a period P.

III. Frequency of visit based MobySpace

To evaluate the routing scheme based on MobySpace, we use a simple kind of space that we describe in the first part of this section. The second part introduces the mobility data that we replay for the evaluation.

A. Description

The frequency of visit based MobySpace we evaluate works as follow. Over a defined time interval, each node spends some portion (possibly zero) of that time at each of the n locations. This set of quantities is a node's mobility pattern, and is described by a MobyPoint in an n dimensional MobySpace. If we consider the frequencies to be reliable estimates of future probabilities, the coordinate of a node along the axis k is its probability of visit for the location k. All MobyPoints in a given MobySpace lie in a hyperplane, since we have:

$$\text{for any point } m_i = (c_{1_i}, ..., c_{n_i}), \sum_{k=1}^{n} c_{k_i} = 1 \quad (2)$$

Recent studies of the mobility of students in a campus [10], [9] or of corporate users [11], equipped with PDAs or laptops able to be connected to wireless access networks, show that they follow common mobility patterns. They show that significant aspects of the behavior can be characterized by power law distributions. Specifically, the session durations and the frequencies of the places visited by users follow power laws. This means that users typically visit a few access points frequently while visiting the others rarely, and that users may stay at few locations for long periods while visiting the others for very short periods. Henderson et al. observed [9] that 50%

of users studied spent 62% of their time attached to a single access point, and this proportion decreased exponentially.

Regarding the distance function, we choose a straightforward one, the Euclidean distance:

$$d(m_i, m_j) = \sqrt{\sum_{k=1}^{n} \left(c_{k_i} - c_{k_j}\right)^2} \tag{3}$$

B. Real mobility data used

There has been considerable growth in the number of small devices people carry every day, such as cell phones, PDAs, music players, and game consoles. The variety of their different networking capabilities allows us to envisage new applications, such as distributed databases, content delivery systems, or self organizing peer to peer networks. We can imagine that such spontaneous and autonomous networks spring up around the movement of people in campus or corporate environments. Contextual applications, services, or basic applications like text messaging could take advantage of such an infrastructure. These scenarios are studied within the framework of delay tolerant networks.

For the purpose of this study, we sought real mobility traces that resembled what one might find in an ambient network environment. Since there are very few traces of this kind, we chose data that tracks mobile users in a campus setting. We used the mobility data collected on the Wi-Fi campus network of Dartmouth College [9]. Jones et al. [12] have recently used the traces in a similar way. The Dartmouth data is the most extensive data collection available that covers a large wireless access network. The network is composed of about 550 access points (APs), the number of different wireless cards (MAC addresses) seen by the network is about 13,000 and the data have been collected between the years 2001 and 2004. The network covers the college's academic buildings, the library, the sport infrastructures, the administrative buildings and the student residences. Users are equipped with devices such as PDAs, laptops, and phones that support voice over IP (VoIP). The majority of the end users are students, who make intensive use of the network, especially since many of them are required to own a laptop. Fig. 1 illustrates the usage levels by showing the evolution of the number of active nodes in the network per day.

The data we analysed track users' sessions in the wireless network. These data have been pre-processed by Song et al. in their prior work [13] on mobility prediction. The traces show the time at which a node associates or dissociates from an access point. Data were collected by a central server with the Syslog [14] protocol. It could happen that a node does not send a dissociation message, or that a Syslog UDP message is lost, in which case a session is considered finished after 30 minutes of inactivity.

For our study, each access point represents a location. We assume that two nodes (represented as networking cards in the data) are assumed to be able to communicate with a low range device (using Bluetooth for instance), if they are attached at the same time to the same AP. This assumption is somewhat artificial as nodes that are attached to two different APs that are close to each other might be able to communicate directly. Similarly, two nodes connected to the same AP might be out of range of each other. Nonetheless, this is the best approximation we can make with the data at hand.

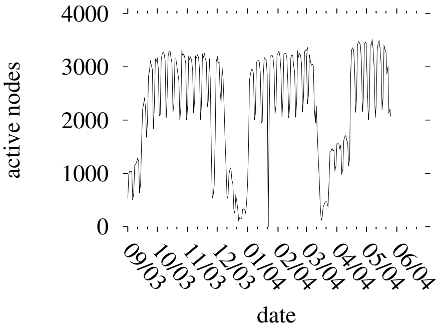


Fig. 1. Number of actives users per day (from 1 September 2003 to 1 June 2004).

Though at present there are few extensive and publicly available data sets that offer mobility traces related to DTN scenarios, the situation should improve shortly. We expect for instance to evaluate MobySpace with the help of data sets like the one acquired with iMotes [8] within Intel's Haggle project or the one of the Reality Mining project [15] captured with mobile phones. These data sets provide information about fine-grained interactions between people instead of their co-presence in a coarse-grained area. Traces such as the one from the UMassDieselNet [16] project with mobile nodes on buses may also be of interest.

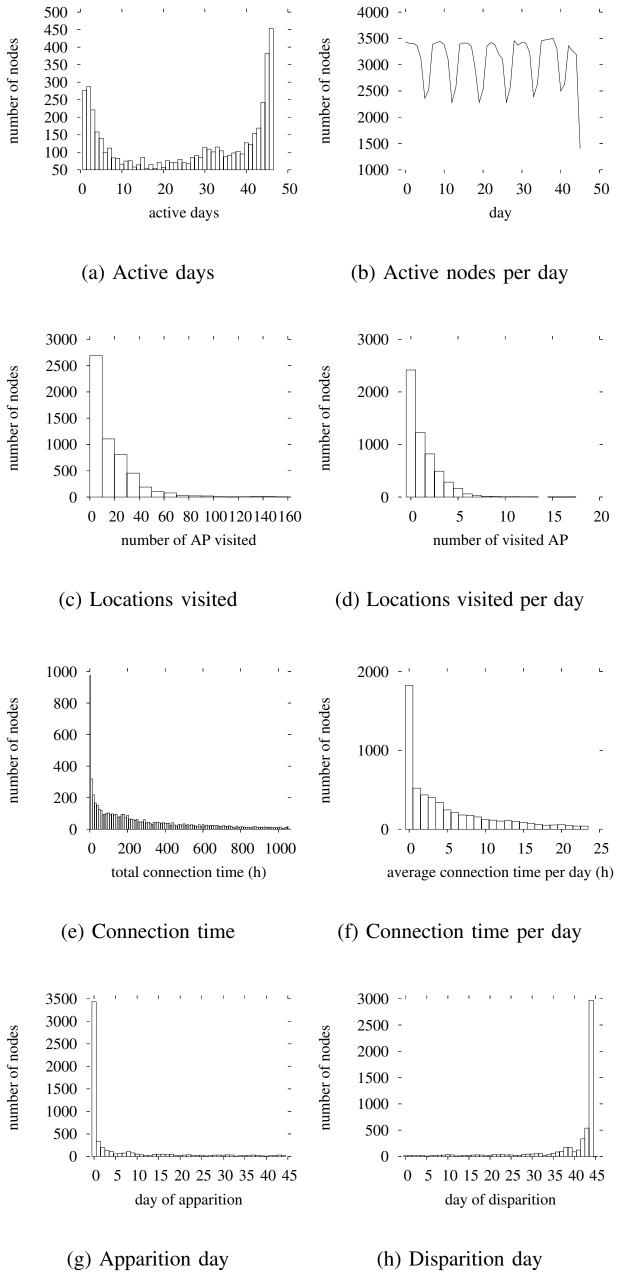
IV. Simulation results

This section presents the manner in which we evaluated the routing scheme that uses a frequency of visit based MobySpace, and the results we obtained. Since we performed the simulations using a subset of 45 days of mobility data, we first describe the properties of the traces collected during that period.

A. Mobility traces

We replayed the mobility traces inferred from Dartmouth data between January 26th 2004 and March 11th 2004. Fig. 2 shows distributions that characterize users' behavior within this period. We choose that period because, as shown in Fig. 2(b), users make an intensive and regular use of the network. As shown by Fig. 1, this period is between Christmas and the spring break. In this period, we have observed a total of 5,545 active users who have visited 536 locations.

Users are mobile. They visit on average 16.66 locations in the period (see Fig. 2(c)) and 1.75 locations per day (see Fig. 2(d)). The distributions of the number of locations visited by the nodes during the period and per day follow heavy tailed distributions. This means that the majority of users have a low level of mobility while some users are very mobile. Users with a low mobility level regarding the number of locations they



(a) Active days

(b) Active nodes per day

(c) Locations visited

(d) Locations visited per day

(e) Connection time

(f) Connection time per day

(g) Apparition day

(h) Disparition day

Fig. 2. Statistics on the data set used for the simulations.

visit may either be users that are not very present in the data or users that stay in one place, as in students who keep their laptop connected in their room at the student residence.

The network usage displays a number of regularities. Fig. 2(b) shows the evolution of the number of active users per day. It highlights the existence of regular weekly cycles and a fairly constant number of active users: 2,901 users per day on average. Regularity is a desirable property for this study because we wish to evaluate the MobySpace based routing scheme in a context were people move in their usual everyday environment having a number of constant habits.

Users make intensive use of the network. The mean presence time for the period is 243 hours and is 5.18 hours per day (see Fig. 2(f)). Having users with a high level of presence is important but not sufficient. That presence must also be distributed over time. Thus, we analyse the distributions of the apparition and disparition days of users, and their total number of days of presence. Fig. 2(g) and Fig. 2(h) show that apparitions and disparitions generally occur close to the limits of the period. This means that the probability that a node will disappear close to the beginning of the simulation is low. Similarly, the probability that a node will appear for the first time close to the end of the period is low. Looking at the distribution of the number of days that users are present (Fig. 2(a)), 25.48 days on average, it appears that either users make an intensive use of their laptop or PDA, or they seldom use it, but a majority of users make an intensive use of the network since 50% of users are present more than 30 days.

B. Methodology

We have implemented a stand alone simulator to evaluate the routing scheme. This simulator only implements the transport and network layers and it makes simple assumptions regarding lower layers, allowing infinite bandwidth between nodes and contention free access to the medium. Nodes are also supposed to have infinite buffers and to have inherent knowledge of all other nodes' mobility patterns. Because in ambient networks, nodes may have limited resources and capabilities, routing solutions should also be evaluated with limited buffers and more realistic models for the MAC and physical layers. One way in which we address the problem of limited resources is to examine, in Sec. V-B, the possibility of limiting the amount of information that is sent regarding nodes' mobility patterns. However, our aim here is principally to validate the idea of MobySpace routing. We leave to future work a detailed study of the modifications that would be required to accommodate resource limitations. Note also that we study the question of learning mobility patterns in Sec. V.

We compare the performance of MobySpace routing against the following:

- *Epidemic routing*: This is described by Vahdat and Becker [7]: Each time two nodes meet, they exchange their bundles. The major interest of this algorithm is that it provides the optimum path and thus the minimum bundle delay. We use it here as a lower bound. This algorithm can be also seen as the extension of Dijkstra's shortest path algorithm proposed by Jain et al. [5] that takes into account time-varying edge weights. In practice, epidemic routing suffers from high buffer occupancy and high bandwidth utilization.
- *Opportunistic routing*: A node waits to meet the destination in order to transfer its bundle. The main advantage of this method is that it involves only one transmission per bundle. Bundle delivery relies just on the mobility of nodes and their contact opportunities.
- *Random routing*: There are many ways to define a random routing algorithm. In order to design one that acts similarly to the MobySpace based routing scheme, we

attribute for each destination node j a preference list l_j, which is a randomly ordered list of all of the nodes. When a node has a bundle destined to j, it sends that bundle to the most preferred neighbor on the preference list l_j. If the most preferred neighbor has a lower preference than the current node, the bundle is not forwarded. This mechanism avoids loops by construction.

- *Hot potato routing*: When a node is at a location and the bundle's destination in not there, the node transfers the bundle to a neighbor chosen at random. We have added a rule to avoid local loops: a node can only handle a bundle one time per location visit.

We will refer to these schemes by the following names: *Epidemic*, using Epidemic routing; *Opportunistic*, using Opportunistic routing; *Random*, using Random routing; *Potato*, using Hot potato routing, and *MobySpace*, using the routing scheme that relies on the MobySpace.

All the scenarios share common parameters that can be found in Table I. We considered the whole set of 536 locations that were visited over the course of the 45 days of data. The virtual space used for routing thus has 536 dimensions. Due to the difficulty of running simulations with the totality of the 5,545 nodes, especially with Epidemic, for which computation explodes with the number of nodes and the number of bundles generated, we used a sampling method. We have defined two kinds of users: *active*, which generate traffic, and *inactive*, which only participate in the routing effort. Every active node establishes a connection towards 5 other nodes. An active node sends one bundle per connection. For active users, we chose only the ones that appear at least one time in the first week of the simulations in order to be able to study bundle propagation over an extended period. In each run, we sampled 300 users with 100 of them generating traffic. The simulator used a time step of 1s.

Parameter	Value
Total nodes	5545
Total locations	536
Users sampled	300
Users generating traffic	100
Simulation duration	45 days
Connections per user	5
Bundles per connection	1
Time step	1 s

TABLE I

SIMULATION PARAMETERS.

We performed 5 runs for each scenario. Simulation results reported in the following tables are mean results with confidence intervals at the 90% confidence level, obtained using the Student t distribution.

C. Results

We evaluate the routing algorithms with respect to their transport layer performance. We consider a good algorithm to be one that yields a low average bundle delay, the highest bundle delivery ratio and a low average route length.

We consider two different kinds of scenarios. One with only randomly sampled users and one with only the most active.

With randomly sampled users

In this scenario, we picked 300 users completely at random and we replayed their traces while simulating DTN routing.

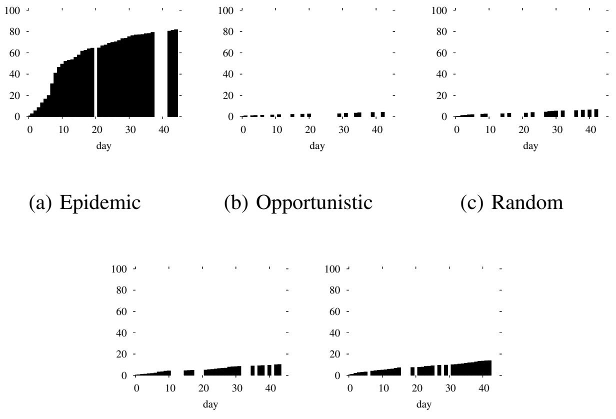
Table II shows the simulation results. It shows for each of the implemented algorithms the mean bundle delay in number of days, the mean delivery ratio, which corresponds to the number of bundles received over the number of bundles sent, and the mean route length in number of hops.

	delivery ratio	**delay**	**route length**
	(%)	(days)	(hops)
Epidemic	82.0 ±2.7	12.5 ±0.9	7.10 ±0.2
Opportunistic	4.9 ±0.6	15.9 ±2.5	1.0 ±0.0
Random	7.2 ±0.5	16.6 ±2.6	3.12 ±0.2
Potato	10.7 ±1.7	19.1 ±1.6	72.7 ±16.5
MobySpace	14.9 ±2.9	18.9 ±1.0	3.8 ±0.2

TABLE II

RESULTS WITH RANDOMLY SAMPLED USERS.

The first thing we can observe is the fact that within the 45 days of simulation there is still a certain number of bundles that are not delivered with Epidemic. The mobility of the 300 nodes or their level of presence were not sufficient to ensure all the deliveries. Our sample included just 5% of the entire set of nodes. By deploying this system on more nodes, the delivery ratio would rise closer to 100%. Furthermore, we did not select nodes based on their mobility characteristics. Some of the nodes may have poor mobility.



(a) Epidemic (b) Opportunistic (c) Random

(d) Potato (e) MobySpace

Fig. 3. Cumulative distribution of packets delivered over the 45 days (Shaded areas represent days during which packets were delivered).

Table II shows that MobySpace delivers twice as many bundles as Random but still far less than Epidemic, which does not miss any opportunities. Random delivers somewhat more bundles than Opportunistic because the bundles are more mobile. This phenomenon is even true for Potato, which outperforms Random but delivers fewer bundles than MobySpace. At first glance, the average bundle delay of MobySpace seems poor. We believe this average is influenced by the fact that more bundles are delivered compared to the other schemes, except Epidemic. The additional bundles delivered by MobySpace might be more difficult to route than the others, leading to higher delays. The investigation of this issue is kept for future work. However, the average bundle delay is an interesting indicator of the performance an algorithm can achieve. Fig. 3 presents the cumulative distribution of packets delivered over time. It shows why the average bundle delay is higher for MobySpace compared to Random. It is simply because MobySpace delivers more packets in a constant fashion over time. Looking now at the average route lengths, we see that in all the cases, except Potato, they are lower than for Epidemic. MobySpace engenders routes that are about half as long as those created by Epidemic. With MobySpace, bundles are transmitted from a node to another because of their mobility patterns, not simply because of the opportunities of contact. Potato engenders routes that are extremely long because, at each contact, bundles switch from one node to another. Potato may not be suitable for a real system because of bandwidth and energy consumption issues.

With the most active users

We also evaluate routing in a scenario with only the most active users, to see the effect of activity on performance. Such a scenario might also be more typical of an ambient network environment. Several metrics can characterize the level of activity. We use the regularity of the users' presence in the network, as measured by the number of active days. The number of users in our data that are active all 45 days is 835. We consider these users as a pool from which we sample for each simulation run.

	delivery ratio	**delay**	**route length**
	(%)	(days)	(hops)
Epidemic	96.7 ±1.9	3.1 ±0.4	7.9 ±0.3
Opportunistic	10.7 ±1.1	17.6 ±1.6	1.0 ±0.0
Random	14.0 ±1.0	17.9 ±1.8	3.5 ±0.1
Potato	38.9 ±1.0	19.1 ±0.4	317.0 ±29.0
MobySpace	50.4 ±4.7	19.5 ±1.3	5.1 ±0.2

TABLE III
RESULTS WITH THE MOST ACTIVE USERS.

Table III shows the simulation results. Considering only the most active users, more bundles are delivered by the algorithms. MobySpace attains a delivery ratio of 50.4% instead of 14.9%. The delivery ratio of MobySpace would have been, as previously, higher if more nodes had participated in the scenario. We intend to study this in future work by performing larger simulations. The average bundle delay achieved is very low for Epidemic compared to the other algorithms. Since nodes are more present in the network, Epidemic certainly needs fewer relays to deliver the packets. Route lengths are still less than Epidemic for Opportunistic, Random, and MobySpace, whereas it is higher for Potato compared to the previous scenario with randomly sampled users.

These results confirm that the MobySpace evaluated in this paper enhances routing as compared to various generic approaches for routing in an ambient network formed by users carrying personal devices in a campus setting. MobySpace achieves a high delivery ratio compared to simple algorithms like Opportunistic, Random, or Potato. It also leads to low bandwidth usage by using routes that are short compared to those constructed by Epidemic.

V. FEASIBILITY

Sec. IV-C has shown encouraging results for the use of MobySpace. However, the simulations rely on the assumption that nodes are aware of their mobility patterns. This section examines two different factors that impact the feasibility of this architecture: the characteristics of the mobility patterns and the possibility of learning them.

A. Mobility pattern characteristics

As noted in our prior work [2], when nodes do not have a high degree of segregation in their mobility patterns, MobySpace can not benefit from the patterns for efficient routing. We analyse here the properties of the mobility patterns we compute on users of Dartmouth College with the help of the relative entropy, S_r, applied to the set of probabilities that make up a mobility pattern. This metric describes the homogeneity of mobility patterns, which is 1 for a pattern with no preference among locations and is small for patterns that strongly prefer a few locations. It is defined for the mobility pattern of node k by:

$$S_r(k) = -\frac{\sum_{i=1}^{n} c_{i_k} \ln c_{i_k}}{\ln n}, \text{ with } n \text{ the number of dimensions} \tag{4}$$

The relative entropy is relevant for the analysis of mobility patterns because it captures a number of important characteristics. The relative entropy is at the same time correlated to the number of locations visited and to the time spent at each location. If a node is equally likely to be found in any location, it has the maximum relative entropy value of 1. If it is very likely to be found in one of a few locations, and unlikely to be found in any other, it has low relative entropy.

Fig. 4 shows the distribution of the relative entropy of users' mobility patterns for the period of 45 days. They display generally low entropy: on average 0.15. The patterns tend to demonstrate good properties for the MobySpace routing scheme because either they contain few components or they contain many components in a non homogeneous fashion.

We study the effect of pattern entropy on MobySpace routing. Table IV shows that the relative entropy of mobility patterns has a great influence the performance in terms of the

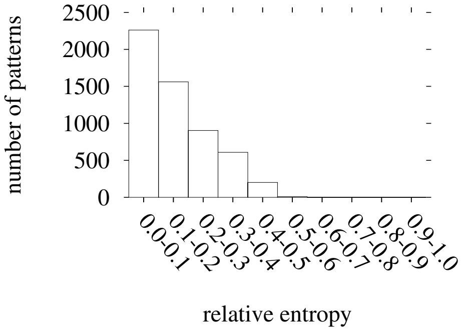


Fig. 4. Relative entropy distribution of mobility patterns.

number of packets that are delivered. The higher the relative entropy, the higher the delivery ratio. Route lengths are stable over the increase of the relative entropy, except for Potato that generates longer routes.

These results show that a lack of diversity in the movements of users does not favor routing in such an environment. In our prior work [2] we demonstrate, with an artificial scenario, that too much diversity can also be a problem if mobility patterns can not be distinguished. In that case, distances in MobySpace have little significance. We were not able to reproduce this demonstration with Dartmouth data because there is no user in the data that visits almost all the locations in a regular fashion. We can conclude that a MobySpace approach is of interest when mobility patterns display a low relative entropy, but not too close to 0.

metric	S_r	**delivery ratio** (%)	**delay** (days)	**route lengths** (hops)
Epidemic	$[0.0 - 0.1]$	45.4 ±5.1	24.1 ±1.7	7.0 ±0.2
	$[0.1 - 0.2]$	79.6 ±3.2	13.1 ±1.8	8.0 ±0.4
	$[0.2 - 0.3]$	97.8 ±1.7	8.7 ±1.3	7.5 ±0.4
	$[0.3 - 0.4]$	99.0 ±0.5	6.0 ±0.9	7.1 ±0.4
Opportunistic	$[0.0 - 0.1]$	2.2 ±0.3	15.0 ±3.8	1.0 ±0.0
	$[0.1 - 0.2]$	4.4 ±0.9	19.8 ±2.4	1.0 ±0.0
	$[0.2 - 0.3]$	9.6 ±2.0	19.9 ±1.0	1.0 ±0.0
	$[0.3 - 0.4]$	24.5 ±2.5	10.9 ±0.9	1.0 ±0.0
Random	$[0.0 - 0.1]$	2.3 ±0.4	11.6 ±4.5	2.0 ±0.3
	$[0.1 - 0.2]$	5.8 ±1.2	20.0 ±2.6	3.0 ±0.2
	$[0.2 - 0.3]$	12.3 ±1.4	17.6 ±2.5	3.5 ±0.1
	$[0.3 - 0.4]$	29.5 ±3.0	12.5 ±1.1	3.9 ±0.1
Potato	$[0.0 - 0.1]$	3.2 ±0.8	16.9 ±1.4	43.0 ±12.0
	$[0.1 - 0.2]$	9.6 ±1.1	19.8 ±2.8	116.2 ±44.2
	$[0.2 - 0.3]$	19.8 ±5.6	20.2 ±1.5	162.7 ±44.7
	$[0.3 - 0.4]$	36.6 ±4.9	12.0 ±1.3	176.6 ±14.3
MobySpace	$[0.0 - 0.1]$	3.4 ±0.4	14.9 ±1.8	2.5 ±0.2
	$[0.1 - 0.2]$	8.4 ±2.4	19.5 ±2.3	3.3 ±0.2
	$[0.2 - 0.3]$	19.8 ±2.4	19.7 ±1.2	4.0 ±0.2
	$[0.3 - 0.4]$	42.3 ±4.8	13.4 ±1.3	4.7 ±0.2

TABLE IV

RESULTS WITH USERS HAVING DIFFERENT ENTROPY.

B. Space reduction

Because transmitting nodes' entire mobility patterns is potentially expensive, we evaluate a scenario in which nodes only diffuse the main components of their mobility patterns. If we sort a node's frequencies of visit to locations in decreasing order, we mean by the main components those frequencies that are at the beginning of this list. All components not transmitted are treated as zeros. (Note that in such a case, MobyPoints no longer all lie on a hyperplane, as the sum of the frequencies can be less than one.) We ran simulations taking into account only the principal 1st, 2nd, or 3rd components of mobility patterns of nodes, and we consider the most active users.

1	**delivery ratio** (%)	**delay** (days)	**route length** (hops)
$l = 1$	39.2 ±5.9	20.2 ±2.6	4.9 ±0.4
$l = 2$	46.3 ±3.3	19.9 ±1.2	5.2 ±0.2
$l = 3$	47.5 ±4.6	19.4 ±1.8	5.2 ±0.2
$l = 536$	50.4 ±4.7	19.5 ±1.3	5.1 ±0.2

TABLE V

RESULTS WITH SPACE REDUCTION. l IS THE NUMBER OF MOST SIGNIFICANT COMPONENTS TAKEN INTO ACCOUNT.

Table V shows that the higher the number of components taken into account, the higher the performance. Surprisingly, the delivery ratio tends very quickly to that of the scenario where all the components are used. These simulations show that only few components are needed to be exchanged between nodes in order to perform routing.

C. Mobility pattern learning

One important condition for the applicability of the MobySpace is whether users can learn their own mobility patterns. In this section we provide a first study on this issue with the Dartmouth data.

For that purpose, we split the 45 days of Dartmouth data into two periods: the learning period and the routing period. The learning period consists of the first 15 days and the routing period, the last 30 days. We study here how well the mobility patterns of nodes learnt in the learning period match the mobility patterns that characterize the routing period. The error is measured as to be the Euclidean distance d between the two mobility patterns, divided by the maximum possible distance between two mobility patterns in the hyperplane:

$$e = \frac{d}{\sqrt{n}}, \text{ with } n \text{ the number of dimensions} \qquad (5)$$

We varied the number of days devoted to learning during the learning period, starting with the one day immediately prior to the routing period, and working back to cover all 15 days of the learning period. Fig. 5 shows the prediction error of mobility patterns, as a function of the number of days devoted to learning. We made this computation for all the nodes and for only the most active ones. We see that, in both cases, the longer nodes learn their own mobility, the closer their mobility patterns approximate the patterns of the routing period. As expected, the most active users learn their patterns more rapidly than the others.

These initial results on the ability of nodes to learn their own mobility patterns are encouraging. They indicate that nodes

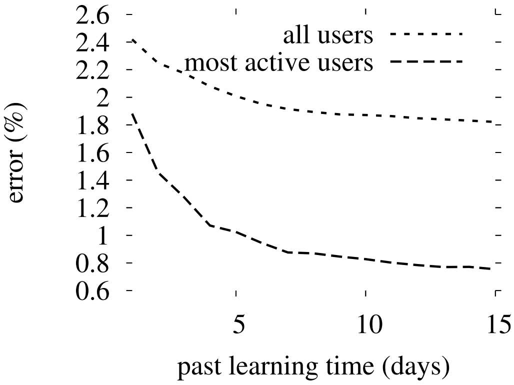


Fig. 5. Prediction error of mobility patterns.

might be able to benefit from their past knowledge to make routing decisions within the MobySpace. Nevertheless, further studies are needed to quantify possible long and short term dependencies in mobility traces. This must be also validated on other mobility traces.

VI. Related work

Some work concerning routing in DTNs has been performed with scheduled contacts, such as the paper by Jain et al. [5] about improving the connectivity of an isolated village to the internet based on knowledge of when a low-earth orbiting relay satellite and a motor bike might be available to make the necessary connections. Also of interest, work on interplanetary networking [17], [18] uses predicted contacts such as the ones between planets within the framework of a DTN architecture.

The case of only opportunistic contacts has been analyzed by Vahdat and Becker [7] using the epidemic routing scheme that consists of flooding. The ZebraNet project [19] is exploring this idea to perform studies of animal migrations and inter-species interactions. Data are flooded in the network such as they get back to access points using animals' mobility. In order to control flooding in DTN, Spyropoulos et al. have introduced the Spray and Wait [20] protocol that distributes a number of copies to relays and then waits until the destination meets one of them. Harras et al. [21] have evaluated simple controlled message flooding schemes with heuristics based, for instance, on hop limits or timeouts. They also introduce a mechanism based on packet erasure. Once a message arrives at the destination after basic flooding, the remaining copies in the buffers of other nodes are erased. Wang et al. [22] reincode the messages with erasure codes and distribute their different parts over a large number of relays, so that the original messages can be reconstituted even if not all packets are received. Widmer et al. [23] have explored network coding techniques. All these approaches distribute multiple copies of packets, they ensure a high reliability of delivery, and a low latency, but they imply high buffer occupancy and high bandwidth consumption. Small et al. [24] propose an analytical study of existing trade-offs between resources consumption such as energy, throughput, buffers and the performance in term of latency.

Some research projects such as Data Mules [25] or SeNTD [26] use mobile network elements to transport data from fixed sensors to a number of access points in an opportunistic fashion. For instance, in SeNTD, data from sensors placed on buoys that monitor the water quality on a lake are relayed by tourist tour-boats or pleasure cruisers.

A large amount of work concerning routing in DTNs has also been performed with predicted contacts, such as the algorithm of Lindgren et al. [27], which relies on nodes having a community mobility pattern. Nodes mainly remain inside their community and sometimes visit the others. As a consequence, a node may transfer a bundle to a node that belongs to the same community as the destination. This algorithm has been designed as a possible solution to provide internet connectivity to the Saami [28] population who live in Swedish Lapland with a yearly cycle dictated by the natural behavior of reindeer. In a similar manner, Burns et al. [29] propose a routing algorithm that uses past frequencies of contacts. Also making use of past contacts, Davis et al. [30] improved the basic epidemic scheme with the introduction of adaptive dropping policies. Recently, Musolesi et al. [31] have introduced a generic method that uses Kalman filters to combine and evaluate the multiple dimensions of the context in which nodes are in order to take routing decisions. The context is made of measurements that nodes perform periodically, which can be related to connectivity, but not necessarily. This mechanism allows network architects to define their own hierarchy among the different context attributes. LeBrun et al. [32] propose a routing algorithm for vehicular DTNs using current position and trajectories of nodes to predict their future distance to the destination. They replay GPS data collected from actual buses in the San Francisco MUNI System, through the NextBus project. Finally, Jones et al. [12] propose a link state routing protocol for DTNs that uses the minimum expected delay as the metric.

VII. Conclusions and future work

The main contribution of this paper has been the validation of a generic routing scheme that uses the formalism of a high-dimensional Euclidean space constructed upon mobility patterns, the MobySpace. We have shown through the replay of real mobility traces that it can applied to DTNs and that it can bring benefits in terms of enhanced bundle delivery and reduced communication costs.

This paper has also presented results of a feasibility study in order to determine the impact of the characteristics of nodes' mobility patterns on the performance and to study nodes' ability to learn their patterns. Thus, to make DTN routing work with the MobySpace, nodes need to have a minimum level of mobility with mobility patterns that can be sufficiently discriminated. We present encouraging results about the capacity of nodes to learn their own patterns. And, we also see that nodes can reduce the number of components in the mobility patterns without great impact on routing performance. This can reduce the overhead of MobySpace and the complexity of handling mobility patterns.

Future work along these lines might include studies concerning the impact of the structure of the Euclidean space, i.e., the number and type of dimensions, and the similarity function. Different kinds of Euclidean space can be investigated by considering schemes like the one described in Sec. III that takes for each dimension the frequency of contacts between a certain pair of nodes or the one that captures cyclic frequential properties during nodes' visits to locations.

Further work remains to be done on the stability of mobility patterns over time and their ability to be learned by nodes. The patterns may contain long term and short term dependencies. Nodes can have different mobility patterns that are each stable. For instance, they can have one for the week-ends, one for the vacations, and one for working weeks.

Because a multi-copy scheme such as Epidemic outperforms all the single-copy schemes in terms of delays, we will focus in future work on the possibility of controlled flooding in MobySpace. It is conceivable that this might bring many of the benefits of epidemic routing, but at a much lower cost in terms of network utilization. There are many ways to do this. One scheme would be to transfer a bundle to not only the first node encountered with a mobility pattern closer to the one of the destination, but to the N first nodes that satisfy this constraint. The degree of flooding can be controlled by modifying the parameter N. Since the principle of routing in a MobySpace is preserved, the flooding is directed, but greater diversity is introduced.

Additionally, further validations need to be conducted on real data and in different environments. MobySpace can be tested on traces coming from larger cell networks, like GSM networks. We might also want to evaluate MobySpace in different social contexts where nodes have specific mobility patterns.

Acknowledgments

We gratefully acknowledge David Kotz for enabling our use of wireless trace data from the CRAWDAD archive at Dartmouth College. We thank Marc Giusti and Pierre Lafon at the STIX laboratory (École Polytechnique / CNRS) for access to the machines we used for the simulations. This work was supported by E-NEXT, an FP6 IST Network of Excellence funded by the European Commission. Also, LiP6 and Thales Communications supported this work through their joint research laboratory, Euronetlab, and the ANRT (Association Nationale de la Recherche Technique) provided the CIFRE grant 135/2004.

References

[1] K. Fall, "A delay-tolerant network architecture for challenged internets," in *Proc. SIGCOMM*, 2003.

[2] J. Leguay, T. Friedman, and V. Conan, "DTN routing in a mobility pattern space," in *Proc. WDTN*, 2005.

[3] "Delay Tolerant Network Research Group (DTNRG)," `http://www.dtnrg.org`.

[4] V. Cerf, S. Burleigh, A. Hooke, L. Torgerson, R. Durst, K. Scott, K. Fall, and H. Weiss, "Delay tolerant network architecture, IRTF draft, draft-irtf-dtnrg-arch-02.txt," July 2004.

[5] S. Jain, K. Fall, and R. Patra, "Routing in a delay tolerant network," in *Proc. SIGCOMM*, 2004.

[6] S. Corson, "Mobile ad hoc networking (MANET): Routing protocol performance issues and evaluation considerations," RFC 2501. IETF, January 1999.

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

[8] P. Hui, A. Chaintreau, J. Scott, R. Gass, J. Crowcroft, and C. Diot, "Pocket switched networks and human mobility in conference environments," in *Proc. WDTN*, 2005.

[9] T. Henderson, D. Kotz, and I. Abyzov, "The changing usage of a mature campus-wide wireless network," in *Proc. Mobicom*, 2004.

[10] M. McNett and G. M. Voelker, "Access and mobility of wireless PDA users," Tech. Rep., UC San Diego, 2004.

[11] M. Balazinska and P. Castro, "Characterizing Mobility and Network Usage in a Corporate Wireless Local-Area Network," in *Proc. MobiSys*, 2003.

[12] E. P. C. Jones, L. Li, and P. A. S. Ward, "Practical routing in delay-tolerant networks," in *Proc. WDTN*, 2005.

[13] L. Song, D. Kotz, R. Jain, and X. He, "Evaluating location predictors with extensive Wi-Fi mobility data," in *Proc. Infocom*, 2004.

[14] C. Lonvick, "The BSD syslog protocol," RFC 3164. IETF, August 2001.

[15] N. Eagle and A. Pentland, "Social serendipity: Mobilizing social software," in *Proc. PerCom*, 2005.

[16] "UMassDieselNet: A Bus-based Disruption Tolerant Network," `http://prisms.cs.umass.edu/diesel/`.

[17] I. F. Akyildiz, O. Akan, C. Chen, J. Fang, and W. Su, "Interplanetary internet: state-of-the-art and research challenges," *Computer Networks*, vol. 43, no. 2, pp. 75–112, 2003.

[18] S. Burleigh, A. Hooke, L. Torgerson, K. Fall, V. Cerf, B. Durst, and K. Scott, "Delay-tolerant networking: an approach to interplanetary internet," *IEEE Communications Magazine*, vol. 41, no. 6, pp. 128–136, 2003.

[19] H Juang, H. Oki, Y. Wang, M. Martonosi, L. Peh, and D. Rubenstein, "Energy-efficient computing for wildlife tracking: Design tradeoffs and early experiences with Zebranet," in *Proc. ASPLOS-X*, 2002.

[20] T. Spyropoulos, K. Psounis, and C. Raghavendra, "Spray and wait: An efficient routing scheme for intermittently connected mobile networks," in *Proc. WDTN*, 2005.

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

[22] Y. Wang, S. Jain, M. Martonosi, and K. Fall, "Erasure coding based routing for opportunistic networks," in *Proc. WDTN*, 2005.

[23] J. Widmer and J. Le Boudec, "Network coding for efficient communication in extreme networks," in *Proc. WDTN*, 2005.

[24] T. Small and Z. J. Haas, "Resource and performance tradeoffs in delay-tolerant wireless networks," in *Proc. WDTN*, 2005.

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

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

[27] A. Lindgren, A. Doria, and O. Schelen, "Probabilistic routing in intermittently connected networks," in *Proc. SAPIR*, 2004.

[28] A. Doria, M. Uden, and D. P. Pandley, "Providing connectivity to the Saami nomadic community," in *Proc. Developement by Design Conference*, 2002.

[29] B. Burns, O. Brock, and B. N. Levine, "MV routing and capacity building in disruption tolerant networks," in *Proc. Infocom*, 2005.

[30] J. A. Davis, A. H. Fagg, and B. N. Levine, "Wearable computers as packet transport mechanisms in highly-partitioned ad-hoc networks," in *Proc. ISWC*, 2001.

[31] M. Musolesi, S. Hailes, and C. Mascolo, "Adaptive routing for intermittently connected mobile ad hoc networks," in *Proc. WOWMOM*, 2005.

[32] J. LeBrun, C. Chuah, and D. Ghosal, "Knowledge based opportunistic forwarding in vehicular wireless ad hoc networks," in *Proc. VTC Spring*, 2005.