SMART: A Selective Controlled-Flooding Routing for Delay Tolerant Networks

Lei Tang, Qunwei Zheng, Jun Liu, Xiaoyan Hong
Department of Computer Science
University of Alabama, Tuscaloosa, AL 35487
{ltang,qzheng,jliu,hxy}@cs.ua.edu

***Abstract*—Delay-Tolerant network (DTN) is a network in which no simultaneous end-to-end path exists. And the messages delivered in the DTN usually have large delivery latency due to network partition. These special characteristics make DTN routing a challenging problem. In this paper, we propose a DTN routing protocol *SMART*. SMART uses *travel companions* of the destinations (i.e. nodes that frequently meet the destination) to increase the delivery opportunities. In the first phase of SMART, a fixed number of copies of a message are injected into the network to forward the message to the companions of the destination. In the second phase of SMART, a companion of the destination only forwards the message to a fixed number of the destination's companions. Our analysis and simulation results show that SMART has a higher delivery ratio and smaller delivery latency than opportunistically controlled-flooding schemes and has a significantly smaller routing overhead than pure flooding schemes.**

***Index Terms*—routing protocol, delay-tolerant network**

I. INTRODUCTION

Delay Tolerant Networks (DTNs) are networks in which no simultaneous end-to-end path exists. Typically, message delivery experiences long delays due to the disconnected nature of the network. Many networking scenarios can be viewed as DTN. ZebraNet [1] is developed for monitoring the long-term behaviors of wild animals (e.g zebra) sparsely distributed over a large area. Another DTN networking scenario is to provide communication among the villages of Saami population of Reindeer Herders living in remote areas in Swedish Lapland [2]. In these disconnected networks, most of the time there is no complete end-to-end path between source and destination and message delivery relies on the mobility of nodes in the network.

Due to network partition, conventional MANET routing protocols designed for connected network such as DSR [3] and AODV [4] are not applicable. Routing protocols designed for DTNs, on the other hand, better accommodate such extreme environment. Messages in the DTN are routed in a store-and-forward manner, i.e. messages are buffered and forwarded hop-by-hop by intermediate nodes until reaching destinations.

Many DTN routing schemes have been proposed: [5], [6], [7], [8], [9], [10], [11], [12], [13], [14], [15], [16], [17]. Depending on their methodologies, the above routing schemes can be summarized into the following three types.

- *Predicting good forwarders:* The schemes in this type try to predict which nodes are useful for delivering the messages based on nodes' history-encounter information [5], [13], [12], nodes' context information [16] or nodes' location visiting patterns [11], [17].
- *Meeting the destinations by schedule:* A representative scheme in this type is *Message Ferry* (MF in short) [9]. In MF scheme, there are a special type of nodes called *ferries* which are able to change their trajectories proactively to help other nodes to deliver messages.
- *Opportunistically forwarding messages:* The schemes in this type such as Epidemic [7] and [8], [14], [15] do not try to predict which nodes are good forwarders nor do they change nodes' trajectories to facilitate message deliveries. Instead, nodes in the network opportunistically forward messages to other nodes until the messages reach their destinations.

The above three methodologies have their specific advantages and disadvantages. The opportunistically-forwarding-based schemes (e.g Epidemic routing) can achieve 100% delivery ratio and smallest delay at the cost of network bandwidth and buffer space [7]. The network bandwidth consumption of opportunistically-forwarding-based schemes can be improved by limiting the number of message copies forwarded [15]. But since the Spray scheme in [15] does not distinguish between nodes with regard to their probability of delivering the messages, each message still needs to be forwarded to a large number of nodes in order to reach the destination with high probability and small delay.

The prediction-based schemes (e.g SOLAR [17]) try to reduce message overhead and buffer contention through only forwarding messages to nodes with high delivery probability. However, due to the disconnected nature of the network, in existing prediction-based schemes, it may take a very long time before every node receives the delivery probability of other nodes. Furthermore, as pointed out in [14], when the distance between a source and a destination is large, it will take the source a long time until it finds a message forwarder with high delivering probability to the destination, which is called *slow start* problem.

The scheduled-meeting schemes can be efficient in terms of message overhead and buffer consumption. But it requires that *ferries* change their trajectories on-demand to help other nodes to deliver messages, which is not the targeted network scenario of this paper.

In this paper, we present a novel routing scheme called

SMART: Selectively MAking pRogress Towards delivery. Nodes in the network are not likely to move around randomly, but exhibit repeating mobility patterns [5]. SMART exploits nodes' mobility patterns and controlled-flooding methodology to achieve high delivery rate and small delivery latency while keeping the messaging overhead low.

In our system, every node keeps track of its *"travel companions"* (or *companions* in short), which are the nodes encountered frequently. Since travel companions of the destination are very likely to meet the destination, we strive to send the message to its destination's companions to enhance routing efficiency. The routing process of SMART can be viewed as two phases and a message could be delivered to its destination in either phase. In the first phase, the message is opportunistically forwarded to a set of companions of the destination. In the second phase, a companion that received the message in the first phase further forwards the message to other companions of the destination until the message is delivered to the destination. By using controlled-flooding, both phases of SMART routing have an upper bound on the number of message transmissions.

SMART combines the good features of opportunistically-forwarding-based schemes and prediction-based schemes. The first phase of SMART overcomes the *slow start* problem by opportunistically forwarding the message to a limited number of nodes. The second phase enhances routing efficiency by focusing on forwarding the message to the destination's companions rather than every node encountered.

Furthermore, SMART is distinguished from existing prediction-based schemes such as SOLAR [17] and PROPHET [5] in that it does not assume all nodes' location-visiting probabilities are known nor does it require nodes propagating their location-visiting and nodes-meeting probabilities to other nodes. This is important since propagating each node's location-visiting probability and other probabilities to all the other nodes is time-consuming and unreliable in a network that is disconnected most of the time.

Meanwhile, SMART avoids using location-visiting probabilities used in MOBISPACE [11] and SOLAR because propagating location-visiting probabilities may raise privacy concerns and may be inaccurate in predicting which nodes are good message forwarders. For instance, we can not tell whether two nodes are more likely to encounter if they both visit a set of locations frequently in the past: they may visit the same locations at different times and never meet each other. Instead, a node in SMART decides who are its companions using the encounter history with other nodes.

Through theoretical analysis, we find that SMART has higher delivery probability and smaller message delivery latency than opportunistically controlled-flooding-based Spray scheme [15]. And our simulation results demonstrate that while achieving a high delivery ratio and small delivery latency, SMART has very small routing overhead compared with flooding-based Epidemic routing scheme.

The rest of this article is structured as follows. In section II, we introduce state-of-art DTN routing schemes and classify them into three categories according to their methodologies. In section III, we present the design of SMART. Section IV analyzes the performance of SMART. In section V, we evaluate SMART through simulations. We summarize our work and outline future plan in section VI.

II. RELATED WORK

In section I, we have categorized the methodologies of existing DTN routing schemes into 3 types: opportunistically forwarding messages, predicting good forwarders, and meeting the destinations by schedule. In this section we overview the state-of-the-art DTN routing protocols based on their methodologies.

A. Opportunistically-forwarding Protocols

The opportunistic-forwarding schemes opportunistically forward the messages to the nodes encountered without predicting which nodes are good message forwarders.

Epidemic Routing [7] proposed by Vahdat and Becker is one of the earliest DTN routing protocols. When nodes encounter, they exchange messages unknown to each other. Eventually messages will be propagated to the destination. This flooding-based propagation consumes buffer space very quickly. *Spray* routing scheme [15] is similar to Epidemic Routing scheme but only injects a fixed number of copies of each message. In [18], *Spray and focus* scheme is proposed to improve the original Spray scheme by performing utility function based forwarding, in which a node computes a utility function to predict the usefulness of other nodes in delivering messages to the destinations. The utility function calculation is transitive. Unlike Spray and focus scheme, a node in SMART does not calculate the delivery probabilities of all nodes in the network nor does it transmit the delivery probabilities to other nodes or transitively calculate delivery probabilities.

Erasure-coding Based Routing (EBR) [8] divides a message into a set of code blocks, which are "sprayed" to a set of relays. Any sufficiently large subset of the generated code blocks can be used to reconstruct the original message. *Data MULE Routing* [6] proposed by Rahul *et al.* exploits the randomly-moving mobile nodes (MULEs) to deliver messages in a sparse sensor network, which receive messages from stationary sensors when in close range, buffer the messages received and drop them off to wired access points when in proximity.

B. Prediction-based Protocols

The motion pattern of nodes in the network can be exploited to predict which nodes are potentially useful to forward messages to destination. SOLAR [17] by J. Ghost *et al.* exploits nodes' mobility pattern to select which nodes are useful in routing. Based on the analysis of the wireless users' mobility traces collected from the ETH Zurich campus, SOLAR assumes that nodes regularly visit a small set of socially significant and geographically distant places called "hubs". Each node has its *"mobility profile"*, which comprises the hub-visiting probabilities. SOLAR assumes that each node knows every other node's mobility profile so that when source has a message to send, it only sends the message to the nodes that are highly possible to visit the set of hubs visited by the destination.

In SMART, a companion of the destination only forwards the message to other companions of the destination since the destination's companions are more likely to encounter the destination. But there are some differences between SMART and SOLAR. First, SMART does not require that every node knows all other nodes' mobility pattern. Knowing all other nodes' mobility pattern may not always be feasible since the number of nodes and the number of hubs in the network can be large, and DTN network is disconnected most of the time. Second, SMART does not have *slow start* problem because it opportunistically spreads the message in the first phase. Lastly, SMART does not use hub-visiting probabilities to predict which nodes are better message forwarders, because it is possible that two nodes both have high probabilities of visiting a hub but they never encounter because they visit the hub at different times.

Prediction-based routing protocols also include *PROPHET* [5], *MobiSpace* [11], *MV* [12], *Seek and Focus* [14], *Context-Aware Routing(CAR)* [16], and *MaxProp* [13]. All these schemes are similar in that all of them attempt to predict either which nodes are more likely to be useful in delivering the message to the destinations or which messages are more likely to be delivered. And the prediction is based on the past nodes' encountering history or the context information such as remaining battery lifetime.

C. Meeting-by-schedule Protocols

Some nodes in the network move according to accurate schedule, such as buses with fixed schedules. If a node knows when and where it will encounter other nodes and the accurate schedules of other nodes, it may be able to use the schedule information to determine which nodes are useful in message delivery and which messages have better delivery probability.

Message Ferrying routing (MF) [9] is a representative routing protocol that exploits scheduled contact. MF utilizes ferries (special mobile nodes) to pick up messages to be sent and to deliver the messages to the destinations. Furthermore, Tariq et al. [19] optimize ferry traversing route to encounter interested nodes with a certain minimum probability.

SMART is different from MF. First, SMART does not require a special ferry that has sufficient storage, communication and energy resources to help other nodes deliver messages since such a powerful ferry node may be impracticable or unnecessary in some network scenarios. In addition, SMART is potentially more time efficient since the nodes collaborate to deliver the messages instead of waiting to be served by a single ferry.

III. SMART ROUTING SCHEME

This section first overviews the rationale of SMART routing and then presents in detail the SMART scheme.

A. Overview of SMART

With the strengths and weaknesses of the existing DTN routing schemes in mind, we propose SMART with the following design goals:

- General and scalable. To be general enough to be used, it should work efficiently in a DTN network with no network infrastructure, no fixed storage devices.
- Require no knowledge of nodes' location, schedules, and motion pattern. Nodes do not transmit location-visiting information and nodes-meeting information to other nodes because in DTN transmitting this information may take a very long time and the quantity of information is huge when network scale is large.
- Exploit nodes-meeting information when forwarding the messages. But we do not assume a node knows other nodes' mobility pattern.
- Achieve high delivery rate and low delivery latency. Perform significantly fewer transmissions than flooding-based schemes such as Epidemic routing.

Since nodes' mobility exhibits patterns, the encounters among nodes also have patterns: some nodes are likely to meet while some are not. In SMART, a node broadcasts beacon messages periodically to declare its presence so that if two nodes frequently meets (i.e they are within each other's radio range), they get to know each other and become companions.

The rationale behind SMART can be illustrated using the following example. Assume the nodes in the network are people and Alice wants to send a message to Bob. With SMART protocol, Alice first sends the message to $f_1 - 1$ other people in hope of one of these people frequently encountering Bob (i.e a travel companion of Bob). After the message reaches a companion of Bob, say Charlie, Charlie sends the message to at most $f_2 - 1$ other companions of Bob hoping they will be able to encounter the destination. To send the message to the companions of Bob, Charlie does not need the prior knowledge of who are Bob's companions. Instead, before forwarding the message to a node x, Charlie asks x whether it is a Bob's companion.

Compared with location-visiting probabilities, companionship more accurately reflects the possibility of two nodes meeting each other since a node x will only regard another node y as its companion only if they meet frequently.

SMART differs from SOLAR, PROPHET and other prediction-based schemes in that a node in SMART does not need to know all the other nodes' mobility patterns (e.g location-visiting probabilities) nor does it transitively computes these probabilities by using other nodes' meeting probabilities. Exchanging the above probabilities may be impractical since most of the time the network is disconnected. Instead, a node in SMART decides who are its companions and does not need to transmit its companion information to other nodes.

SMART routing can be viewed as two phases though they may progress simultaneously. In the first phase, a fixed number of copies of the message are injected into the network to forward the message to the companions of the destination. In the second phase, a companion of the destination only forwards the message to other companions of the destination until the message is delivered to the destination. In both phases, we use opportunistically-forwarding method to forward the message with the control of the maximum number of message transmissions to reduce overhead. In the following subsection, we will introduce SMART routing scheme in detail.

B. Design of SMART

In SMART, when a node x moves in the network, it records the information of when it encounters other nodes. Based on the encounter history, x calculates the *companion value (CV)* between x and another node j using Eq. 1, in which n_j denotes how many times x meets j during the past T time units; $\alpha \in (0, 1)$ is aging factor and $tick_j$ is the number of time units since x last meets j; $t_{j1} \ldots t_{jn_j}$ are the times of x meeting j in increasing order.

$$CV(j) = \frac{n_j \times \alpha^{tick_j}}{\sum_{k=1}^{n_j} \frac{t_{j(k+1)} - t_{jk}}{n_j}} \quad (1)$$

$\sum_{k=1}^{n_j} \frac{t_{j(k+1)} - t_{jk}}{n_j}$ calculates the mean inter-contact time between x and j. CV between x and j is proportional to n_j and α^{tick_j} but is inversely proportional to mean inter-contact time. Among all the nodes encountered during the past T time, x selects γ largest-CV nodes as as its companions. The value of γ is determined by the number of frequently-encountered nodes each node has. If on average each node frequently encounters a large number of nodes, we can set γ to be a large value and vice versa. It is a difficult problem to give a deterministic solution for configuring the parameter γ, T and time unit. In this paper, we empirically configure these parameters and give details in section V.

It is important to forward the messages to the destination's companions quickly and efficiently (i.e using only a small number of message transmissions). To address this issue, we use the *Binary Spray* algorithm presented in [15]. *Binary Spray* algorithm controls the maximum number of message transmissions for each message. Each message M on a node x is associated with a counter, which specifies how many copies of M are stored on x. Here we use the notation $\{M, \lambda\}$ to denote that the number of copies of the message M on a node is λ. When a node x with $\{M, \lambda\}$ ($\lambda \geq 2$) meets a node y that does not have M, x forwards $\{M, \lfloor \lambda/2 \rfloor\}$ to y and keeps $\{M, \lceil \lambda/2 \rceil\}$ for itself. A node with $\{M, 1\}$ will stop forwarding M to other nodes. Instead it holds the message M until it encounters the destination or the message expires.

The message source of a message M starts with $\{M, \lambda\}$ so that M will be forwarded to at most $\lambda - 1$ other nodes. The configuration of λ is closely related to delivery ratio, delay and the number of message transmissions. We will study how to tune λ to achieve the best tradeoff between the performance and overhead in the later sections. Note we use the term "*spray*" interchangeably with the term "*binary spray*" in this article.

In SMART, a message may be transmitted using one of the three transmission modes: *normal spray mode*, *companion spray mode* and *direct transmission mode*. A message to be transmitted using *normal spray mode* will be sprayed to all the nodes that do not have it, whereas a message to be transmitted using *companion spray mode* is only sprayed to the companions of the destination. And a message to be sent using *direct transmission mode* will only be forwarded to the destination directly.

In the first phase of SMART, the message is sprayed to at most $f_1 - 1$ nodes using *normal spray mode*. If during the first phase the message reaches a companion x of the destination, then in the second phase the companion x uses *companion spray mode* to spray M to at most $f_2 - 1$ companions of the destination. If the message reaches none of the companions of the destination during first phase, the nodes that received the message during the first phase will directly send the message to the destination without further forwarding.

SMART routing algorithm is showed in Fig. 1. When source S needs to send a message M to a destination D, it checks if it is a companion of D. If so, M is binary sprayed to at most f_2 companions of D. If not, M is binary sprayed to f_1 nodes in the network. When a companion of D receives M, it sprays M only to at most $f_2 - 1$ destination's companions.

Fig. 2 illustrates the routing process of SMART. In the example, f_1 and f_2 is set to be 8 and 4, respectively. We use circle to denote a companion of D and use square to denote a non-companion node. The number in the figure denotes the number of message copies (e.g $A(4)$ means that node A has 4 copies of the message). In this example, the message reaches a companion of D, which sprays the message to other companions of D until the message reaches D.

SMART, as a hybrid scheme, combines the strengths of prediction-based schemes and opportunistically-forwarding schemes. It uses Spray algorithm to reach the companions quickly and exploits companions to enhance delivery efficiency. The original Spray algorithm controls how many copies of a message are injected into the network but it does not distinguish between nodes and the message is forwarded to every node with equal probability. The SMART's practice of a companion only spraying to other companions in the second phase improves routing efficiency by focusing on forwarding the message among the companions of the destination.

IV. ANALYSIS OF SMART

In this section, we analyze the performance of SMART regarding to the message delivery rate and delivery latency. We use the *carriers* of a message to denote the nodes that have the message. When a carrier is not a companion of the destination, we call it a *generic carrier*. Otherwise, we call it a *companion carrier*. The notations used in the paper are listed in Table I.

Algorithm 1 SMART Routing

```
Input: $M$, $S$, $D$, $f_1$, $f_2$
if $M$ not sent yet then
  if $S$ is a companion of $D$ then
    $S$ stores $\{M,\lceil f_2/2\rceil\}$ and sends $\{M,\lfloor f_2/2\rfloor\}$ to the first companion of $D$ or $D$;
    $S$ uses companion spray mode to send $M$;
  else
    $S$ stores $\{M,\lceil f_1/2\rceil\}$ and sends $\{M,\lfloor f_1/2\rfloor\}$ to the first node to meet;
    $S$ uses normal spray mode to send $M$;
  end if
end if
if a node $y$ with $\{M, n\}$ meets a node $x$ without $M$ then
  $y$ decides whether it forwards $M$ to $x$ based on the transmission mode of $M$ and the value of $n$;
end if
if node $x$ receives $\{M, n\}$ from $y$ for the first time then
  if $x = D$ then
    $M$ is delivered and stop;
  else if $x$ is not a companion of $D$ then
    if $n = 1$ then
      $x$ uses direct transmission mode to send $M$;
    else
      $x$ stores $\{M,\lceil n/2\rceil\}$ and sends $\{M,\lfloor n/2\rfloor\}$ to the first node to meet;
      $x$ uses normal spray mode to send $M$;
    end if
  else
    if $y$ is a companion of $D$ then
      if $n = 1$ then
        $x$ uses direct transmission mode to send $M$;
      else
        $x$ stores $\{M,\lceil n/2\rceil\}$ and sends $\{M,\lfloor n/2\rfloor\}$ to the first companion to meet;
        $x$ uses companion spray mode to send $M$;
      end if
    else
      $x$ stores $\{M,\lceil f_2/2\rceil\}$ and sends $\{M,\lfloor f_2/2\rfloor\}$ to the first encoutering companion of $D$ or to $D$;
      $x$ uses companion spray mode to send $M$;
    end if
  end if
end if
```

Fig. 1. SMART routing algorithm

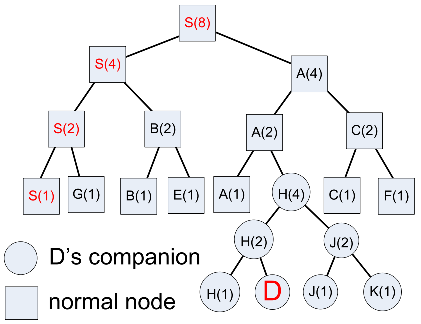

Fig. 2. SMART routing algorithm illustration with $f_1 = 8$ and $f_2 = 4$

TABLE I
NOTATIONS

Notation	Explanation
$CC(t)$	Number of companion carriers at time t.
$GC(t)$	Number of generic carriers at time t.
$NC(t)$	Number of nodes not having the message at time t.
N	Total number of nodes in the network.
D	Message destination.
M	A message.
γ	Average number of companions a node has.
ξ	Total number of locations in the network.
η	Mobility similarity of the companions.
α	Aging factor.
R_1	Message propagation rate among generic carriers.
R_2	Message propagation rate among companion carriers.
f_1	SMART phase 1 message transmission parameter.
f_2	SMART phase 2 message transmission parameter.
λ	Spray parameter controlling the max. number of transmissions for a message

A. *Delivery Rate Analysis*

Message delivery rate is the percentage of all messages that are delivered to destinations. When analyzing the delivery rate, we make the following assumptions:

- The nodes move among the locations in the network (e.g. shopping malls, classrooms, restaurants and etc) in discrete steps. Every unit time, a node moves from one location to another location.
- A node's location-visiting probabilities are independent and follow uniform distribution.
- The time spent on visiting a set of locations is proportional to the size of the set.

Nodes exchange messages when they meet at a location. For instance, assume node x have a message for node y. If x's location visiting sequence is $\{4, 3, 2, 1\}$ and y's location visiting sequence is $\{2, 3, 1, 4\}$. Then the messages will be delivered when x and y are both at location 3.

Since a message can be either delivered by a generic carrier or a companion carrier, the delivery rate equals (*1- probability of generic carriers not meeting the destination - probability of companion carriers not meeting the destination*). Equation 5 calculating the delivery rate is based on the above rationale. Now we see the steps of computing equation 5.

We use k to denote the number of message carriers (source, companion and generic carriers) and use t to denote time. First we analyze $P^g_{deliver}(k, \xi, t)$, the cumulative probability of k carriers delivering a message to the destination when the mobility of the message carriers is independent of the destination's mobility. At any time the destination and k carriers may have ξ^{k+1} possible location-visiting combinations. Based on the counting principle, $P^g_{deliver}(k, \xi, t)$ is computed in equation 2.

$$P^g_{deliver}(k, \xi, t) = 1 - \left(\frac{\xi(\xi-1)^k}{\xi^{k+1}}\right)^t = 1 - \left(\frac{\xi-1}{\xi}\right)^{kt} \quad (2)$$

Next we compute the delivery probability of a set of companion carriers meeting the destination. $P^c_{deliver}(\gamma, \xi, \eta, t)$ in equation 3 is the cumulative probability of γ companion carriers delivering the message to the destination. Two nodes become companions because they often meet. So we use η to model the mobility similarity of the companions. $\eta\xi$ represents the size of the small common location set, where the companions meet. In our assumptions, we assume the time spent on visiting a set of locations is proportional to the size of the set. So γ companion carriers spend ηt visiting $\eta\xi$ locations. It is worth noting that while the companions visit a set of locations at similar time, their visiting probabilities of the other locations are independent and follow the uniform distribution. Based on the above analysis, we have

$$\begin{aligned} P^{c}_{deliver}(\gamma,\xi,\eta,t) &= 1-\left[\frac{(\eta\xi)(\eta\xi-1)^{\gamma}}{(\eta\xi)^{\gamma+1}}\right]^{\eta t} \\ &= 1-\left(\frac{\eta\xi-1}{\eta\xi}\right)^{\gamma\eta t} \end{aligned} \tag{3}$$

A generic carrier may also encounter the destination at the $\eta\xi$ locations. So $P^{g'}_{deliver}(k,\xi,\eta,t)$ in equation 4 calculates the cumulative probability of k generic carriers delivering the message to the destination when they visit the $\eta\xi$ locations.

$$P^{g'}_{deliver}(k,\xi,\eta,t) = 1-\left[\frac{\eta\xi(\xi-1)^{k}}{\eta\xi\times\xi^{k}}\right]^{\eta t} = \frac{\xi^{t\eta k}-(\xi-1)^{t\eta k}}{\xi^{t\eta k}} \tag{4}$$

SMART uses controlled-flooding mechanism so eventually there are at most f_1 generic carriers and at most f_2 companion carriers. The destination and its companions are more likely to encounter at $\xi\eta$ locations, while at the other locations the companions have the same probability meeting the destination and the other nodes. Using equations 2, 3 and 4, equation 5 computes SMART's message delivering probability.

We can see from Fig. 3 that SMART delivers more messages than Spray when $f_1 + f_2 = \lambda$ and $\xi = 400$. Fig.4 shows the delivery rate of SMART increases when η increases from 0.05 to 0.2.

$$\begin{aligned} &P_{deliver}(f_1,f_2,\xi,\eta,t) = \\ &\quad = 1-\left[1-P^{g}_{deliver}(f_1+f_2,\xi(1-\eta),(1-\eta)t)\right] \\ &\quad \times\left[1-P^{g'}_{deliver}(f_1,\xi,\eta,t)\right]\left[1-P^{c}_{deliver}(f_2,\xi,\eta,t)\right] \\ &\quad = 1-\left(\frac{\xi(1-\eta)-1}{\xi(1-\eta)}\right)^{(f_1+f_2)(1-\eta)t} \\ &\quad \times\left(\frac{\xi-1}{\xi}\right)^{t\eta f_1}\left(\frac{\eta\xi-1}{\eta\xi}\right)^{f_2\eta t} \end{aligned} \tag{5}$$

B. Delivery Latency Analysis

Message delivery latency measures how long it takes for a message to be delivered to the destination. In this section, we first analyzes the delivery latency of Spray, followed by the delivery latency analysis of SMART.

Initially, only message source has the message. The first phase of SMART is similar to Spray protocol, in which the message is propagated to a number of nodes without differentiating between generic nodes and companions of the destination. In the second phase of SMART, companion carriers strive to send the messages to the destination or other companions of the destination.

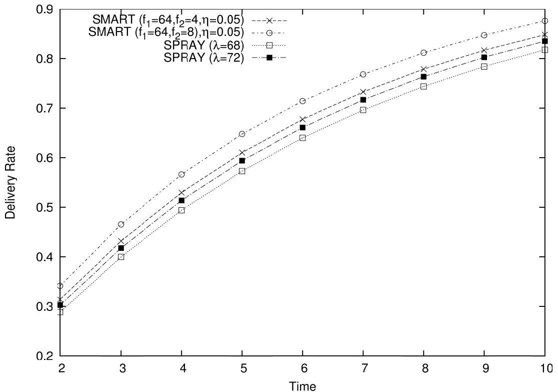

Fig. 3. Delivery Rate Comparison of SMART and Spray

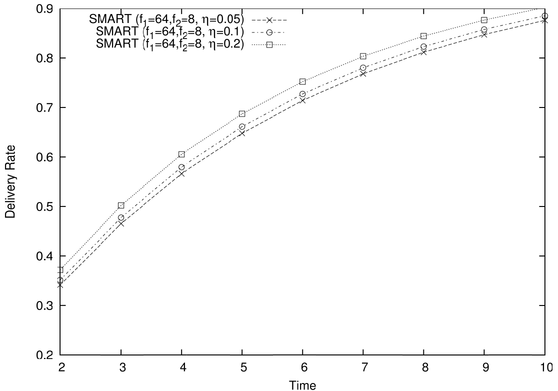

Fig. 4. SMART Delivery Rate with Varying η

To determine how fast a message is propagated from a generic carrier to another generic node and the propagation speed among the companions, we use continuous differential equations to model the message propagations. When we construct differential equations to model message propagation speed, we refer to the principles used in Kermack-McKendrick model [20], a model widely used for modeling epidemic spreading. Without losing generality, in the following analysis, we assume that source node is a generic node instead of being a companion of the destination.

1) Spray Delivery Latency Analysis: We assume a source node has uniform delivery probabilities to the destinations. Therefore, the delivery latency of Spray is determined by the speed of propagating the message to all the nodes in the network. In Spray, the message from the source reaches more and more nodes as time goes on. So $GC(t)$ increases while $NC(t)$ decreases. And we have

$$GC(t) + NC(t) = N. \tag{6}$$

In Kermack-McKendrick model, the change of infected hosts is proportional to the product of infection rate, number of infected hosts and number of non-infected hosts. Based on this model and equation 6, the change of the number of generic message carriers follows the equation

$$GC(t+\Delta t) - GC(t) = R_1 \times GC(t) \times [N - GC(t)]\Delta t, \tag{7}$$

where R_1 is the message propagation rate from a generic carrier to other generic carriers. Since only source has the message at the beginning, we have $GC(0) = 1$. Dividing both sides of equation 7 by Δt yields the equation 8:

$$\frac{dGC(t)}{dt} = R_1 \times GC(t) \times [N - GC(t)]. \qquad (8)$$

To solve equation 8, we need to determine R_1. R_1 is closely related to ξ and the smaller ξ the larger R_1. So we assume $R_1 = \frac{w}{\xi}$ and conduct simulations using Qualnet Network Simulator [21] to determine w. In the simulations, we set $N = 200$, $\lambda = N$ and measure the number of message carriers as time t goes. We run each simulation for 200 times and plot the propagation speed curve using the mean value of 200 simulations. Fig. 5 compares the simulation results and the results obtained by solving equation 8. We find that when $R_1 = \frac{0.8}{\xi}$, the simulation results matches the results obtained by solving 8. Hence, with $R_1 = \frac{0.8}{\xi}$, we could solve 8 and have the following equation:

$$GC(t) = \frac{N}{1 - e^{-0.8\frac{Nt}{\xi}} + Ne^{-0.8\frac{Nt}{\xi}}} \qquad (9)$$

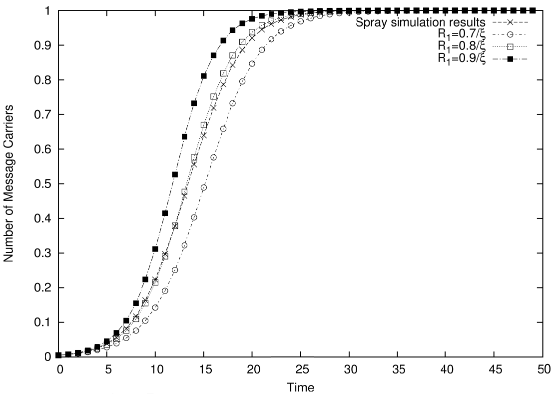

Fig. 5. R_1 vs. Propagation Speed to All Nodes

2) SMART Delivery Latency Analysis: The delivery latency of SMART depends on the speed of propagating the message to the companions of the destination and the speed of propagating the message from one companion to other companions. Therefore, to know the delivery latency of SMART, we need to find out the latency from the source node to a companion of the destination and the latency from the companion to the destination.

Section V-C1 studies the latency from the source to the companions of the destination by conducting simulations. From the results of Fig. 8, we find that under our network scenario it takes on average 5 unit time for a message to reach a companion. Here a unit time is 100S, which captures the average time interval between message exchanges in our network scenario. So on average it requires 5 message exchanges for a message to reach a companion of the destination.

Since the network we study is large-scale and the number of generic nodes is significantly larger than the average number of companions of each node (i.e γ), we can assume the message propagations among the generic nodes and the message propagations among the companions of the destination are independent. Therefore, based on the principles used in constructing equation 8, we derive equation 10 to model the message propagations among the companions.

$$\frac{dCC(t)}{dt} = R_2 \times CC(t) \times [\gamma - CC(t)] \qquad (10)$$

R_2, the propagation rate from a companion to other companions, relies on the mobility similarity (i.e $\eta\xi$) of the companions. The more frequently two companions meet, the more similar their mobilities are. When $N = 200$, $\eta = 0.05$, $\gamma = 16$, $f_1 = 64$, and $f_2 = 16$, we use simulations to find out the value of R_2. Fig. 6 compares the propagation speed among companions obtained from simulations and the propagation speed computed by equation 10. With R_2 being set to $\frac{0.57}{\eta\xi}$, the propagation speed computed by equation 10 matches the simulation results very well.

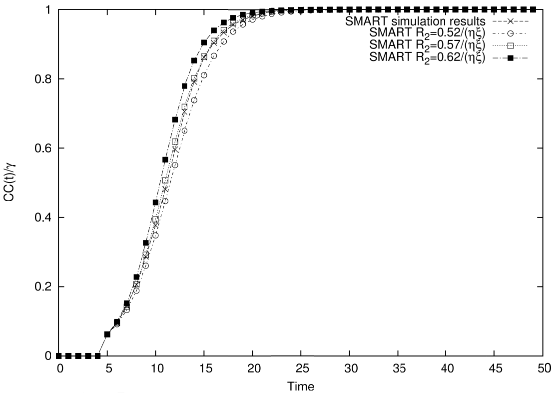

Fig. 6. R_2 vs. Propagation Speed among Companions

Because on average it takes 5 unit time for a message to reach a companion from the source, we have $CC(t) = 0$ when $0 \le t < 5$. Combined with the solution of equation 10, we have

$$CC(t) = \left\{ \begin{array}{l} 0,\ 0 \le t < 5 \\ \frac{\gamma}{1 - e^{-\frac{14\gamma t}{\xi}} + \gamma e^{-\frac{14\gamma t}{\xi}}},\ t \ge 5 \end{array} \right\} \qquad (11)$$

Based on formula 9 and 11, Fig. 7 compares the propagation speed of Spray and SMART when $N = 200$, $\lambda = 200$, $\gamma = 16$, $\eta = 0.05$, $f_1 = 64$ and $f_2 = 16$. The y-axis is the percentage of all nodes reached by Spray (i.e number of message carriers divided by N) and the percentage of companions reached by SMART (i.e $CC(t)/N$). From Fig. 7, we can see that the message reaches all the companions before it reaches all nodes, meaning SMART has smaller delivery latency than Spray.

V. EVALUATION

A. Protocol Comparisons

In this section, we evaluate SMART scheme and compare its performance with Spray and Epidemic routing by conducting simulations using Qualnet Network Simulator [21]. We focus on comparing the three routing protocols regarding to the following three metrics.

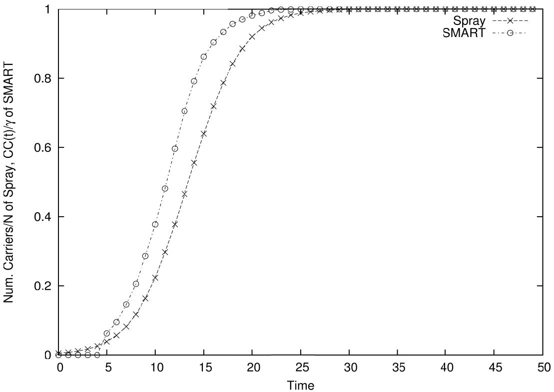

Fig. 7. Message Propagation Speed of SMART, Spray

- *message delivery rate:* This metric measures the percentage of all messages that are delivered to destinations.
- *message delivery latency:* This metric measures how long it takes for a message to be delivered.
- *message overhead:* This metric measures the number of message transmitted per delivered message.

In Epidemic routing protocol, a node forwards messages to every node it encounters. The routing process of Epidemic routing protocol can be summarized into 2 stages.

- *Beacon propagation:* In this stage, a node periodically broadcasts beacon messages to declare its presence. The beacon message uses a bit vector to specify which messages are now on this node.
- *Message forwarding:* when source has a message to send, it will first check whether the destination is within its radio range. If so, it sends the message to the destination directly. Otherwise, source will forward the message to all the nodes it meets, which will further disseminate the message until the message reaches the destination.

B. Simulation Setup

When evaluating the routing protocol performance, it is important to use a realistic mobility model. As pointed out in many works such as [16], nodes are unlikely to move completely randomly but follow some mobility pattern. Therefore, we have used a mobility model similar to the Agenda mobility model [22], in which nodes do not move randomly but exhibit realistic mobility patterns. There are various types of locations in the terrain, such as school, working place, gym, restaurant, home, shopping mall and so on. And nodes visit various locations based on typical mobility pattern of human beings.

The simulation configurations used by SMART, Spray and Epidemic routing are summarized in the table II. For each simulation, we run 5 times (each with a different random seed) and calculate the average value of the results.

C. Results

1) The Configuration of f_1: f_1 is one of the important parameters of SMART, which controls the maximum number of message copies injected into the network and affects the number of companion carriers reached at the end of the first phase of SMART. Here we use simulations to study how to configure f_1 so that a message will be able to reach a reasonable number of companion carriers at the end of the first phase of SMART. In addition, through simulations we study the latency for a message to reach a companion of the destination.

TABLE II
SIMULATION CONFIGURATION

Number of nodes	200
Terrain dimension	25600×12800
Simulation time	1800 S
f_1	128
f_2	4 or 16
γ	16
λ of Spray protocol	128
α	0.98
Mobility model	agenda-based mobility model
Message traffic model	random (source,destination) pairs
Message Size	512 bytes

Fig. 8 shows that on average it takes about 5 unit time for a message to reach a companion of the destination. The reason of using 100 seconds as an unit time is to capture the average time for a message exchange between two nodes since the message propagation relies on the message exchanges among nodes. Also we let T (a parameter used in CV computation) equal the simulation time. As the time goes on, the dramatic increase of the number of companions reached by Epidemic routing reflects that the message overhead of Epidemic increases exponentially. Since SMART uses controlled flooding, the number of companions reached stabilizes after all copies of the message are forwarded.

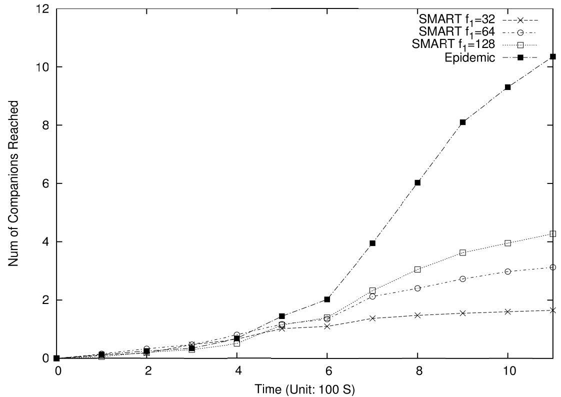

Fig. 8. Number of Companions Reached vs. f_1

2) Performance vs. Traffic Load: In this set of simulations, we vary traffic load (i.e. number of messages sent during the simulation) and measure the aforementioned metrics for the above three routing protocols.

Fig. 9 shows the delivery rate of the three routing schemes when changing the message traffic load. The simulation results demonstrate that SMART outperforms Spray protocol and delivers more than 90% messages. Epidemic routing delivers all the messages since it floods a message to all the nodes. The performance of Spray degrades as traffic load increases because each message has less time to be transmitted when message traffic load increases given that the meeting time between nodes remains the same. But for SMART, the meeting time between nodes has a less significant influence on delivery rate since companions frequently meet each other, thereby having more time to deliver messages to each other. In addition, since message source and destination are randomly selected, the message traffics are more evenly distributed to all the nodes when message traffic load increases, which contributes to the slight improvement of SMART's delivery rate.

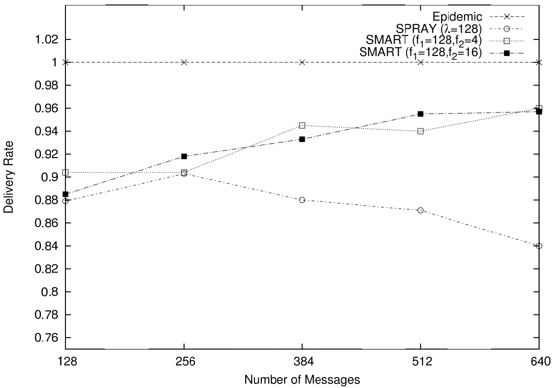

Fig. 9. Delivery Rate vs. Message Traffic

Figure 10 of the delivery latency demonstrates that SMART has much smaller delivery latency than Spray. The reason is that SMART exploits companions to deliver the messages while Spray does not. Because the companions of the destination are more likely to meet the destination, it takes a shorter time for them to deliver the message to the destination. Epidemic Routing delivers messages faster than SMART because a node forwards the messages to every node it meets. In this way, nodes quickly deliver the message to the destination.

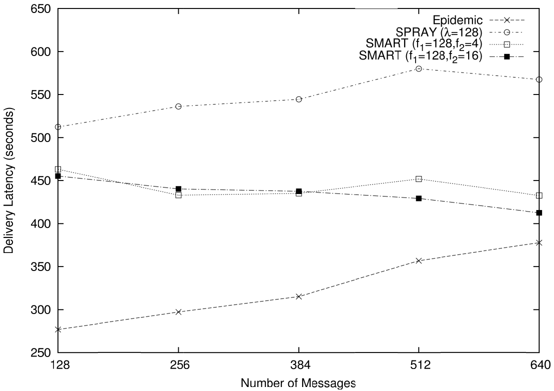

Fig. 10. Delivery Latency vs. Message Traffic

Fig. 11 shows that the message overhead of Epidemic Routing is much larger than the message overhead of SMART and Spray. The reason that Epidemic Routing incurs huge overhead is because it relies on flooding to deliver message. SMART and Spray use controlled flooding mechanism to forward messages so that the number of message transmissions has an upper bound, which is a small constant. It is worth noting that the overhead of SMART and SPRAY are almost identical when $f_1 = 4$. The reason is as follows. In SMART after a message reaches a companion, the companion will only forward the message to other companions, which reduces overhead.

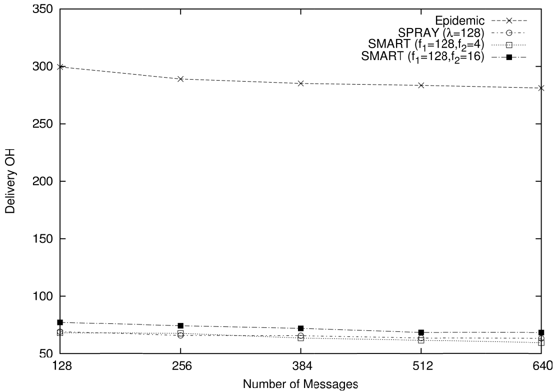

Fig. 11. Message Overhead vs. Message Traffic

3) Performance vs. Mobility: DTN network is a sparse network, which is disconnected most of the time. So message deliveries rely on nodes' mobility. In different DTN scenarios, nodes' mobility may vary. A node's mobility influences the number of locations it may visit during the given the simulation time. Hence nodes' mobility influences the meeting probability of the nodes. In addition, mobility affects two nodes' meeting time, which is the time period two nodes stay in each other's radio range.

In this set of simulations, we measure the influences of the mobility on the performance of the above three schemes. We alter a node's mobility by changing its average dwelling time at each location. If we want a node to become more mobile, we decrease its average dwelling time. In the following set of simulations, we vary each node's average dwelling time at each location from 1 to 5 (normalized against the largest average dwelling time) and fix the message traffic as 256 messages.

Fig. 12 demonstrates that when nodes become more mobile, all three routing schemes achieve higher delivery rate. From Fig. 12, we can see that the positive effects on the delivery rate brought by the increase of meeting probability outweighs the negative effects brought by the changes of nodes' meeting time. Moreover, when nodes become more mobile, the delivery rate of SMART and Spray increasingly approaches that of Epidemic routing. So when nodes have high mobility, we can use SMART to achieve a high delivery ratio with a small near constant overhead.

Fig. 14 shows that the delivery latencies of all three routing schemes decrease when nodes become more mobile. The performance enhancement is because nodes visit more locations in a given time so they have more opportunities to meet and meet each other more quickly. From Fig.13, we know that SMART and Spray keep a very low overhead when mobility changes.

VI. CONCLUSION

The paper has presented a scheme SMART that exploits controlled opportunistic flooding and each node's travel companions to achieve efficient message delivery. SMART pro-

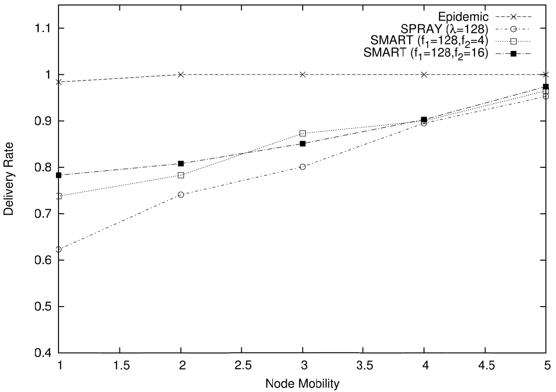

Fig. 12. Delivery Rate vs. Nodes' Mobility

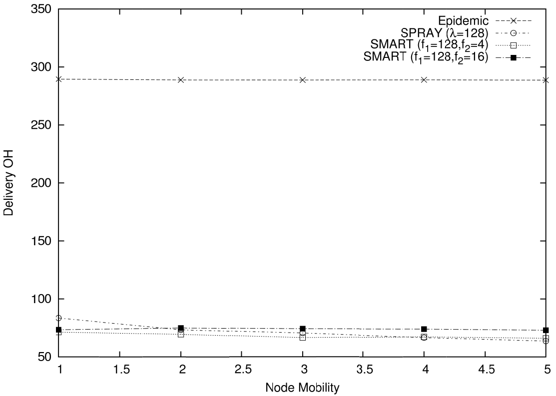

Fig. 13. Message Overhead vs. Nodes' Mobility

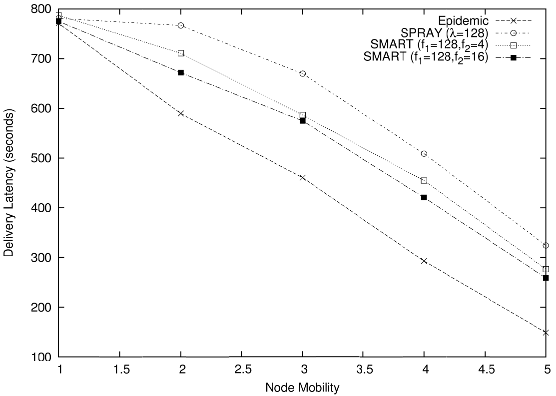

Fig. 14. Delivery Latency vs. Nodes' Mobility

tocol uses two phases to achieve high delivery ratio while keeping a low message overhead. In the first phase, a fixed number of messages are injected into the network to forward the message to a companion of the destination. The second phase starts by a companion of the destination, which only sprays the message to a fixed number of the destination's companions.

The paper has presented both analytic results and simulation results comparing SMART scheme with Spray scheme and Epidemic Routing scheme. The analytic results demonstrate that SMART has higher delivery rate and smaller message delivery latency than Spray. And the simulation results show that SMART has a significantly lower message overhead than Epidemic routing scheme while maintaining comparable delivery rate. From the simulation results, we find that the overhead of SMART is almost identical to the overhead of Spray but SMART has a higher delivery rate and smaller delivery latency.

In our future research of SMART, we plan to further develop SMART by exploring mobility control method for proactively meeting companions and extending our performance study to cover more DTN routing protocols.

REFERENCES

[1] P. Juang, H. Oki, and Y. W. et al., "Energy-Efficient Computing for Wildlife Tracking: Design Tradeoffs and Early Experiences with ZebraNet," in *Proceedings of ASPLOS-X*, 2002.

[2] A. Doria, M. Udn, and D. P. Pandey, "Providing Connectivity to the Saami Nomadic Community," in *Proc. 2nd Int. Conf. on Open Collaborative Design for Sustainable Innovation*, 2002.

[3] D. B. Johnaon and D. A. Maltz, "Dynamic Source Routing in Ad Hoc Wireless Networks," in *Mobile Computing*, 1996, pp. 153–181.

[4] C. Perkins and E. Royer, "Ad hoc On-demand Distance Vector Routing," in *2nd IEEE Workshop on Mobile Computing Systems and Applications*, 1999.

[5] A. Lindgren, A. Doria, and O. Schelen, "Probabilistic Routing in Intermittently Connected Networks," in *SIGMOBILE Mobile Computing Communications Review*, July 2003, pp. 7:19–20.

[6] R. Shah, S. Roy, S. Jain, and W. Brunette, "Data MULEs: Modeling a Three-tier Architecture for Sparse Sensor Networks," in *IEEE SNPA*, 2003.

[7] A. Vahdat and D. Becker, "Epidemic Routing for Partially-connected Ad hoc Networks," in *Technical report, Duke University*, 2000.

[8] Y. Wang, S. Jain, M. Martonosi, and K. Fall, "Erasure Coding based Routing in Opportunistic Networks," in *ACM SIGCOMM Workshop on Delay Tolerant Networking*, 2005.

[9] W. Zhao, M. Ammar, and E. Zegura, "A Message Ferrying Approach for Data Delivery in Sparse Mobile Ad Hoc Networks," in *MobiHoc*, 2004.

[10] S. Jain, K. Fall, and R. Patra, "Routing in a Delay Tolerant Network," in *ACM Sigcomm*, 2004.

[11] J. Leguay, T. Friedman, and V. Conan, "DTN Routing in a Mobility Pattern Space," in *ACM SIGCOMM - Workshop on delay tolerant networking and related topics (WDTN-05)*, 2005.

[12] B. Burns, O. Brock, and B. N. Levine, "MV Routing and Capacity Building in Disruption Tolerant Networks," in *IEEE Infocom*, 2005.

[13] J. Burgess, B. Gallagher, D. Jensen, and B. Levine, "Maxprop: Routing for vehicle-based disruption-tolerant networking," in *IEEE Infocom*, 2006.

[14] T. Spyropoulos, K. Psounis, and C. S. Raghavendra, "Single-copy routing in intermittently connected mobile networks," in *Proc. of IEEE Secon'04*, 2004.

[15] T. Spyropoulos, K. Psounis, and C. Raghavendra, "Spray and wait: an efficient routing scheme for intermittently connected mobile networks," in *WDTN '05*, 2005, pp. 252–259.

[16] M. Musolesi, S. Hailes, and C. Mascolo, "Adaptive Routing For Intermittently Connected Mobile Ad hoc Networks," in *WOWMOM'05*, 2005.

[17] J. Ghosh, H. Q. Ngo, and C. Qiao, "Mobility profile based routing within intermittently connected mobile ad hoc networks (icman)," in *IWCMC '06*, 2006, pp. 551–556.

[18] T. Spyropoulos, K. Psounis, and C. S. Raghavendra, "Spray and focus: Efficient mobility-assisted routing for heterogeneous and correlated mobility," in *PERCOMW '07*, Washington, DC, USA, 2007, pp. 79–85.

[19] M. M. B. Tariq, M. Ammar, and E. Zegura, "Message ferry route design for sparse ad hoc networks with mobile nodes," in *MobiHoc '06*, 2006, pp. 37–48.

[20] W. Kermack and A. McKendrick, "A contribution to the mathematical theory of epidemics," in *Proceedings of the Royal Society of London Series A*, 1927, pp. 700–721.

[21] Scalable Network Technologies (SNT), "Qualnet Network Simulator," http://www.qualnet.com/.

[22] Q. Zheng, X. Hong, and J. Liu, "An agenda based mobility model," in *39th Annual Simulation Symposium*, 2006.