geoDTN: Geographic Routing in Disruption Tolerant Networks

Jó Ágila Bitsch Link, Daniel Schmitz, Klaus Wehrle
Communication and Distributed Systems
RWTH Aachen University, Germany
{jo.bitsch,daniel.schmitz3,klaus.wehrle}@rwth-aachen.de

***Abstract*—In this paper we present a disruption tolerant routing algorithm based on geographic location information, which improves upon the hop count compared to the current state of the art by up to a factor of three in large scale human networks. Leveraging only the history of geographic movement patterns in the two-hop neighborhood, our algorithm is able to perform well in the absence of knowledge of social interaction between nodes and without detailed future schedule information.**

Representing previously visited locations as probability distributions encoded in an efficient vector, we formalize a heuristic for efficiently forwarding messages in disruption tolerant networks, implement a framework for comparing our approach with the state of the art, and evaluate key metrics, such as hop count and delivery rate, as well as energy consumption and battery depletion fairness on real world data. We are able to outperform the state of the art in human mobility based networks considerably in terms of energy usage per node, thereby extending data network availability further into areas devoid of otherwise necessary communication infrastructure.

***Index Terms*—pocket switched networks; geography; routing**

I. Introduction

In the absence of network infrastructure, pocket switched networks [1] provide communication capabilities to mobile devices using the nodes provided by other network participants to store and forward messages. While the lack of continuous end-to-end connectivity, and the dependence on typically resource constrained devices present us with a plethora of challenges, this fundamentally different type of communication network offers a chance for next generation networks to provide data connectivity to areas, where infrastructure is either too expensive or otherwise infeasible.

Previous approaches mainly use social characteristics [2], statistics of meetings between nodes [3] or other heuristics [4], we base our scheme on the notion of location distribution. Upon the meeting of two nodes, our approach compares their distributions and chooses the subsequent carrier for a message bundle accordingly. Using this approach, we are able to make meaningful forwarding decisions with only local knowledge even if an intermediary node has never heard of the recipient.

A. Contributions

1) **Modeling human mobility:** We model human mobility as a probability distribution around known anchor points. With this, probabilities for future locations can be quantified.
2) **Routing heuristic based on location distribution:** Based on the above model, we predict future connectivity and present a heuristic for disruption tolerant networking, specifically addressing regional and global mobility.
3) **Trace of self reported global mobility:** The advent of location based online social networks allowed us to collect global location data for a variety of users.

B. Organization

After a discussion of relevant related work, we present our design in section III. A special focus is on the representation of mobility, the heuristic used to compare meeting nodes, and the actual routing mechanism. The evaluation in section IV introduces the mobility traces, and the traffic model we used, as well as the performance evaluation. We also compare the algorithms presented in the related work section in terms of energy and fairness to our approach. Finally, we conclude in section V with a short reflection on our approach.

II. Related Work

A multitude of disruption and delay tolerant routing schemes have been proposed. While the concept is as old as the postal system itself, new interest was found in 2004 when Jain et al. [5] formulated the routing problem in DTNs.

Proposed routing strategies can be categorized in four main categories: statistics, social network, geography based, and other heuristics. A prominent instance of the statistics based approach is PRoPHET [3], which predicts future delivery by modeling links between nodes as a probability, rising on each meeting, but aging during inactivity. The viability of a path is then measured as the product of all intermediate links. This approach assumes information on every link being available to all nodes, thus limiting its scalability to well-defined regions.

Social network based approaches, like SimBet [2] are based on social network characteristics like the similarity of a node with the destination of a message bundle and the betweenness of a node compared to its neighbors. SimBetAge [6] adds the concept of aging as in PRoPHET. While these approaches depend only on local knowledge, their scalability is still limited by the rise of local maxima in social networks over a certain size.

This paper is not the first to explore the practicality of geographic information for disruption tolerant routing. GPSR

This research was funded in part by the DFG Cluster of Excellence on Ultra High-Speed Mobile Information and Communication (UMIC), German Research Foundation grant DFG EXC 89.

greedily forwards messages closer to the destination in an urban city environment [7]. However, it assumes storage nodes at every intersection on the path, making it infeasible in practice. Move [8] builds on the heuristic that if a node is moving towards the destination, it is better suited to forward messages. In MobyPoints [9] a vector encodes meeting probabilities for all possible locations. They then use the similarity score between these vectors for forwarding decisions. Nagy et al. [10] provide a more general survey of location based approaches.

Routing mechanisms tend to optimize message transfer per message. However, as Pujol et al point out in [11], this typically puts a significant load on very few central nodes, draining their energy unacceptably fast. Fairness in load distribution is therefore another important aspect to consider.

The core idea of our approach bases on the predictability of human mobility, which Song et al. [12] impressively showed using data which was collected tracking the mobile customers of a cellular GSM network. Previously, Brockmann et al. showed that human travel roughly follows a power law distribution [13], a notion we also consider in our approach.

III. Design

GeoDTN bases its routing decision on previous node movements with a probabilistic node meeting heuristic. The nodes' movement patterns are reactively compared to the destination's pattern resulting in a transitive probabilistic meeting score $s_{i,dest} \in [0,1]$, denoting of how probable it is that node i and the destination node have a common movement domain.

A. Mobility Representation

The nodes mobility representation extends the approach of MobyPoints [9] that each node has a mobility vector containing all necessary information for the routing decision. The locations of observed nodes are collected into *clusters* represented by a bivariate normal distribution and a confidence. For n clusters, each with j_m locations, the probability for cluster m is:

$$p_m = \frac{j_m}{\sum_{i=1}^{n}(j_i)}$$

We model the confidence $\zeta \in [0,1]$ of a cluster as an exponential decay function:

$$\zeta(\Delta t) = e^{-\alpha \cdot \Delta t}$$

with Δt denoting the time since the last update and α an aging parameter. Every time the node visits a location in the corresponding cluster, we reset ζ to 1.

All nodes exchange the mobility vectors of their 2-hop neighborhood upon contact. Therefore a node n_i may receive a mobility vector of another node n_j several times. These vectors may have different confidence values for the same clusters or include new clusters. Therefore the mobility vector of a node n_i may not be consistent between neighboring nodes.

If node n_i receives a mobility vector v from node n_j, we consider the following cases:

1) **$\mathbf{n_i}$ receives the mobility vector of node $\mathbf{n_j}$, $\mathbf{v_{n_j}}$.**
   As node n_j is the authoritative source for information on itself, node n_i updates all its information accordingly.
2) **$\mathbf{n_i}$ receives $\mathbf{v_{n_k}}$ which it already has.**
   Node n_i compares all clusters in the mobility vector v_{n_k} with the ones received from n_j keeping the cluster with the higher confidence value ζ.
3) **$\mathbf{n_i}$ receives $\mathbf{v_{n_k}}$ which it does not have yet.**
   n_i has no previous knowledge of v_{n_k} and therefore adopts the complete vector v_{n_k} from n_j.

Node n_i prunes the clusters in the mobility vectors of neighboring nodes when their confidence drops under a fixed threshold ϵ_ζ. Entries in its own mobility vector are pruned using an adaptive threshold ϵ_p. Node n_i calculates this threshold using the mean over all probabilities in its mobility vector $\bar{p}$ and its standard deviation σ_p^2 using the following formula, as well as a spreading factor γ:

$$\epsilon_p = \bar{p} - \gamma \cdot \sigma_p^2$$

B. Neighbor Score

We define two nodes as being neighbors if they are in radio range at a common location frequently. To determine this, we define a neighbor score using the mobility vector v_{n_i} of a node, with each entry consisting of the bivariate distribution represented by $f_{i,k}(x,y)$, parametrized by $\bar{x}$, $\bar{y}$, σ_x^2, σ_y^2 and ρ, as well as its probability $p_{i,k}$ and its confidence $\zeta_{i,k}$.

The movement score of an element $e_{i,k}$ of the mobility vector of node n_i within the area $([x_{\text{min}}, x_{\text{max}}], [y_{\text{min}}, y_{\text{max}}])$ can thus be represented as:

$$s_{e_{i,k}}^{\text{local}} = \int_{x_{\text{min}}}^{x_{\text{max}}} \int_{y_{\text{min}}}^{y_{\text{max}}} f_{i,k}(x,y)\,dx\,dy$$

Using this, two nodes n_i and n_j can calculate their meeting score upon meeting. For this, they calculate their common movement area as the intersection of their own movements. The neighbor score can thereby be defined as:

$$\text{Neighbor Score} = \frac{\text{Movement in } [x_{\text{min}}, x_{\text{max}}] \text{ and } [y_{\text{min}}, y_{\text{max}}]}{\text{complete movement}}$$

The actual calculation also weights in the probabilities p of the actual entries in the mobility vector as well as its confidences ζ.

$$s_{i,j} = \frac{\sum_{e_{i,k} \in v_i} \zeta_{e_{i,k}} \cdot p_{e_{i,k}} \cdot s_{e_{i,k}}^{\text{local}} + \sum_{e_{j,l} \in v_j} \zeta_{e_{j,l}} \cdot p_{e_{j,l}} \cdot s_{e_{j,l}}^{\text{local}}}{|v_i| + |v_j|}$$

Two nodes are neighbors if their neighbor score $s_{i,j}$ within their common area reaches a neighbor threshold ϵ_n.

C. Routing

The routing decision of geoDTN bases on the neighbor scores $s_{i,\text{dest}}$. We distinguish three different cases, depending on how far a node is from the destination. When node n_i comes into radio range of node n_j the neighbor scores $s_{i,\text{dest}}$ and $s_{j,\text{dest}}$ are compared. If both are below a threshold, we employ (1) *distance mode*. If a message is stuck in a local minimum, we employ (2) *rescue mode*. Finally, when a node is in the vicinity of the destination, we use (3) *scoring mode*.

1) Distance mode: If both $s_{i,\text{dest}}$ and $s_{j,\text{dest}}$ fall below the scoring mode threshold, a bundle is forwarded into the neighborhood of the message destination following a hill climbing strategy. The distance of node n_i to the destination n_{dest} is the minimum distance between their mobility vectors weighted with confidence ζ and probability p:

$$d_{i,\text{dest}} = \min_{\substack{f_{i,k} \in v_i \\ f_{j,\text{dest}} \in v_{\text{dest}}}} \left(m \cdot \text{dist}(f_{i,k}, f_{j,\text{dest}})\right)$$

The factor $m = 10 \cdot e^{-\beta \cdot \zeta \cdot p}$ increases the distance for elements of the mobility vector that have only a small probability or a low confidence, thereby favoring those elements with a high probability p and a high confidence ζ. The factors 10 and β are used to adjust this weighting.

2) Rescue mode: As distance mode employs a hill climbing strategy, message bundles may get stuck in a local minimum. To counter problem, we use random walks. Thus, if a node n_i detects a stuck bundle, this bundle b is then forwarded randomly within the network until the distance $d_{b,\text{dest}}$ of said bundle is either smaller than $d_{i,\text{dest}}$, the bundle reaches scoring mode distance to n_{dest}, or a maximum number of hops ϵ_h is reached.

We used two variants for random forwarding of a bundle:

1) Upon contact with another node the bundle is forwarded with a probability of $p = 0.5$, or
2) Upon contact with another node the bundle is forwarded with a probability of $p = 0.3$ if this bundle already passed through this particular node, else it is forwarded with a probability of $p = 0.7$.

The second variant allows the message bundle to explore new network parts with a higher probability and thus gets out of the local minimum network faster.

Detection of stuck bundles bases on a time threshold ϵ_s and a log of all nodes previously visited by that bundle. Since a node is aware of its neighborhood it can detect if it tried all neighbors and if ϵ_s elapsed since it received the bundle.

3) Scoring mode: Finally, when a bundle reaches the vicinity of the destination n_{dest}, we use a scoring mode to select the best possible next hop. When node n_i carrying the bundle comes into range of node n_j, they both calculate their respective neighbor scores with the destination $s_{i,\text{dest}}$ and $s_{j,\text{dest}}$. Additionally, they calculate the neighbor scores of their neighborhood, weighting them with their own score. Node n_i therefore forwards the bundle to n_j, if:

$$\begin{aligned} &\max\left(\{s_{i,\text{dest}}\} \cup \{s_{k,\text{dest}} : n_k \in N_i(2)\}\right) \\ <&\max\left(\{s_{j,\text{dest}}\} \cup \{s_{l,\text{dest}} : n_l \in N_j(2)\}\right) \end{aligned}$$

Using this scheme, the bundle is continuously forwarded closer to the destination, using the best possible path based on the mobility patterns of the encountered nodes.

On a final note about addressing, we intentionally leave out this problem in this paper, assuming the sending node is aware of the destination's mobility vector. We also anticipate that an iterative refinement of the destination's mobility vector, while the bundle traverses the network and nodes are more aware of the destination's actual mobility, leads to good results.

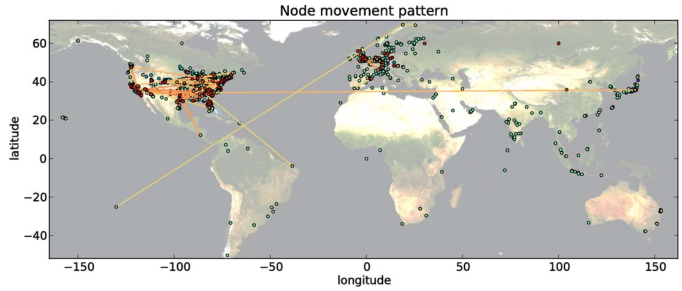

Fig. 1. Resulting network from the observation of 221 BrightKite users over the course of two months. For readability, we also display connecting lines for a sample of 50 users.

IV. Evaluation

We evaluate geoDTN based on two different scenarios. First, we use *UMass DieselNet Fall 2007* traces [14] containing GPS movements of 33 nodes for 18 days, where we assume a simulated radio range of 150 meters.

Secondly, we use a human based network trace making use of the social networking site *BrightKite*. We follow the *check-ins* of 221 very active users over a period of two months assuming a contact, whenever they come within 300 meters of each other. Hence, these contacts are based on real human behavior, which for the purpose of this paper are assumed to be 221 test users making use of pocket switched networking devices. The resulting network is shown in Figure 1.

We generate discrete events from the real world timing so that between node actions, such as meetings and position changes, at least 5 and at most 15 events occur. This filtering is necessary for running our simulation in acceptable time; however, it does not change location order and prominence. After filtering we obtain 43786 discrete time events.

A. Traffic Model

The assumed traffic model generates a message from each node to d random destinations every t time units. For evaluation purposes, we assume timing intervals $t \in \{5, 10\}$ and the number of destinations $d \in \{2, 3, 5, 7, 10\}$. Those numbers mirror assummed probabilistic usage patterns.

B. Simulation and Metrics

The simulation uses static message and node trace files within a custom python simulation software. Assuming a trace file length of 5000 time events, we allow a learning period of 2000 time events, upon which message generation takes place for 500 time events. Finally we allow messages another 2500 time events to arrive at their destination. Using static message and node trace files enable us to repeat experiments accurately.

Our key metrics are delivery rate, delivery time and hop count. In addition to that we also consider the energy consumption per node and received bundle, as well as the fairness, i.e. how fast more central nodes are drained of energy.

Epidemic routing with unlimited queue length and transmission capacity provides us with the theoretical upper bound in terms of delivery rate, and the lower bounds in terms of delivery time and hop count. In contrast, Direct Delivery—only forwarding messages to the destination when the destination is directly in radio range—provides us with a sensible upper

Fig. 2. Evaluation of the delivery time, delivery rate, and hop count in UMass DieselNet.

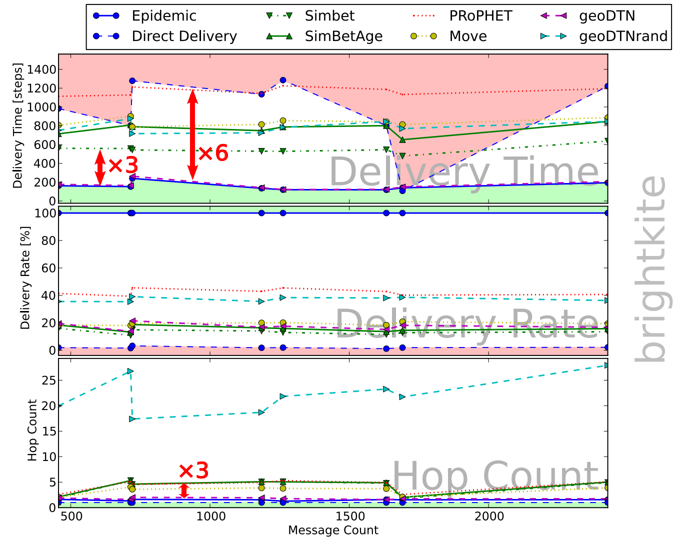

Fig. 3. Evaluation of the delivery time, delivery rate, and hop count in BrightKite.

bound for the delivery time, and a sensible lower bound for the delivery rate as any slower or less effective algorithm is evidently flawed. At the same time it is a true lower bound in terms of hop count—namely 1.

Additionally we compare geoDTN with PRoPHET, SimBet, SimBetAge, and Move. Furthermore we evaluate geoDTN with (*geoDTNrand*) and without (*geoDTN*) rescue mode.

A detailed parameter evaluation using hill climbing strategies to optimize parameters of the DieselNet trace files (not shown) results in the following optimal parameter settings: Aging parameter $\alpha = 0.7$, neighborhood score $\epsilon_n = 0.0009$, and distance threshold $\epsilon_d = 0.0005$.

C. Performance in DieselNet

Referring to Figure 2, we observe a delivery rate of approximately 85% for both variants of geoDTN, very similar to the results of SimBet and SimBetAge, still outperforming the simpler Move. PRoPHET however consistently outperforms our approach by a margin of 5% in terms of delivery rate, underlining the quality of PRoPHET as one of the best state of the art DTN routing algorithms wrt. delivery rate.

Regarding hop count, geoDTNrand needs an average of 1.2 more hops than geoDTN to the benefit of a 2% higher delivery rate. PRoPPHET with an average of 2.2 more hops and Move with 5.1 more hops are significantly outperformed by our approach, whereas SimBet and SimBetAge are nearly optimal with respect to this metric, with SimBetAge performing best—aside from the theoretical boundaries.

The delivery time of geoDTN is worse than the other algorithms we considered, while SimBet and SimBetAge outperform them. This indicates that our geoDTN variants occasionally miss the optimal point in time for forwarding a message. Deeper inspection of the log files revealed the reason for this slightly lesser performance lying in the transitive score calculation which leaves room for further improvement.

D. Performance in BrightKite

Figure 3 shows the evaluation results for the BrightKite traces. The simulations use the same parameter as before.

Like in the DieselNet traces, geoDTN's performance in delivery rate is similar to SimBet and SimBetAge with geoDTN being slightly better. GeoDTNrand performs significantly better than geoDTN at the cost of a very high hop count. Nevertheless, the rescue mode shows itself to be a reasonable extension to improve delivery metrics. The hop count can be further reduced when optimizing the random walk parameter for BrightKite like scenarios.

PRoPHET outperforms the other algorithms but delivers only about 2% more packets than geoDTNrand. Move performs better in this scenario in contrast to the other algorithms. This indicates that taking the angle of movement into account is a reasonable indicator for routing decisions in networks based on human mobility.

In this scenario, geoDTN (not geoDTNrand) significantly outperforms the other algorithms in terms of hop count and delivery time. On average, it performs 230% better in terms of hop count, in individual cases 300%. With respect to delivery time, it outperforms SimBet by a factor of 3, SimBetAge and Move by a factor of 4 and PRoPHET by a factor of almost 6.

Therefore, we show that geoDTN performs well in this scenario based on human mobility on a global level scale.

E. Overall Energy Consumption

Figure 4 shows the energy consumed per simulation run for each routing algorithm. We distinguish between the overall energy consumption and the per received bundle energy consumption. Each *send and receive* action raises the used amount of energy by one, assuming constant message size.

Since we model message forwarding as taking place in a broadcast medium, energy consumption counts towards all nodes in radio range, not just the node a message was addressed to on this hop.

As a growing hop count results in additional send and receive actions, PRoPHET and Move display the highest energy consumption. Counterintuitively, Move consumes less energy than PRoPHET although it has a higher average hop count. The reason for that lies in that PRoPHET uses nodes in

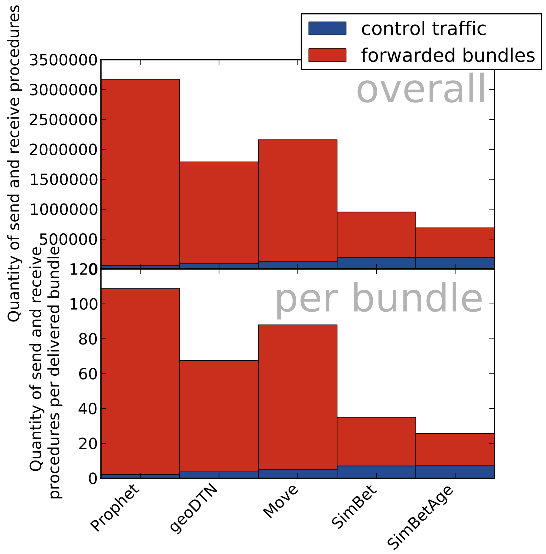

Fig. 4. Quantity of sent and received messages overall (top) and per received bundle (bottom). Control messages (blue) and message bundles (red) are counted separately.

more central and dense areas in the network to a greater extent. This results in more nodes receiving bundles not addressed to them and therefore a higher energy consumption.

The amount of control messages is inversely related to the hop count. This suggests that more and most recent control data is useful to find optimal paths in dynamic networks.

F. Fairness

The fairness evaluation is a measure of the energy distribution over the different nodes and is related to the fairness approach of Pujol et al. We define fairness such that all nodes need approximately the same amount of energy instead of exploiting a small subset of – more central – nodes.

Figure 5 shows the energy needed per node in the DieselNet evaluation. SimBetAge followed by SimBet clearly outperform the other algorithms with low overall energy consumption and a relatively fair distribution over the different nodes.

However, geoDTN outperforms Move and PRoPHET in this scenario as well. PRoPHET, in comparison to all other algorithms we investigated, fails being energy efficient as well as being fair. It thus depletes the most central nodes the fastest lowering the incentive for central users to participate.

V. Conclusion

We presented a novel disruption tolerant routing scheme based on the mobility of the participating nodes. The scheme employs three routing modes. *Distance mode* routes a packet by minimizing the distance to the destination on the path to the destination node's neighborhood. In case of failure *rescue mode* assures that a message bundle makes progress when stuck in a local minimum. Within the destination's neighborhood a probabilistic *scoring* function determines the similarity of a node's movement pattern with that of the destination's with respect to probability and confidence in the stored information. The *neighbor score* $s_{i,j} \in [0, 1]$ allows the comparison of mobility vectors of different nodes, taking the *probability* and *confidence* of individual locations into account.

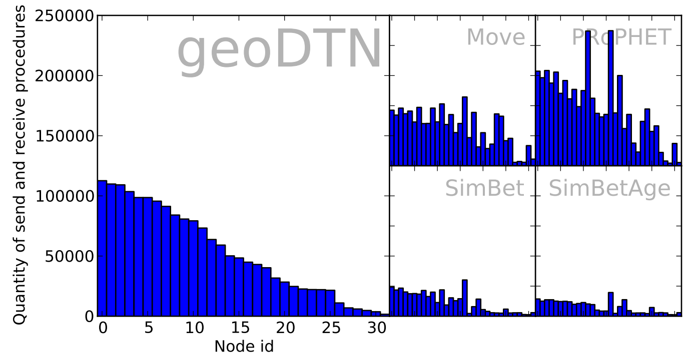

Fig. 5. Fairness evaluation on the UMass DieselNet traces on an equal scale.

The evaluation shows that geoDTN works similarly well as algorithms based on social group characteristics such as SimBetAge and outperforms binary movement algorithms as Move. In dynamic, less periodic networks, geoDTN significantly outperformed the other algorithms in hop count and delivery time. It performs in general 130%, in individual cases even 200%, better in hop count. In terms of delivery time, it outperforms the other algorithms on average by a factor of 3, PRoPHET by factor 6. This demonstrates geoDTN to be a fast self adapting algorithm for human mobility based networks.

The energy and fairness evaluation shows that geoDTN performs significantly better than the best state of the art algorithm in delivery rate, PRoPHET. GeoDTN needs less hops, thus less transmission energy and distributes the load fairer over the different nodes.

References

[1] P. Hui, A. Chaintreau, R. Gass, J. Scott, J. Crowcroft, and C. Diot, "Pocket Switched Networking: Challenges, feasibility and implementation issues," *LNCS*, vol. 3854, 2006.

[2] E. M. Daly and M. Haahr, "Social network analysis for routing in disconnected delay-tolerant manets," in *Proc. of MobiHoc '07*. Montreal, Quebec, Canada, 2007.

[3] A. Lindgren, A. Doria, and O. Schelén, "Probabilistic routing in intermittently connected networks," in *Service Assurance with Partial and Intermittent Resources*, ser. LNCS. Springer, 2004, vol. 3126/2004.

[4] T. Spyropoulos, K. Psounis, and C. S. Raghavendra, "Spray and wait: an efficient routing scheme for intermittently connected mobile networks," in *Proc. of WDTN '05*. Philadelphia, PA, USA, 2005.

[5] S. Jain, K. Fall, and R. Patra, "Routing in a delay tolerant network," in *Proc. of ACM SIGCOMM'04*, 2004.

[6] J. Á. Bitsch Link, N. Viol, A. B. D. Goliath, and K. Wehrle, "SimBetAge: dealing with change in social networks for pocket switched networks," in *Proc. of UNET'09*, Rome, Italy, 2009.

[7] B. Karp and H. Kung, "GPSR: greedy perimeter stateless routing for wireless networks," in *Proc. of ACM MobiCom'00*. Boston, MA, USA, 2000.

[8] J. Lebrun, C. nee Chuah, D. Ghosal, and M. Zhang, "Knowledge-based opportunistic forwarding in vehicular wireless ad hoc networks," in *In IEEE Vehicular Tech. Conf*, 2005.

[9] J. Leguay, T. Friedman, and V. Conan, "DTN routing in a mobility pattern space," in *Proc. of ACM WDTN'05*, 2005.

[10] G. Nagy and S. Salhi, "Location-routing: Issues, models and methods," *European Journal of Operational Research*, vol. 177, no. 2, 2007.

[11] J. Pujol, A. Toledo, and P. Rodriguez, "Fair routing in delay tolerant networks," in *Proc. of IEEE INFOCOM'09*, 2009.

[12] C. Song, Z. Qu, N. Blumm, and A. Barabasi, "Limits of predictability in human mobility," *Science*, vol. 327, no. 5968, 2010.

[13] D. Brockmann, L. Hufnagel, and T. Geisel, "The scaling laws of human travel," *Nature*, vol. 439, no. 7075, 2006.

[14] J. Burgess, B. N. Levine, et al.,"CRAWDAD data set umass/diesel (v. 2008-09-14)," Downloaded from http://crawdad.cs.dartmouth.edu/umass/diesel, 2008.