Social-Similarity-based Routing Algorithm in Delay Tolerant Networks

Daniel Rothfus[1], Christina Dunning[2], Xiao Chen[3]
[1]Department of Computer Science, LeTourneau University, Longview, TX USA
[2]Department of Computer Science, University of Montana, Billings, MT USA
[3]Department of Computer Science, Texas State University, San Marcos, TX USA
Email: danielrothfus@letu.edu, christina.dunning@umontana.edu, xc10@txstate.edu

***Abstract*—A Delay Tolerant Network (DTN) is a type of wireless mobile network that does not guarantee continuous network connectivity. One application can be found in the social communication networks that are becoming ever more ubiquitous with the development of more portable, affordable, and powerful mobile devices. In such a network, people move around and contact each other based on their common interests. Recently, some social-feature-based routing protocols that take advantage of recorded social features to steer the routing in the right direction have been proposed. In such protocols, every node finds its differences in social features with the destination. The routing hence becomes a process to resolve the social feature differences between a source and a destination. However, we believe that merely distinguishing nodes by "same" or "different" social features is insufficient in reflecting nodes' dynamic behavior. Therefore, we put forward SOSIM, a novel routing algorithm that uses similarity metrics from data mining on nodes' contact history to more accurately evaluate social similarities between nodes. To improve efficiency, we apply delegation forwarding in our algorithm. Analysis indicates that our algorithm can improve routing performance with a low implementation cost. The simulation results using real trace also show that our algorithm outperforms the existing ones.**

***Index Terms*—delay tolerant networks, delegation forwarding, routing, similarity, social features**

I. Introduction

A Delay Tolerant Network (DTN) is a type of wireless mobile network that may lack continuous network connectivity. It can appear in the following applications: satellite communication networks [14], village area networks [6], connected vehicle networks [1], and social communication networks [7]. As more portable, affordable, and powerful mobile devices such as smartphones, tablets, and laptops are developed, social communication networks are becoming more ubiquitous. In such a network, people move around and contact based on their common interests. Because of this, the social features of people play an important role in their contact patterns.

Recently, several social-feature-based DTN routing schemes have been proposed [5], [13]. The idea is to use the social features of a node (an individual) for routing guidance. The features $F_1, F_2, \cdots$ may refer to $nationality, city, language$, and so on. The intuition is that *people come in contact more frequently if they have more social features in common*. In the routing process, feature differences are resolved hop-by-hop until the destination is reached. For example, assume we consider four social features: $\langle Nationality, City, Affliation, Language\rangle$. Suppose destination D's values in these four social features are: $\langle USA, NewYork, Student, English\rangle$. These are the target social features that a source wants to reach, so we set the vector of D to $\langle 1, 1, 1, 1\rangle$. Suppose there is a source that wants to send a message to D. If it has the same value for feature F_i, then the value in its F_i dimension is set to 1, otherwise it is set to 0. Suppose a source has nothing in common with the destination, so its vector is $\langle 0, 0, 0, 0\rangle$. The routing process then attempts to resolve the differences between $\langle 0, 0, 0, 0\rangle$ and $\langle 1, 1, 1, 1\rangle$ via intermediate nodes. A possible path, represented by nodes' social feature vectors, would be $\langle 0, 0, 0, 0\rangle \rightarrow \langle 1, 0, 0, 0\rangle \rightarrow \langle 1, 0, 1, 0\rangle \rightarrow \langle 1, 1, 1, 0\rangle \rightarrow \langle 1, 1, 1, 1\rangle$.

In this paper, we take the idea a step further, motivated by the reality that people's social features do not always reflect their dynamic behavior. For example, consider people from New York who actually spend most of their time in Texas. A simple feature value in their profiles will not reflect their dynamic behavior. Another situation is that in real life, people mostly communicate with people who have many social features in common. So if we just look at 1 or 0 difference in social features, then it is hard to tell which one is better. For example, assume we just consider social features like $\langle City, Affiliation\rangle$. Suppose destination D's social feature values in these two dimensions are $\langle NewYork, Student\rangle$. Then the vectors of two candidate forwarders A and B who have the same social feature values as D will both be set to $\langle 1, 1\rangle$, which makes them indistinguishable.

Therefore, we propose a more accurate way to evaluate social closeness or similarity of nodes that takes their dynamic behavior into account. Our method uses nodes' meeting ratios with other nodes having these social feature values in history. For the above example, if node A meets New Yorkers 90% of the time and students 80% of the time while B's frequencies for the same meetings are $\langle 60\%, 40\%\rangle$ during the time we observe, then we can tell candidate A is a better choice. Generally speaking, a better candidate should be the one who is more socially similar to the ideal candidate R who has a vector of $\langle 100\%, 100\%\rangle$, meaning that it meets people like the destination all the time. So the key to more accurately evaluate candidate forwarders is to calculate the

978-1-4673-3122-7/13/$31.00 ©2013 IEEE

social similarity between nodes based on their contact history. We derive the similarity metrics from those in data mining [3] and will explore Euclidean, Weighted Euclidean and Tanimoto similarity metrics.

Based on the above idea, we put forward a novel routing algorithm called *SOSIM* based on *SOcial SIMilarities* of nodes if their contact history is considered. In addition, to achieve efficient routing, we apply delegation forwarding [2]. Delegation forwarding is known to bring down the expected cost of delivery from $O(n)$ to $O(\sqrt{n})$, where n is the number of nodes in the network. In delegation forwarding, a copy is transferred to a newly encountered node if the node is "closer" to the destination than other nodes that the current node has already met. Here, we use social similarity as a forwarding metric. Analysis indicates that SOSIM can improve delivery ratio and reduce latency with a low implementation cost. The simulation results comparing SOSIM to existing algorithms also exhibit its efficiency.

The rest of the paper is organized as follows: Section II references the related works; Section III presents our routing algorithm; Section IV gives the analysis; Section V shows the simulation results; and the conclusion is in Section VI.

II. Related works

Many DTN routing protocols have been proposed in the literature. In the beginning there were rudimental approaches such as Flooding [11], [12] and Wait [4]. More recently, algorithms were proposed to utilize social features in DTN routing [5], [13]. The details are as follows.

A. Rudimental Approaches

One rudimental routing approach in DTNs is to perform a Flooding-based route discovery as in [12] where a host will forward a message to all hosts it comes into contact with so that the spread of the message is like an epidemic of a disease. The Flooding-based routing and its derivatives use multiple copies of a single message to decrease latency and improve delivery ratio and robustness. However, they have a high cost [10]. Another basic algorithm in DTNs is Wait (or direct delivery) [4], where the source does not forward copies to any intermediate nodes at all. It just waits and sends the message to the destination when they meet. In this approach, the number of copies is low (only one copy) but the latency can be very high. The Flooding and Wait algorithms will be used as benchmarks for our simulations.

B. Social-feature-based Approaches

Some more recent DTN routing algorithms use social features to guide routing. In [5], Mei et al. found that individuals with similar social features tend to come into contact more often in DTNs. The individuals are characterized by high dimensional feature profiles, though usually only a small subset of important features are extracted from feature profiles. Although the initial idea of social feature-based routing was proposed by [5], Wu et al. [13] provide a systematic approach to multi-path routing in the feature-space by taking advantage of the structural property of hypercubes to resolve social feature differences between a source and a destination. The advantage of the social-feature-based approach is that it does not need to record nodes' contact history. The drawback is that it can not accurately capture the dynamic behavior in the network. Therefore, a new routing algorithm that can adapt to a node's dynamic behavior is needed.

III. Routing Protocol

In this section, we put forward a routing algorithm called SOSIM that identifies the best forwarding candidate using nodes' social similarities based on their contact history.

A. Routing Algorithm

The routing algorithm is shown in Fig. 1, where we consider m social features in the network. Each individual node has a vector based on its social features. For convenience, we use the node's label as its vector's label. Thus, a node X has a vector of $X\langle x_1, x_2, \cdots, x_m\rangle$ and a node Y has a vector of $Y\langle y_1, y_2, \cdots, y_m\rangle$. Metric $S(X,Y)$, whose details are described in the next section, is used to calculate social similarity between two nodes X and Y. In the routing process, we apply the idea of delegation forwarding proposed by Erramilli et al. [2] because it can bring down the expected cost of delivering messages from $O(n)$ to $O(\sqrt{n})$, where n is the number of nodes in the network. The main idea of delegation forwarding is that it assigns a quality and a level value to each node. The quality value of a node here is $S(X,Y)$ and the level value is τ. Initially, the level value of each node is equal to its quality value. During the routing process, a message holder compares the quality of the node it meets with its own level. It only forwards the message to a node with a higher quality than its own level. In addition, the message holder raises its own level to the quality of the higher quality node. The result of delegation forwarding is that a node will forward a message only if it encounters another node whose quality metric is greater than any seen by the node so far.

B. Social Similarity Metrics $S(X,Y)$

To evaluate the similarity of two nodes in a more accurate way, we look at the nodes' past meeting ratios. Each individual node X has a vector of length m: $\langle x_1, x_2, \cdots, x_m\rangle$, where $x_i = \frac{M_i}{M_{total}}$. That is,

$$< x_1, x_2, \cdots, x_m >= \left\langle \frac{M_1}{M_{total}}, \frac{M_2}{M_{total}}, \frac{M_3}{M_{total}}, \cdots \frac{M_m}{M_{total}} \right\rangle \quad (1)$$

where M_i is the number of meetings of X with nodes whose social feature F_i is the same as the destination's feature F_i, and M_{total} is the total number of meetings of X with any other node in the history we observe. Thus $0 \leq x_i \leq 1$ for all $1 \leq i \leq m$. With the node's vector defined, the next task is to use similarity metrics to compare the similarity of two vectors. Now the heuristic for selecting the best forwarder in routing becomes the selection of the node whose vector is most similar to that of the ideal forwarder R for the destination node. An

Algorithm SOSIM: routing algorithm using SOcial SIMilarity metrics

Let $B_1, B_2, \cdots, B_n$ be nodes. Each node has a vector of $\langle B_{i1}, B_{i2}, \cdots, B_{im}\rangle$ $(1 \leq i \leq n)$
R is an ideal node who meets nodes like D all the time and has a vector of $\langle 100\%, 100\%, \cdots, 100\%\rangle$
INITIALIZE $\forall i : \tau_i \leftarrow S(B_i, R)$
On contact between message holder B_i and node B_j:
if B_j is the destination D **then**
  B_i forwards the message to B_j and the algorithm is terminated
else if $\tau_i < S(B_j, R)$ **then**
  $\tau_i \leftarrow S(B_j, R)$
  if B_j does not have the message **then**
    B_i forwards the message to B_j
  end if
end if

Fig. 1. The DTN routing algorithm based on social similarities

ideal forwarder R is a theoretical node who meets people like destination D all the time. Hence its vector consists of m 100%s: $\langle 100\%, 100\%, \cdots 100\%\rangle$.

The various similarity metrics we use in our routing algorithm are derived from those in data mining [3]. In our metrics, 1 means 100% identical and 0 means not similar at all. To deal with various values of social data, we normalize the outputs of all metrics to the range of $[0, 1]$. The details are as follows:

1) Tanimoto Similarity: The Tanimoto coefficient to measure the similarity of X and Y is:

$$S(X, Y) = \frac{X \cdot Y}{X \cdot X + Y \cdot Y - X \cdot Y} \tag{2}$$

where $X \cdot Y$ is the dot product of the two vectors.

For example, suppose we look at three social features: *City, Language,* and *Position* in the network. If the values of the social features of destination D are: $\langle NewYork, English, Student\rangle$ and node X has met people from New York 70% of the time, people that speak English 93% of the time of the time, and students 41% of the time in the history we observe, then node X has a vector of $X = \langle 0.7, 0.93, 0.41\rangle$. And an ideal forwarder R for D should have a vector of $R = \langle 1, 1, 1\rangle$. Using the Tanimoto metric in equation (2), $S(X, R) = 0.82$.

2) Euclidean Similarity: We can also use the Euclidean distance to measure a node's social similarity to another node. To make the similarity definition consistent, we normalize the original definition of Euclidean similarity to the range of $[0, 1]$ and subtract it from 1. Now the Euclidean similarity of X to Y is defined as:

$$S(X, Y) = 1 - \frac{\sqrt{\sum_{i=1}^{m}(y_i - x_i)^2}}{\sqrt{m}} \tag{3}$$

3) Weighted Euclidean Similarity: In addition to the basic Euclidean similarity mentioned above, we also employ the weighted Euclidean similarity. To determine the weight of a social feature, we use the Shannon entropy [9] which quantifies the expected value of the information contained in the feature [13]. The Shannon entropy for a given social feature is calculated as:

$$w_i = -\sum_{i=1}^{k} p(f_i) \cdot log_2(f_i) \tag{4}$$

where w_i is the Shannon entropy for feature F_i, $\langle f_1, f_2, \cdots f_k\rangle$ are the possible values of feature F_i, and p denotes the probability mass function of F_i. The weighted Euclidean similarity normalized to the range of $[0, 1]$ is as follows:

$$S(X, Y) = 1 - \frac{\sqrt{\sum_{i=1}^{m} w_i \cdot (y_i - x_i)^2}}{\sqrt{\sum_{i=1}^{m} w_i}} \tag{5}$$

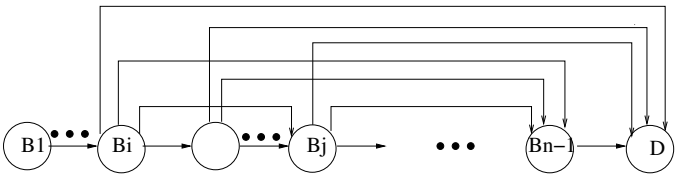


Fig. 2. The routing process

IV. Analysis

In this section, we show that our algorithm can improve routing performance over the social-feature-based algorithm with a low implementation cost.

Our social-similarity-based delegation routing process can be described by Fig. 2. $B_1, B_2, \cdots, B_{n-1}$ represent nodes. Assume B_1 is the source and node D is the destination. The solid arrow between two nodes represents that they directly meet each other. From B_1 to B_{n-1} on the horizontal path in Fig. 2, the similarity of the node to the ideal node R who meets nodes like the destination all the time is increasing and the possibility to meet the destination is also increasing. With each hop of forwarding, the message gets to a node that is more similar than previous to the ideal forwarder of the destination.

The latency to deliver a message from a node B_i to the destination D can be expressed as:

$$L_{B_iD} = min(J_{B_iD},\ L_{B_iB_{i+1}} + J_{B_{i+1}D},\ L_{B_iB_{i+2}} + J_{B_{i+2}D}, \cdots,\ L_{B_iB_j} + J_{B_jD}, \cdots,\ L_{B_iB_{n-1}} + J_{B_{n-1}D})$$

where J_{XY} represents the latency if node X directly meets node Y, and L_{XY} refers to the latency needed for the message to go from X to Y through direct meeting of the two or through some intermediate nodes. In other terms, the formula means that the latency of message from B_i to D is the minimum latency of the following: B_i meets D directly, the message is delivered to B_{i+1} from B_i and then B_{i+1} meets D directly, the message is delivered to B_{i+2} from B_i and B_{i+2} meets D directly, and so on.

Now take any two nodes B_i and B_j $(i < j)$ on the horizontal path from B_1 to D. If they have the same social feature values as D, for example, both of them are New Yorkers and Students, but B_i lives in Texas and B_j lives in New York. In the existing

social-feature-based routing algorithm, both of their vectors are $\langle 1, 1 \rangle$ and they are not distinguishable. In our algorithm, node B_j is closer in similarity to the ideal forwarder R of D because it meets New Yorkers more. If it is chosen as the next forwarder, then it is more likely to reduce the latency and has a higher chance to deliver the message to the destination. Also, our algorithm can be implemented with a low cost if a node uses m counters (for m features) to record the number of meeting times with other nodes having the same value in each feature and a counter for the total number of meetings in the observed time period to calculate the meeting ratios.

V. Simulations

This section describes the simulations we conducted using a custom simulator written in Java. We first performed simulations to select the social similarity metric to use for our protocol. We then compared our algorithm with the existing ones. Finally, we showed that our protocol performs well on sparse networks.

For all of our simulations, we used the INFOCOM 2006 trace [8]. This data set consists of two parts: contacts between the iMote devices that were carried by participants and the self-reported social features of the participants. The six features we used for all social protocols were *Affiliation, City, Nationality, Language, Country,* and *Position*.

In our simulations, we utilized the first two days of the data as the initial history for SOSIM and performed our simulations on the remaining three days. We generated messages from a randomly chosen source to a randomly chosen destination every two seconds in the first 24 hours of the simulation. We then averaged five separate simulations of each algorithm with identical setups to mitigate the effect of any outliers in the performance. To perform a fair comparison of the algorithms, the time to live of all packets (except for those created in the Flooding simulation) was set to 9, meaning that a given packet can be transferred at most nine times.

In order to compare the routing strategies, we define three important metrics to evaluate their performance:

1) *Delivery ratio*: The fraction of generated messages that are correctly delivered to the final destination since the beginning of the simulation.
2) *Delivery latency*: The time between when a message is generated and when it is received.
3) *Packet duplication*: The number of duplications needed to deliver a message to its destination.

Efficient routing entails a high delivery rate and low latency with an acceptable number of duplications.

A. Social Similarity Metrics Comparison

To find the best fit for our simulated context, we compared Tanimoto, Euclidean, and Weighted Euclidean social similarity metrics by performing delegation forwarding based on $S(X, Y)$ for each algorithm. Results in Fig. 3 show that all of the metrics performed similarly in delivery ratio, latency, and duplication because their lines are overlapped. With the Weighted Euclidean Similarity metric, we hoped to make the protocol favor the social features of the destination that were more influential to the delivery of the packet through the use of weights. However, this approach seemed to have negligible significance in our simulated environment. We therefore decided to use the Euclidean metric since it did not require the calculation of additional weighting values and performed slightly better than the Tanimoto similarity in latency.

B. Comparison with Existing Algorithms

Simulations were conducted to compare our algorithm with the social-feature-based algorithm. The Flooding and Wait algorithms were included as benchmarks in the comparison.

1) *The Flooding Algorithm* (Flooding): Every message is spread epidemically throughout the network until it reaches its destination.
2) *The Wait Algorithm* (Wait): The source holds the message until it meets the destination.
3) *The SOSIM Algorithm* (SOSIM): Our algorithm, using the Euclidean distance as the social similarity metric.
4) *The Social-Feature-based Algorithm* (Social): This algorithm takes the idea from [13] and converts it into a single-copy delegation forwarding scheme for fair comparison where routing is guided by resolving social feature differences between source and destination.

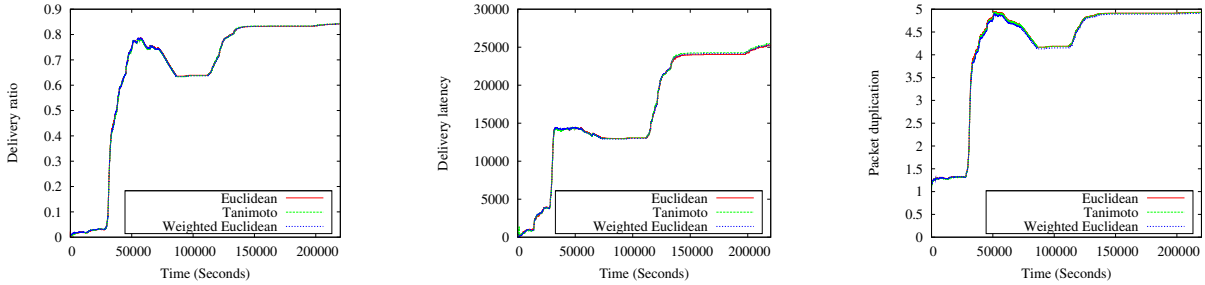
The results in Fig. 4 show that, as expected, Flooding has the highest delivery ratio and lowest delivery latency but highest number of packet duplications. Wait has the lowest number of packet duplications but lowest delivery ratio and highest delivery latency. SOSIM outperforms Social in delivery ratio and latency with a little increase in the number of duplications. The little increase in the number of duplications indicates that SOSIM is more active in delivering the message to better forwarders because it can more accurately identify them. The simulation results confirm our analysis.

C. Sparse Networks

In the experiments above, we used all 62 available nodes in the INFOCOM 2006 trace. We also tested our algorithm on a smaller random subset of the trace with 16 nodes which results in a sparse network. The results in Fig. 5 are consistent with those in the denser network.

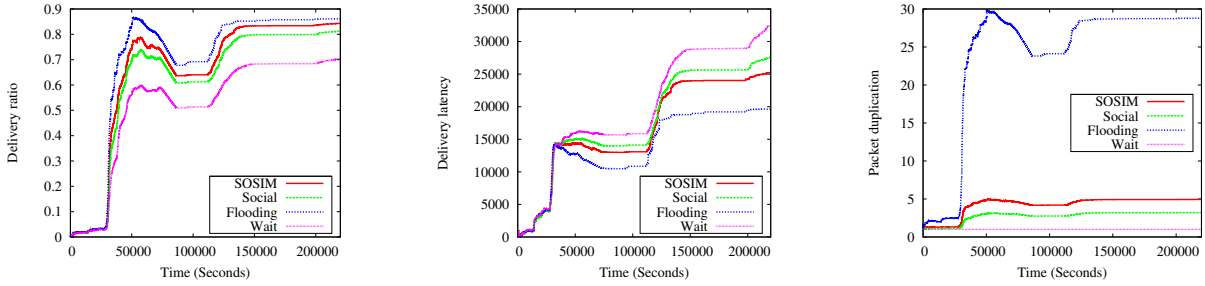
VI. Conclusion

In this paper, we have proposed a novel algorithm named SOSIM for DTN routing that uses similarity metrics on nodes' contact history to more accurately evaluate the social similarity between nodes and guide routing towards the destination. The analysis has indicated that our algorithm can improve the performance of DTN routing with a low implementation cost. The simulation results using real trace have also shown that our algorithm outperforms the existing social-feature-based algorithm. In our future work, we plan to test our algorithm using more traces with social features as they become available and to find better metrics to improve routing efficiency.



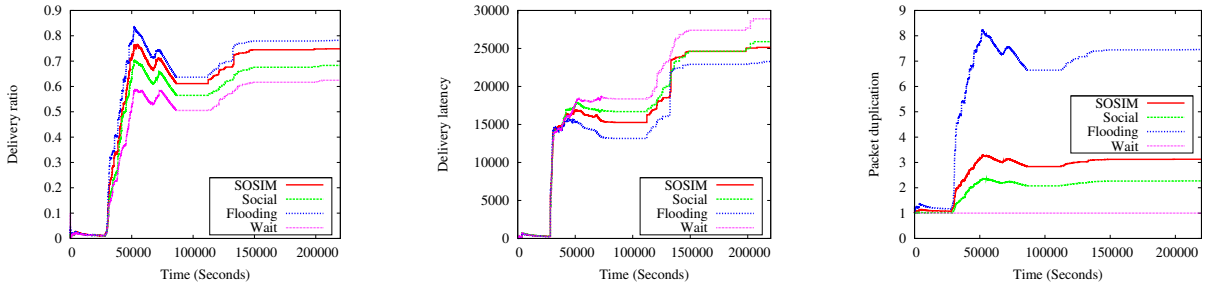
(a) Delivery ratio (b) Delivery latency (c) Packet duplication

Fig. 3. Comparison of Euclidean, Weighted Euclidean, and Tanimoto social similarity metrics



(a) Delivery ratio (b) Delivery latency (c) Packet duplication

Fig. 4. Comparison of SOSIM with Flooding, Social, and Wait algorithms using all devices in the trace



(a) Delivery ratio (b) Delivery latency (c) Packet duplication

Fig. 5. Comparison of SOSIM with Flooding, Social, and Wait algorithms on sparse networks

ACKNOWLEDGEMENTS

This research was supported in part by the Department of Defense in partnership with the NSF REU grant 1156712.

REFERENCES

[1] J. Burgess, B. Gallagher, D. Jensen, and B.N. Levine. Maxprop: Routing for vehicle-based disruption-tolerant networks. In *Proceedings of IEEE INFOCOM*, volume 6, pages 1–11. Barcelona, Spain, 2006.

[2] V. Erramilli, M. Crovella, A. Chaintreau, and C. Diot. Delegation forwarding. In *Proceedings of the 9th ACM international symposium on Mobile ad hoc networking and computing*, pages 251–260, 2008.

[3] J. W. Han, M. Kamber, and J. Pei. *Data Mining: concepts and techniques*. Morgan Kaufmann, MA, USA, 2012.

[4] E. P. Jones and P. A. Ward. Routing strategies for delay-tolerant networks. *Proceedings of ACM SIGCOMM*, 2004.

[5] A. Mei, G. Morabito, P. Santi, and J. Stefa. Social-aware stateless forwarding in pocket switched networks. In *Proceedings of IEEE INFOCOM*, pages 251–255, 2011.

[6] A. Pentland, R. Fletcher, and A. Hasson. Daknet: rethinking connectivity in developing nations. *Computer*, 37(1):78–83, 2004.

[7] J. Scott, J. Crowcroft, P. Hui, and C. Diot. Haggle: a networking architecture designed around mobile users, 2006.

[8] J. Scott, R. Gass, J. Crowcroft, P. Hui, C. Diot, and A. Chaintreau. Crawdad trace cambridge/haggle/imote/infocom2006 (v.2009-05-29), May 2009.

[9] C. Shannon, N. Petigara, and S. Seshasai. A mathematical theory of communication. *Bell System Technical Journal*, 27(1):379–423, 1948.

[10] T. Spyropoulos, K. Psounis, and C.S. Raghavendra. Single-copy routing in intermittently connected mobile networks. In *IEEE SECON*, pages 235–244, 2004.

[11] T. Spyropoulos, K. Psounis, and C.S. Raghavendra. Spray and wait: an efficient routing scheme for intermittently connected mobile networks. In *Proceedings of the 2005 ACM SIGCOMM workshop on Delay-tolerant networking*, pages 252–259, 2005.

[12] A. Vahdat, D. Becker, et al. Epidemic routing for partially connected ad hoc networks. Technical report, CS-200006, Duke University, 2000.

[13] J. Wu and Y. Wang. Social feature-based multi-path routing in delay tolerant networks. In *Proceedings of IEEE INFOCOM*, 2012.

[14] J. Wyatt, S. Burleigh, R. Jones, L. Torgerson, and S. Wissler. Disruption tolerant networking flight validation experiment on nasa's epoxi mission. In *Procceedings of SPACOMM*, pages 187–196, 2009.