# Probabilistic Routing With Multi-copies in Delay Tolerant Networks

Ze Li and Haiying Shen

Department of Computer Science and Computer Engineering
University of Arkansas, Fayetteville, AR 72701
{zxl008, hshen}@uark.edu

## Abstract

*Intermittently connected mobile networks don't have a complete path from a source to a destination at most of the time. Such an environment can be found in very sparse mobile networks where nodes meet only occasionally or in wireless sensor networks where nodes sleep most of the time to conserve energy. Current approaches in such networks are primarily based on two kinds of transmissions: multi-copy flooding scheme and single-copy forwarding scheme. However, they have either high overheads due to excessive transmission or long delays due to the possible incorrect choices during forwarding. In this paper, we propose a hybrid probabilistic routing algorithm using multi-copies called HUM, in which a packet is initially replicated to a certain number of nodes, which sequentially forward those packets to the destination node based on a probabilistic routing scheme. Simulations show that compared to Epidemic routing, Spray and wait routing, HUM routing scheme provides a nearly optimal delay performance with a stable packet arrive rate with the community mobility model.*

.

## 1 Introduction

With the development of the techniques such as IEEE 802.11, and other radio solutions (e.g. low power radios designed for use in sensor networks), it has become viable to equip almost every device with wireless networking capabilities. Therefore, it is very easy to establish a network even without networking infrastructure. Wireless ad-hoc network is such kind of network that has received much attention recently. In a wireless ad-hoc network, packets can be forwarded by the intermediate nodes if a pair of source node and a destination node can not communicate directly. However, one of the most basic requirement for the wireless ad-hoc network is that a continuous end-to-end path between source and destination nodes should always exit. However, there are a lot of extreme environments scenarios, where node communications are intermittently connected such as wildlife monitoring sensor networks [1], interplanetary communication networks [2], vehicular ad hoc networks (VANETs) [3], terrestrial wireless networks, ocean sensor networks [4, 5], satellite networks, and etc. In these networks, the intermittent connectivity can be a result of mobility [3], power management [1], wireless range, sparsity [6], or malicious attacks [7].

Therefore, conventional internet routing protocols (e.g. RIP, OSPF) as well as ad-hoc network routing schemes, such as DSR [8], and AODV [9] that assume a complete existing path between a source and a destination, and try to discover minimum cost paths before data is sent out are not applicable. Because of such unique characters of the delay tolerant network (DTN), a new routing concept which consists of a sequence of independent, local forwarding decision, based on current connectivity information and predictions of future connectivity are widely studied. In this paper, the term "intermitted connected network" and "DTN" are interchangeable used. The DTN structure has been proposed recently [10]. It defines an end-to-end message-oriented overlay called "bundle layer" that exists at a layer above the transport layers of the networks and below applications. Packets are transformed by the bundle layer into one or more protocol data unites called "bundles", which are forwarded by mobile nodes. Such a structure is adopted in our approach. The DTN architecture does not expect that network links are always available or reliable, and instead expects that nodes may choose to store bundles for some time. We anticipate that most DTN nodes will use some form of persistent storage for this– disk, flash memory, etc, but in this paper we assume the packet are stored in the buffer to simplify the analyze.

A number of routing schemes proposed for DTN are flooding-based [1, 11, 12]. Despite they increase robustness and low delay of the transmission, flooding-based protocols consume a high amount of energy, bandwidth, and memory space which are crucial to the wireless network applications. In addition, under high traffic loads, they suffer from severe contention and packet drops that can significantly degrade their performance and scalability. On the other hand, a lot of single-copy based routing schemes have been proposed [13] for the DTN. However, such kind schemes will suffer from severe transmission delay if the node choose a wrong path for the delivery.

In this paper, we present the design, implementation and evaluation of a hybrid probabilistic routing scheme using multi-copies (HUM) to leverage the previous two methods. Taking advantages of current connectivity information and predictions of future connectivity information, HUM "stores and forwards" the packet to the destined node in a distributed manner. The basic idea of HUM is that it initially and randomly replicates a packet to a certain number of its neighbor nodes. These nodes will independently "store and forward" the packet copies to another node that has a higher utility (the possibility to meet the destination node) for the packet's destined node. This process will be repeated until one of the packet copies arrives at the destination. Fundamentally, the benefits of HUM is that the packets transmission are over multiple relays with a fixed amount of overhead. It makes the transmission much more robust to failures of a few relays or some bad choices, leading to high performance. Moreover, based on buffer management approaches, HUM outperforms other flooding based schemes in respect to the delay performance, congestion preclusions, and packet receiving rate. Spyropoulos *et al.* [12] also proposed a hybrid multi-copies routing method called Spray and Wait (SW). There are two main differences between HUM and SW: (1) HUM uses possibilities routing after replicate phrase while SW uses direct routing after that. (2) Several buffer management approaches are implemented in HUM while SW just uses a simple $TTL$ scheme to management buffer. Simulations results confirm that HUM improves the performance of SW in terms of number of received packets and transmission delay.

In the next section we go over existing related work. Section 3 presents HUM routing algorithm. Simulation results are presented in section 4. Finally, section 5 concludes the paper.

## 2 Related Work

Although numerous routing protocols for wireless ad-hoc networks have been proposed [9, 14, 8], traditional routing protocols are not appropriate for DTNs that are sparse and disconnected. These protocols don't work well even if the network was only "slightly" disconnected [13].

The intuitive ideas to deal with connectivity disruptions in DTNs is to reinforce connectivity on demand by sending out a number of specialized nodes (e.g. robots, satellites) which are assigned to move to fill the "communication gap" when a disconnection happens [15, 16]. However, such kind of networks are not applicable in a highly dynamic self-organized networks needed to be "fixed" at all time.

Probabilistic routing is another approach for the DTN. In [17, 2, 6], the routing path has been figured out before transmission. In [1], the nodes record the history of past encounters in order to make fewer and more informed decisions. Those routing paths are predicted either by statistics of the mobility module or by the historical moving path record. Unfortunately, these schemes can only reduce the transmission overhead of Epidemic routing at a significant penalty on delivery delay.

The third approach for the DTN is opportunistic routing. A simplest approach is direct routing that lets the source or a moving relay node carry the message all the way to the destination [18]. Although these schemes can achieve high throughput performance, the delay will also be very long especially if base stations are sparse in the system. A faster way to perform routing in DTN is flood routing which is also called Epidemic routing [19]. This scheme can guarantee a short delay by finding a shortest routing path, but it costs a lot of network resources. There are some improved approaches for the Epidemic routing to reduce its overhead and enhance its performance [20, 11, 21–23]. In [20], a message is "gossiped" to other nodes instead of flooding, which means the message is forwarded to only some of the neighbors. In [22], nodes will clean up redundant copies of certain message when that message has been transmitted. In [11], the author points out that consulting the age of the last node encounter when making forwarding decision results in superior performance than flooding. Network coding [23] and [21] have been used to improve the performance of the flood routing. Although all these schemes can improve the performance of Epidemic routing to a certain extent, they still inherits the shortcomings of the flooding routing and can not significantly improve the delay performance.

## 3 The HUM Protocol

In this section we describe the HUM protocol. We start off by describing the goals of HUM design and then discuss the approaches to achieve the goals. We will also discuss various aspects of the protocol in details.

To be an optimized protocol for DTN, HUM has the following goals and corresponding approaches. (1) In order to improve the delay performance of existing single-copy scheme, making the transmission more robust to failures of relays or bad choices, HUM replicates a certain number copies for the transmission. (2) In order to improve the delay performance in forwarding, HUM adopts probabilistic routing to "guide" a packet forwarding in a correct path. Even in a situation that the nodes carrying packets fail to make right forwarding choices at all the time, HUM just degenerate to "Spray and Wait"scheme [12] which has been proved to outperform some current routing methods. (3) In order to improve the performance in the high loaded system, HUM uses buffer management methods to efficiently manage the buffer.

Since modelling a system with many interactions is

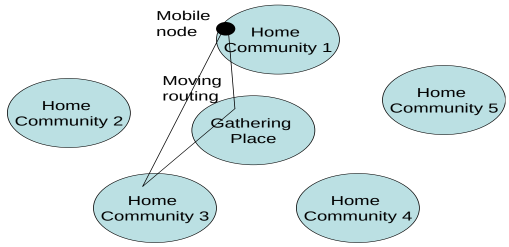



**Figure 1. Community mobility model**

complicated, we model the replication of a single packet at a time as an example to explain the packet replication processes of HUM. Because the replication of one unique packet does not affect the replication of another packet, all the transmissions can be regarded as states in a Markov chain. HUM routing spreads a number of copies generated by each source node and therefore a number of transmissions can be performed in network size wide. Specifically, HUM has three phases listed below:

(1) Replicate phase: For every packet originating at a source node, $N_c$ packet copies are initially replicated to $N_c$ distinct random chosen neighbors. If the destination node is among these neighbor nodes, the transmission is completed. Otherwise, it goes to the forwarding phase.

(2) Forwarding phase: For every node in the system, it will hold a utility vector that record the meeting possibility value for every other node it has met. The utility vector indicates how likely this node will be able to meet other nodes. Based on these utility information, each copy of the packet is forwarded to the node with a higher utility of the destination node until one of those copies arrive at the destination node.

(3) Clear phase: After the transmission is completed, the destination node sends a message or piggybacks it on other packets back to the system to delete the offloaded packets in their buffers. This message includes the identifiers of the offloaded packets received by the destination nodes. The details of the buffer management will be discussed below.

### 3.1 Mobility models

The traditional popular simulation scenarios such as random walk, random way-point model assume that each node may move equally frequently to every network location with identical, and independently distributed mobility process. However, numerous recent studies based on mobility traces from real networks (e.g. university campuses, conferences, etc.) have demonstrated that these two assumptions rarely hold in real-life situations [13]. For this reason, we will simulate HUM under a more realistic mobility model, called "Community-based Mobility Model" that better resembles real node movement [24]. we used a scenario that is very similar to the one used in [11] as a reference. In the community mobility model, a node selects a destination and moves to it by a selected speed, and then repeats this process. If node is at home community, it will go to a gathering place (e.g. in reality it can be a mall for the people, feeding ground for the animals) with a high possibility , but it can still go to other places. If it at a gathering place, it is very likely that the next destination of the node is the home. Moreover, if the node is in other places, it will definitely go back to the home community.

### 3.2 Delivery utility calculation

There are serval metrics that can be used to determine a node's transmission utility to others according to different systems based on different mobility models [25]. In this paper, since we adopt the community model for the HUM, we devise a new utility called the meeting possibility to lead the transmission of the packets in the forwarding phrase. The calculation of the delivery possibilities has three parts. The first thing to do is to update the metric whenever a node is encountered, so that nodes that are often encountered have a high delivery predictability. The calculation is shown below.

$$P_{(i,j)} = \frac{T_{(i,j)}}{T_{(i)}},$$

where $P_{(i,j)}$ denotes the utility of node $i$ meets node $j$, $T_{(i,j)}$ is the total meeting time between $node_{(i)}$ and $node_{(j)}$ in a time interval $T(i)$. $T(i)$ denotes a time period between $node_{(i)}$ leaving home community in consecutive two times. For example, the first time when $node_{(i)}$ leaves home community is at $t_0$, and the second time it leaves at $t_1$, then $T(i) = (t_0 - t_1)$. However, when calculating the current utility of the node, the old utility value of the node should be also brought into account, since the latest acquiring utility may not accurately reflect the meeting probability of two nodes because of the signal interference or link break. Therefore, every time when nodes leave home, they will recalculate $P_{(i,j)}$ as

$$P_{(i,j)_{current}} = \alpha P_{(i,j)} + (1-\alpha) P_{(i,j)_{new}}$$

and then clear $T_{(i,j)}$ and $T_i$ in order to record the next round meeting possibility $P_{(i,j)_{new}}$, where $\alpha$ is a weighing constant ($\alpha \in (0,1)$). The packets forwarding are based on the value $P_{(i,j)_{current}}$.

### 3.3 packet replications

HUM uses replications to increase the probability that a packet copy will be offloaded to a destination node. However, too much replicas will result in traffic congestion or unnecessary node energy consumption. Therefore, in the HUM, a small number of copies are replicated in the system for the packet transmission. The tradeoff number of copies is depended on the system size, mobility model or number of nodes in the system [12]. Suppose that there are $N_c$ packet copies in the system. It is intuitive that the delays of the packet can be reduced if the packet is replicated as fast as possible, and the replication process can just stop after $N_c - 1$ copies are presented. Therefore, an approach is needed

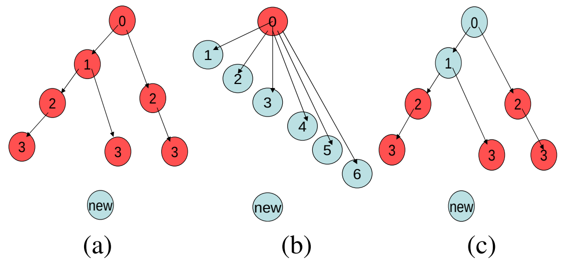



**Figure 2. Packets replication tree**

to ensure a node knows the number of replicated packets. A replicate distributing approach called locally-optimal tree described in [22] is adopted by HUM to replicate the copies. Figure 2(a) shows that at time $t = 0$, $node_{(1)}$ knows there are $N_c$ copies required in the system. Thus, when $node_{(1)}$ comes across $node_{(2)}$, it entitles $node_{(2)}$ to replicate $\frac{N_c-1}{2}$ copies to other nodes while itself remains $\frac{N_c-1}{2}$ transmissions. Sequentially, if $node_{(2)}$ meets $node_{(3)}$, each node will be entitled to have half of the remaining transmissions, i.e. $\frac{N_c-2}{4}$. The process is continuous until each node has only one copy of the packet. Compared to figure 2(b): source tree algorithm (only source node can replicate copy to other) which need $O(N)$ time steps to replicate the copies, optimal tree algorithm only needs $O(\log_2 N_c)$ time steps. Meanwhile, comparing Figure 2(a) and Figure 2(c), we can see that the optimal tree algorithm can replicate the packet to another node with possibility as 1 at all time, while the binary tree algorithm only has a possibility of $5/7$ in this case. Therefore, although the binary routing tree algorithm (each node can only replicate copies to two other nodes) also has a replication time step in the order of $O(\log_2 N_c)$, the replication process is still slower than the optimal tree algorithm.

### 3.4 Buffer Management

Buffer size is a critical issue for many networks such as wireless sensor network. Network performance will deteriorate when highly loaded, because the limited buffer size of each node will lead to serious traffic congestion. In HUM, several buffer management methods are adopted to release the burden of the buffer which sequentially increase the throughput of the network without increasing the transmission delay.

#### 3.4.1 Manage Copies in the Buffer

In 3.3, a packet is replicated for $N_c$ nodes with different utility values to the destined node. However, since the replicated nodes are randomly chosen, some selected nodes may have low utilities to the destination node. In order to save the buffer, the buffer slots should be always assigned to more promising packets. Therefore, a utility threshold is assigned to the buffer to classify the packets into $corecopy$ and $backupcopy$ categories. If the utility of the packet's destination node is lager than the threshold, the packet is regarded as a $corecopy$, otherwise a $backupcopy$. Although Pasztor [26] also proposed a classifying method, that method is not applicable here. When two nodes exchange their utilities information, they also exchange the information of the number of the empty buffer slots and number of backup copies in their buffers [26]. Packets copies forwarded in other nodes firstly use the available empty slots and then overwrite the slots of the backup copies. Since $backupcopy$ has a lower possibility to meet destination node in a short time, and the "core copies" of a packet are still remain in the system, replacing "backup copies" in the buffer will not degrade the delay performance of transmission.

#### 3.4.2 Deleting Copies in the Buffer

Backup and core packet management method can significantly release the burden of buffer. However, when the buffer is full of "core copies", some promising packets from other nodes will still be dropped. Therefore, we also propose a copy deleting approach to guarantee the robustness of the transmission.

When a node is about to load off packets to the destination node, the destination sends a hash table containing the identifiers of the received packets to that node. Then the node deletes the packet indicated in the hash table and forwards all the packets to the destination node. After receiving the new packets, the destination node sends the identifiers of those new received packets back into the system or piggybacks them on other packets. The nodes that receive the identifers will delete the packets indicated in the list and exchange them with other neighbor nodes when they meet in order to get rid of the unnecessary packets.

#### 3.4.3 Maintain the Forwarding Sequence

Since the transmission between two mobile nodes happens only when they are within transmission range of each other, the transmission time is limited especially in a high dynamical situation. Therefore, which packet should be transmitted at first will affect the performance of the system. In HUM, except traditional field in the packet head such as $sourcenodeID$, $destinationnodeID$, and etc, another two new fields are included in packet head, $[priority, time\_stamp]$. Each node $i$ maintains a separate queue of packets sorted in a decreasing order of priority and value of $time\_stamp$. Priority is used to differentiate traffic based upon an application's desire to affect the delivery urgency for packets. $time\_stamp$ is used to record time since packet creation. $time\_stamp$ and $sourcenodeID$ together can be used to identify each packet. When two nodes meet in the system, the packet with the highest priority will be delivered at first. The reason why the packets are sent in the order of $time\_stamp$ value is to avoid the long delay in the communication. The longer time a packet stay in the system, the higher relay priority it will have.

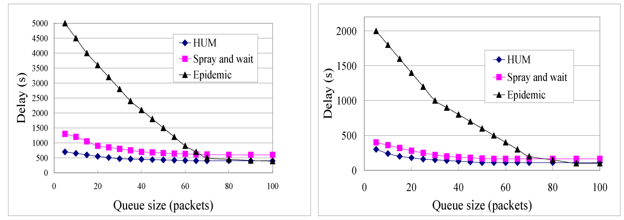
# 4 Performance Evaluation

This section demonstrates the distinguishing properties of HUM through simulation built on a custom discrete event-driven simulator [13] in comparison with Epidemic routing [19], and Spay and wait routing [12]. We used Community model what we have defined in 3.1. This scenario consists of a $500 \times 500m$ area where $100$ nodes are identical, independent distributed placed. Every $10$ nodes share a home community. The mobile nodes move in the scenario with speeds of $0 - 20m/s$. Each node is generating a new packet for a randomly selected destination with transmission rate $1packet/5s$ for $2000s$. The hop count of the Epidemic routing is $5$ hops. the number of replicate copy of HUM and Spray and wait routing is $16$. When a node is in the home community, it will then go to the gathering place with possibility of $0.8$ and go to other place with possibility of $0.2$. When a node is in gathering place, it will then go home with possibility of $0.5$, other place with possibility of $0.5$. When node is at other places, it will go back to home community directly. All results are average over $5$ runs. A warm up period of $500s$ is used in the beginning of the simulations to initialize the utility of HUM.

We used two metrics in the simulation: (1) Packet delivery ability, i.e. the number of the packets that is able to be delivered to the destination. (2) Packet delivery delay, i.e. The average time that it takes a packet to be delivered. Simulation results demonstrate that the HUM routing scheme outperforms the Epidemic routing and Spray and wait routing in terms of transmission delay and number of received packets.

### 4.1 Comparison of Packet Delivery Delay

Figure 3(a) and figure 3(b) present the Packet delivery delay with different transmission ranges. We can notice that the transmission delay is greatly reduced with the increase of the transmitting range of the node. The reason is that a larger transmission range makes it easier to find other neighbor nodes which may be the potential receivers or promising relay nodes. Moreover, the speed of the electromagnetic wave moves much faster than the moving node, thus data transmission with larger transmission range is faster. The figures also show that with the increase of the buffer size, transmission delay is decreased. It is because a larger buffer size enables more packets to be buffered, thus packets have low possibility to be thrown away. The figure also indicates HUM has the best delay performance when a node has a small buffer size. The reason is with the buffer management approaches, the transmission in the HUM will not be significantly affected by the buffer size. Although in HUM routing scheme, a packet is replicated to several nodes as Spray and wait scheme does, some buffer slots taking up by the backup copies can be replaced by other promising packets if the buffer is full, and the core packets with the highest possibility to be delivered still remain in the system. However, Relying on flooding,




(Range = 20m) (Range = 40m)

**Figure 3. Transmission delay**

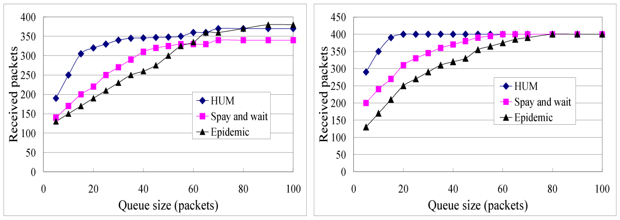
the Epidemic routing will inevitably suffer from severe congestion. Moveover, figure 3 shows HUM still outperforms Spray and wait routing in a low loaded system where the nodes are equipped with large buffer size. It is due to the reason that in the forwarding phrase, HUM uses a probabilistic routing while Spray and wait just adopts direct routing which may lead to a little more delay as $O(\frac{N}{N_c})$. The factor that delay of the HUM is almost the same as Epidemic routing with a large buffer size, which is the lower bound of the delay performance [19] can also be found in the figure.

### 4.2 Comparison of Packet Delivery Ability

Figure 4 shows that as the queue size increases, so does the number of the packets delivered to their destination. A larger queue size means more packets can be buffered, and the possibility that a packet is thrown away decreases. Therefore, the number of packets received by the destination node will increase. The figure also shows that the HUM is less sensitive to the queue size changes than Epidemic routing for its significant buffer management approaches, and the Epidemic routing with flooding nature is still manifest to suffer from the congestion severely especially under high load. Figure 4 also indicates in a low load, where the butter is large enough for all the packets, the transmission performance of HUM can still reach the upper bound as Epidemic routing does. The reason why the number of received packets is decreased with the decrease of transmission range is the shorter transmission range of the node results in a smaller possibility of nodes to meet neighbors. Meanwhile, in the Spray and wait routing scheme, the forwarding phase is direct transmission with $TTL$ of the packet calculated by time. Therefore, it is very likely that the packet will be dropped when $TTL$ expires. That is why it is so obvious in the figure that Spray and wait suffers more than other routing schemes from the decrease of the transmission range.

# 5 Conclusion

Traditional routing scheme in wireless ad-hoc networks can not achieve a good performance in Delay tolerate networks (DTN), since DTN can not guarantee a end-to-end link established all the time. Epidemic routing and Spray and wait routing are two representative routing schemes for DTN. The former is based on flooding routing for the packets transmission and the




(Range = 20m) (Range = 40m)

**Figure 4. Number of Received Packets**

latter is a hybrid routing which compounds direct routing and multi-copies routing together. In this paper, we proposed a hybrid probabilistic routing algorithm using multi-copies for DTN, namely HUM. HUM replicates a new packet to a certain number of nodes. These nodes will hold the copies until they meet another node with a higher utility for the packet's destination and then forward the copies to them. This process will be repeated until one of the packets arrives at the destination. Simulation results based on community mobility model show that HUM outperforms the Epidemic routing scheme and Spray and wait scheme in terms of packets delivery rate and transmission delay. In the future work we intend to implement HUM to other mobility modules to see whether HUM can be applied to other famous model such as random walk model and random way point model. We also plan to do some theoretical work to find out the optimized number of replication copies that can be generated in best number of replicated copies for each model.

## References


[1] P. Juang, H. Oki, M. Martonosi Y. Wang, L. S. Peh, and D. Rubenstein. Energy-efficient computing for wildlife tracking: design tradeoffs and early experiences with zebranet. In *Proc. of ASPLOS*, 2002.

[2] S. Burleigh, A. Hooke, L. Torgerson, K. Fall, B. Durst V. Cerf, and K. Scott. Delay-tolerant networking: an approach to interplantary internet. *IEEE communication magzine*, 2003.

[3] H. Wu, R. Fujimoto, R. Guensler, and M. Hunter. MDDV: Mobility-centric data dissemination algorithm for vehicular networks. In *Proc. of ACM on VANET*, 2004.

[4] J. Partan, J. Kurose, and B. N. Levine. A survey of practical issues in underwater networks. In *Proc. of WUWNet*, 2006.

[5] A. Maffei, K. Fall, and D. Chayes. Ocean instrument internet. In *Proc. of AGU*, 2006.

[6] S. Jain, K. Fall, and R. Patra. Routing in a delay tolerant network. In *Proc. of ACM SIGCOMM*, 2004.

[7] Delay tolerant networking research group. http://www.denrg.org.

[8] D. B. Johnson and D. A. Maltz. Dynamic source routing in ad hoc wireless networks. *IEEE Mobile Computing*, 1996.

[9] C. Perkins, E. Belding-Royer, and S. Das. RFC 3561: Ad hoc on demand distance vector (AODV) routing, 2003.

[10] V. Cerf, S.Burleigh, A. Hooke, L. Torgerson, R. Durst, and K. Scott. Delay-tolerant networking architecture. *RFC4838*, 2007.

[11] A. Lindgren, A. Doria, and O. Schelen. Probabilistic routing in intermittently connected networks. *SIGMOBILE Mobile Computing and Communication Review*, 2003.

[12] T. Spyropoulos, K. Psounis, and C. S. Raghavendra. Routing in a delay tolerant network. In *Proc. of WDTN*, 2005.

[13] T. Spyropoulos, K. Psounis, and C. Raghavendra. Efficient routing in in termittently connected mobile networks: The single-copy case. *ACM/IEEE Transactions on Networking*, 2007.

[14] E. P. Charles and P. Bhagwat. Highly dynamic destination sequenced distance vector routing (DSDV) for mobile computers. In *Proc. of SIGCOMM*, 1994.

[15] W. Zhao, M. Ammar, and E. Zegura. A message ferrying approach for data delivery in sparse mobile ad hoc networks. In *Proc. of Mobihoc*, 2004.

[16] Q. Li and D. Rus. Communication in disconnected ad hoc networks using message relay. *ACM/IEEE Transactions on Networking*, 2003.

[17] E. P. C. Jones, L. Li, and P. A. S. Ward. Practical routing in delay-tolerent networks. In *Proc. of SIGCOMM*, 2005.

[18] H. Shen and Z. Li. MDR: Market-based distributed routing scheme in hybrid wireless networks. Technical Report TR-2007-07, CSCE Department, University of Arkansas, 2007.

[19] A. Vahdat and D. Becker. Epidemic routing for partially connected ad hoc networks. Technical Report CS-200006, Duke University, 2000.

[20] X. Zhang, G. Neglia, J. Kurose, and D. Towsley. Performance modeling of epidemic routing. In *Proc. of IFIP*, 2006.

[21] J. Widmer and J. Y. L. Boudec. Network coding for efficient communication in extreme networks. In *Proc. of WDTN*, 2005.

[22] T. Small and Z. Haas. Resource and performence tradeoffs in delay-tolerant wireless networks. In *Proc. of WDTN*, 2005.

[23] Y. Wang, S. Jain, M. Martonosi, and K. Fall. Erasure-coding based routing for opportunistic networks. In *Proc. of SIGCOMM*, 2005.

[24] M. McNett and G. M. Voelker. Acm mobile computing and communication review. In *Access and mobility of wireless pda users*, 2007.

[25] B. N. Levine A. Balasubramanian and A. Venkataramari. DTN routing as a resource allocation problem. In *Proc. of SIGCOMM*, 2007.

[26] B. Pasztor, M. Musolesi, and C. Mascolo. Proc. of mass. In *Opportunistic mobile sensor data collection with Scar*, 2007.