# Sociological Orbits for Efficient Routing in Intermittently Connected Mobile Ad Hoc Networks

Joy Ghosh, Seokhoon Yoon, Hung Ngo, Chunming Qiao
Department of Computer Science and Engineering
University at Buffalo, The State University of New York
201 Bell Hall, Buffalo, NY 14260-2000
Email:{joyghosh, syoon4, hungngo, qiao}@cse.buffalo.edu

***Abstract*— Routing in Intermittently Connected Networks (ICN) is a challenging problem due to the uncertainty and time varying nature of network connectivity. In this work, we focus on a special class of ICN formed by mobile ad hoc users called ICMAN. We first consider a new and practical probabilistic mobility model where the nodes move between a set of "hubs" in a partially repetitive and nondeterministic pattern to form the so-called "sociological orbits". Second, to leverage the sociological orbit based mobility pattern in routing within ICMAN, we propose a series of multi-path Sociological Orbit aware Location Approximation and Routing (SOLAR) protocols. We present theoretical analysis of the mobility model and routing algorithms under consideration, and show that the proposed routing algorithms can outperform other conventional routing approaches in an ICN by taking advantage of the sociological orbit based mobility pattern.**

***Index Terms*— Mobility framework, Routing protocols, Intermittently Connected Networks, Theoretical model, Performance analysis**

## I. INTRODUCTION

An Intermittently Connected Network (ICN) may be modeled as a graph, where the capacities and durations of edges between nodes are time varying due to the mobility of users. The most important characteristic of all ICN, (also referred to as Delay Tolerant Networks [8], Disruption Tolerant Networks, etc.) that sets them apart from conventional mobile ad hoc networks (MANET) is the possibility that an end-to-end path via intermediary peers may not exist from a source to a destination at any one point in time. This renders the traditional MANET routing protocols useless for ICN. In this work, we focus on Intermittently Connected Mobile Ad hoc Networks (ICMAN), whose features include those of ICN as well as those of MANET such as lack of infrastructure, and non-deterministic mobility pattern of the nodes. We aim to study practical mobility and accordingly connectivity patterns of such ICMAN, and propose efficient routing protocols by taking full advantage of the unique mobility patterns of the nodes within ICMAN.

Earlier work [27] on temporally disconnected networks proposed intelligent means of data dissemination, which led to various other propositions based on similar concepts. More recent research in this area of ICN [17] has considered some algorithms based on deterministic mobility of the nodes, which is neither applicable to MANET, nor to ICMAN.

For a conventional MANET (as opposed to intermittently connected networks), we have proposed a novel framework called Sociological Orbit aware Location Approximation and Routing (SOLAR), which takes advantage of the "macro-mobility" information obtained from the sociological movement pattern of mobile users. This mobility information, also referred to as the "mobility profile", was extracted from our observation that the movement of a mobile user exhibits a partially repetitive "orbital" pattern involving a set of "hubs". We have already shown that SOLAR is not only general enough to be realistic, but is also specific enough to be useful [11], [12].

One of the main contributions of this paper is a new orbit based probabilistic mobility model that is particularly suitable for ICMAN. Another main contribution is the extension of the concept of SOLAR protocol for routing in ICMAN where a message from a source hub to a destination hub may be delivered only if the source or some intermediate nodes carrying the message move into the the destination hub (as opposed to the assumption that there are sufficient number of intermediate nodes in between the two hubs and hence geographical forwarding can be used [11], [12]).

In particular, we consider ICMAN where each node may have a list of hubs to visit such that the node may visit a hub on that list with a certain probability and once in a hub, the node may stay there for a while before moving to another hub in the list with a certain probability. In addition, we propose several different variations of multi-path SOLAR protocols for ICMAN, and analyze them both theoretically and via simulations. Amongst these variations, our proposed *Dynamic SOLAR-KSP* protocol (as to be described in detail later) caters to the semi-deterministic or pseudo-random orbit based mobility of nodes by incorporating offline calculation of end-to-end delivery probability with dynamic selection of next hop. It is to be noted that such work differs from all prior work done on deterministic mobility, from routing algorithms based on either completely offline path calculation or completely dynamic selection of the next hop, and from purely probabilistic routing. The readers are referred to Section VII for a detailed comparison of our work with other related work in this area.

To the best of our knowledge, this is the first work to exploit the sociological "orbit" and "hub" based routing concepts within ICMAN. We compare the performances of the different

SOLAR protocols along with the simple and efficient approach of Epidemic Routing [27] and show that all SOLAR protocols outperform Epidemic Routing in terms of higher data throughput, lower network overhead, and lesser end-to-end data delay.

The rest of the paper is outlined as follows. In Section II, we motivate our work by discussing the sociological movement pattern of mobile ICMAN users, and present an example for our proposed Probabilistic Orbit model. In Section III, we theoretically analyze the main SOLAR framework, and in Section IV, we propose several Sociological Orbit aware Location Approximation and Routing (SOLAR) protocols for ICMAN. In Section V, we evaluate the performance of SOLAR through simulations, and showcase its simplicity and superiority in terms of higher data throughput, lower network overhead, and lesser end-to-end data delay. In Section VI, we further analyze the effect of cache size and cache timeout on our SOLAR protocols. In Section VII we uphold SOLAR against other related work and finally conclude this work in Section VIII.

## II. Sociological Movement Pattern

In the real world, users routinely spend a considerable amount of time at a few specific place(s) that we refer to as hub(s). For example, a graduate student in school may visit and spend some significant amount of time in his/her laboratory, a seminar room, or the cafeteria. Although it is hard (or may be even against privacy policies) to keep track of an individual at all times, one can still take advantage of the fact that most users' movements are within and in between a list of hubs. In these situations, it is often possible to estimate/measure hub-visit probabilities and inter-hub movement patterns of an individual. This information then constitutes a part of the users' *mobility profiles*. For example, even if we do not know the exact location of the graduate student at any given time, given his/her mobility profile we can most probably find him/her in either the laboratory, or the seminar room, or the cafeteria, without having to look all over the building/campus. The more "periodic" the movement pattern is, the more we can take advantage of the mobility profile. This orbital concept is illustrated in Figure 1.

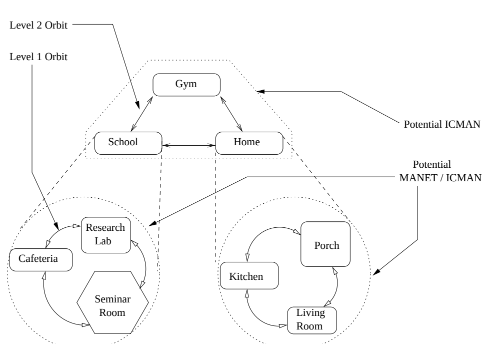



Fig. 1. A hierarchical view of sociological orbits

In practice, hubs can be identified in a variety of ways. The use of GPS service is the obvious first choice. Alternatively, signal strengths of wireless Ethernet packets can be used for location sensing and real-time tracking [20]. In the broader contexts of *pervasive/ubiquitous computing* [1], and *Ambient Intelligence* (AmI) [24], localization in a cosmopolitan area will be even more readily available. As part of our ongoing work we are in the process of analyzing real time experimental data on user mobility and network usage patterns to validate the existence of hubs and mobility profiles.

### A. An Example Probabilistic Orbit Model

To illustrate the concept of the sociological orbital movement, we first construct a simple yet practical orbital model called the Probabilistic Orbit, which is different from our earlier proposed Random Orbit model [12] in ways that are explained in the following description. The Probabilistic Orbit model allows for the creation of a certain number of hubs within the simulation terrain for all the nodes, as specified by the parameter *Number of Hubs*. These hubs are located at random places within the terrain. However, unlike in Random Orbit, in Probabilistic Orbit hubs are not allowed to overlap with each other (in accordance with the disconnected nature of an ICMAN). Each node can visit a subset of randomly chosen hubs, forming its hub list, following a specified transition probability matrix creating a *Probabilistic Orbit*. The list of hubs a node visits is bounded by *Hub List Size*, and the time it spends in each hub is an exponential random variable with mean specified by *Hub Stay Time* (unlike the uniformly distributed hub stay time in Random Orbit). Together, these two parameters define an Inter-Hub Orbit (IHO). Probabilistic Orbit further differs from Random Orbit in that Probabilistic Orbit assumes the hub list assigned to each node to stay constant for the entire duration of the simulation.

The mobility pattern of individual nodes shall comprise of two parts: movement inside a hub, and movement in between hubs. For convenience, the movement inside each hub, which shall also be referred to as the Intra-Hub Movement (IHM), was chosen to follow a modified Random Waypoint mobility model, whose speed range is denoted by *Intra-Hub Speed* (with a non-zero minimum as suggested by [30]), and whose pause time is denoted by *Intra-Hub Pause*. For movement in between hubs, we define a Point-to-Point Linear (P2P Linear) model. In this model, when a node wants to leave one hub for another, it first randomly selects a point within the destination hub. Second, it chooses a transition time that is exponentially distributed with mean *Inter-Hub Transition Time*. Third, it moves towards the destination point linearly from its current position with the velocity obtained by dividing the total travel distance by the total travel time. However, unlike in Random Orbit, nodes in Probabilistic Orbit are assumed *in most cases* to not communicate with any other node while it is traveling from one hub to another. ***Note that for each of the two parts, any known practical mobility models satisfying similar properties as above, may be chosen.***

Figure 2 illustrates the Probabilistic Orbit model. ***Note that, this example Probabilistic Orbit model does not simply integrate two common mobility models (Random Waypoint, and P2P Linear), but most importantly also introduces the practical orbital movement amongst hubs.*** Such a model is suitable for modeling wireless devices carried by users working in an office building, attending a convention, or around a campus, which may constitute the ICMAN. As users move around, devices either automatically, or with the user's permission/assistance may record the hubs visited, along with the frequency of visits to each of those hubs, and share the hub-based orbital mobility profile with trusted "acquaintances". Such mobility profile can then help improve routing as described in Section IV.

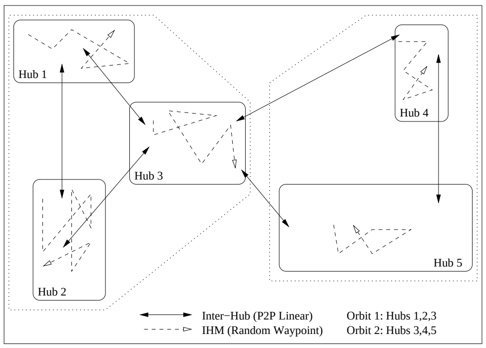



Fig. 2. The Probabilistic Orbit Model

where the state $(h, h')$ represents $X$ being on the move from hub $h$ to hub $h'$. The transitional probability $p_{ij}^X$ of the corresponding jump chain can then be computed as

$$p_{ij}^X = \begin{cases} \beta_{hh'}^X & i = h \text{ and } j = (h, h') \\ 1 & i = (h, h') \text{ and } j = h' \\ 0 & \text{otherwise,} \end{cases}$$

where $i, j \in I_X$ and $h, h' \in \mathcal{S}, h \neq h'$. The states in this CTMC when node X moves from hub $h$ to hub $h'$ are illustrated in Figure 3.

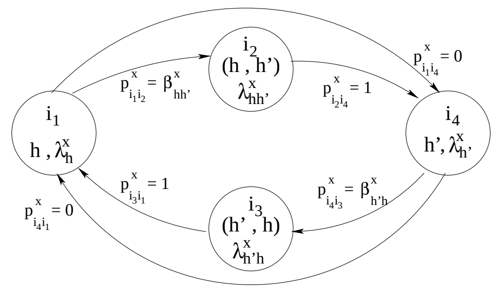


Fig. 3. States of Markov Chain for movements between hubs $h$ and $h'$

## III. Theoretical Analysis of SOLAR Framework

Before we describe the proposed SOLAR protocols, we shall motivate our work on a node's hub list and its associated transition probability matrix by studying its advantage in routing through an analytical model. To facilitate our discussion, we shall define a couple of terms:

1) *contact probability*: The probability of two nodes ever coming within each other's radio range (in contact) during the entire simulation.
2) *delivery probability*: The probability of a source delivering a packet to a destination via all possible paths of intermediary nodes that come "in contact" with their predecessors and successors in their respective paths.

We now discuss our model for computing this *contact probability* and present the complexity analysis of computing the *delivery probability*.

### A. Analytical Model for Contact Probabilities

Consider a node $X$ whose set of hubs is $\mathcal{S}$. Assume $X$'s staying time at a hub $h \in \mathcal{S}$ is exponentially distributed with parameter $\lambda_h^X$. After staying at $h$, $X$ moves to another hub $h' \in \mathcal{S}$ with probability $\beta_{hh'}^X > 0$. Obviously,

$$\sum_{h' \neq h} \beta_{hh'}^X = 1, \quad \forall h \in \mathcal{S}.$$

In our model, the time it takes for $X$ to move from $h$ to $h'$ is also exponentially distributed with parameter $\lambda_{hh'}^X$. This movement pattern can be modeled with a continuous time Markov chain (CTMC) $\{X_t\}_{t \geq 0}$ where - abusing notation - $X_t$ is the hub that $X$ is in at time $t$. The state space of this process is

$$I_X = \mathcal{S} \cup \{(h, h') \mid h, h' \in \mathcal{S}, h \neq h'\},$$

Suppose we have all these CTMC modeling the movements of nodes within their respective hub lists. Consider two nodes $X$ and $Y$ whose hub sets are $\mathcal{S}$ and $\mathcal{T}$ respectively. Assume $\mathcal{R} = \mathcal{S} \cap \mathcal{T} \neq \emptyset$. We would like to calculate the probability that $X$ meets $Y$ at some particular time $t$ in the future with $t$ sufficiently large (i.e., at equilibrium), and also the probability that $X$ meets $Y$ at a particular hub $h \in \mathcal{R}$ at time $t$.

Let $I_X$ and $I_Y$ be the state spaces of the CTMCs capturing the movements of $X$ and $Y$, respectively. In order to track the relative positions of $X$ and $Y$ together, define a continuous-time, discrete space stochastic process $\{Z_t\}_{t \geq 0}$ defined as follows.

$$Z_t = (X_t, Y_t), \quad t \geq 0.$$

Hence, the state space of $\{Z_t\}_{t \geq 0}$ is $I = I_X \times I_Y$. Note that, this stochastic process is like a Cartesian product of the other two chains. It is not difficult to show that $\{Z_t\}_{t \geq 0}$ is also a CTMC, as outlined below. To characterize this process, we need to compute the holding time distribution of any state $(i, i') \in I_X \times I_Y$, and the jumping probability from $(i_1, i_1')$ to $(i_2, i_2')$.

Obviously, the holding time of $(i, i')$ is the minimum holding times of $X$ at $i$ and $Y$ at $i'$. Hence, the holding time at $(i, i')$ for $Z_t$ is exponentially distributed with parameter

$$\mu_{(i,i')}^{XY} = \lambda_i^X + \lambda_{i'}^Y.$$

Moreover, with probability $\lambda_i^X / \mu_{(i,i')}^{XY}$ the holding time of $X$ at $i$ is smaller than the holding time of $Y$ at $i'$. Consequently, the jumping probability from $(i, i')$ to $(j, i')$ is

$$p_{(i,i'),(j,i')}^{XY} = \frac{\lambda_i^X}{\mu_{(i,i')}^{XY}} \cdot p_{ij}^X,$$

and the jumping probability from $(i, i')$ to $(i, j')$ is

$$p^{XY}_{(i,i'),(i,j')} = \frac{\lambda^Y_{i'}}{\mu^{XY}_{(i,i')}} p^Y_{i'j'}.$$

Now that we have the holding times distributions and the jumping probabilities, we can compute the generator matrix $Q^{XY} = \left(q^{XY}_{(i,i'),(j,j')}\right)$ for this chain:

$$Q^{XY}_{(i,i'),(j,j')} = \begin{cases} \mu^{XY}_{(i,i')} p^{XY}_{(i,i'),(j,i')} & i \neq j \ \& \ i' = j' \\ \mu^{XY}_{(i,i')} p^{XY}_{(i,i'),(i,j')} & i' \neq j' \ \& \ i = j \\ 0 & i \neq i' \ \& \ j \neq j' \\ -\sum_{k \in I_X, k' \in I_Y} Q^{XY}_{(i,i'),(k,k')} & (i,i') = (j,j') \end{cases}$$

The generator matrix in turn helps us solve for the steady state probabilities. It is easy to see that the product chain is finite, irreducible and positive recurrent. Hence, the chain converges to equilibrium. Let $\pi^{XY}$ denote the steady state probability distribution. Then, we can solve for $\pi^{XY}$ by solving the system

$$\pi^{XY} Q^{XY} = 0, \quad \sum_{i \in I_X, i' \in I_Y} \pi^{XY}_{(i,i')} = 1.$$

We are interested in the probability that the product chain is in states $(h, h)$ where $h \in \mathcal{R}$, i.e. the values of $\pi^{XY}_{hh}$ for all $h \in \mathcal{R}$. This is precisely the probability that $X$ meets $Y$ in $h$ at equilibrium. Finally, the probability that $X$ meets $Y$ at equilibrium is the sum of $\pi^{XY}_{hh}$ over all $h \in \mathcal{R}$.

Suppose $X$ holds a packet it would like to transmit to a downstream neighbor towards a destination. It cannot hold the packet forever due to limited buffer size (and possibly delay requirements). Some routing strategy may require $X$ to try its best to deliver the packet to (some of) the best neighbor(s) within a pre-defined time interval $T$. Consequently, given a time interval $T$ and given that $X$ is in some hub $h \in \mathcal{R}$, we are also interested in the probability that $Y$ will be in $h$ within $T$. Computing this probability is the same as computing the densities of the hitting times of the CTMC corresponding to $Y$ (probability that $Y$ hits $h$ given some initial distribution). There is no known general formulae. Computationally however, there are methods to compute these densities using Laplace transforms [16] for larger chains or uniformization [5] for smaller chains.

### B. Complexity of Computing the Delivery Probability

The common objective of various routing algorithms we propose in this paper is to maximize the delivery probability from a source $s$ to a destination $d$ subject to various constraints. This objective is the most reasonable due to the uncertainty and time varying nature of our ICMAN network connectivity. The constraints restricting our routing choices include buffer sizes, network overheads, end-to-end delay, data throughput, etc.

Let $G$ be a complete directed graph whose nodes represent the mobile nodes in the ICN under consideration. Fix a source $s$ and a destination $d$. Let $A$ denote a routing algorithm which aims to maximize the delivery probability from $s$ to $d$ subject to some constraints. The routing algorithm $A$ forces packets to be delivered through a subset of edges of $G$.

Let $G(A)$ denote the subgraph of $G$ induced by $A$, i.e. $(u, v)$ is an edge of $G(A)$ if there is a possibility that $u$ delivers a packet to $v$ under $A$. For instance, if $A$ is a naive broadcast strategy where each node delivers a packet it receives to all nodes it meets within a time interval $T$, then $(u, v)$ is an edge of $G(A)$ if the probability that $u$ meets $v$ within $T$ is positive. We will refer to $G(A)$ as the *delivery subgraph* associated with $A$.

More often than not, to each edge of $G(A) = (V, E)$ there associates an existence probability (the probability that the edge exists). For example, a sensible routing strategy is as follows. Given a fixed positive integer $k$, each node $u$ in the network chooses at most $k$ "down-stream" neighbors $v_1, \dots, v_l$ ($l \leq k$), i.e. $(u, v_i) \in E$, $\forall i = 1, \dots, l$. The routing strategy is to start transmission from $s$, and every node receiving a new packet forwards the packet to all available neighbors among the $k$ chosen ones.

Here, "availability" may mean availability within some specific time interval $T$, or availability without time limit. Whatever notion of "availability" we choose, there is a probability $p_e$ for each edge $e$ to exist. For instance, if the routing strategy specifies $T$, then the probabilities $p_e$ can be obtained by computing the hitting times densities as discussed earlier.

The overall objective function is the delivery probability, which is a function of $G(A)$ along with the existence probabilities $p_e$ of the edges of $G(A)$. For the $k$-downstream neighbor example, the delivery probability is then the probability that $s$ is connected to $d$ in $G(A)$. Let $p(G, A)$ denote this probability. The objective is to find an $A$ that maximizes $p(G, A)$.

Unfortunately, computing the connectedness probability in a random graph is very hard (even for graphs with bounded degree like in our case). There is a vast literature on this problem. Chapter 7 of [6] contains a partial set of references. Basically, this probability has very sharp threshold, and thus it is unlikely that it is a simple function [19][1]. In essence, the optimization problem may not even be in **NPO**.

Given this negative result, one can envision two general approaches:

- Find another function $p'(G, A)$ which approximates $p(G, A)$, yet $p'(G, A)$ is computable in polynomial time; then, find $A$ that maximizes $p'(G, A)$. This approach shall be a major future research topic for us. For the present, we do not know of any good strategy to estimate $p(G, A)$ which is polynomial-time computable for general $A$.
- Find a routing strategy $A$ for which $p(G, A)$ can reasonably be computed or estimated. This is the approach we take for the rest of this paper. We will propose several heuristics to maximize the delivery probability and compare them with related existing routing algorithms. These heuristics will be presented in the next section.

[1] We thank Prof. Van H. Vu for communicating this result to us.

### C. Approximation Algorithm for Delivery Probability

In the light of the discussion in the previous section, we propose an approximation algorithm for computing the delivery probability from source $s$ to destination $d$ in a network that is modeled as mentioned before: a directed graph $G = (V, E)$, where edge $e$ exists between two nodes $u$ and $v$ with probability $p_e(u, v)$ = *contact probability* of $u$ and $v$, as shown in Figure 4(a). First, we construct another graph $G_k = (V, E_k)$ from the graph $G$ by having each node (starting from $s$ onwards) choose at most $k$ edges to downstream neighbors, and deleting all other edges not chosen, as shown in Figure 4(b). Second, we modify the weight of each edge in $G_k$ to be $w_e = -1 * \log(p_e(u, v))$ for all nodes $u$ and $v$, and call this new graph as $G'_k$. Third, we construct a shortest path tree $G_{sp} = (V, E_{sp})$ from $G'_k$ as shown in Figure 4(c), and assign a level number to each node in a breadth first manner. Fourth, we replace the weight of each edge $w_e$ in $G_{sp}$ with $p_e(u, v)$, as in the original graph $G$. Finally, we add *special edges* (dotted edges in Figure 4(d)) between any node $v$ and destination $d$ in graph $G_{sp}$ that were connected by an edge $e \in E$ in the original graph $G$, to get our *delivery subgraph* $G' = (V, E')$.

Let $P^d(u, v)$ denote the delivery probability of node $u$ to node $v$. We apply our Algorithm 1 to this graph $G'$ starting with any node $u \neq d$ with maximum assigned level number, to obtain the delivery probability $P^d(s, d)$ of the source $s$ to the destination $d$. For each chosen node $u$, we consider all outgoing edges from $u$ to nodes $v_1, v_2, ...v_k$ say, and get a list of probabilities $p_1, p_2, ...p_k$, where $p_i = w_e(u, v_i) * P^d(v_i, d)$. Then, we can compute the delivery probability from $u$ to $d$ as

$$P^d(u, d) = 1 - \Pi_1^k(1 - p_i)$$

This process is repeated with decreasing level numbers till node $s$ is reached, and the required probability $P^d(s, d)$ is computed.

The optimal approach for computing the delivery probability from a source $s$ to a destination $d$ would include the following steps:

1) Calculate all possible paths from $s$ to $d$
2) Apply Algorithm 2 to compute the delivery probability by rules of inclusion and exclusion

We are currently in the process of simulating the two approaches: approximation algorithm, and the optimal algorithm, to evaluate the approximation ratio of our suggested approximation algorithm.

**Algorithm 1** : Approximation of Delivery Probability

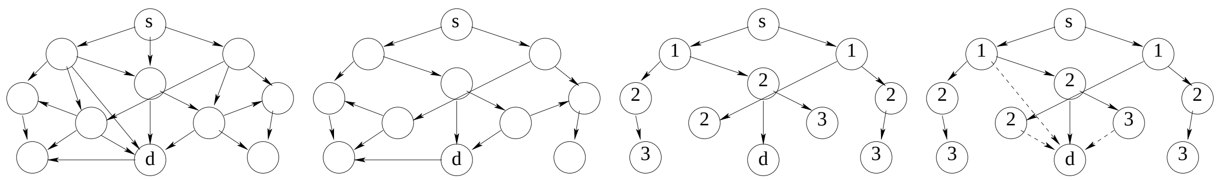
Input $\leftarrow G = (V, E), s, d$
$P^d(d, d) \leftarrow 1$
$L \leftarrow$ maximum assigned level number
**while** $L \geq 1$ **do**
  **for all** $u \in V, u \neq d$ with assigned level number $L$ **do**
    $i \leftarrow 1$
    **for all** outgoing edge $e \in E$ from $u$ **do**
      $v \leftarrow$ head of edge $e$
      $p \leftarrow$ weight of edge $e$ * $P^d(v, d)$
      $P[i] \leftarrow p$
      $i \leftarrow i + 1$
    **end for**
    $p1 \leftarrow 1$
    **for** $j \leftarrow 1$ to $(i - 1)$ **do**
      $p2 \leftarrow 1 - P[j]$
      $p1 \leftarrow p1 * p2$
    **end for**
    $P^d(u, d) \leftarrow 1 - p1$
    **if** $u = s$ **then**
      print $P^d(s, d)$
      exit
    **end if**
  **end for**
  $L \leftarrow L - 1$
**end while**

**Algorithm 2** : Optimal computation of Delivery Probability

Input $\leftarrow$ All paths $PATH_1, PATH_2, ...$ from $s$ to $d$
$m \leftarrow$ total number of paths
$P^d(s, d) \leftarrow 0$
**for** $n \leftarrow 1$ to $m$ **do**
  coefficient $\leftarrow (-1)^{n-1}$
  **for** start $\leftarrow 1$ to $m$ **do**
    All edges are un-marked
    **for** path-index $\leftarrow$ start to (start+$n-1$) modulo $n$ **do**
      Mark all edges in path $PATH_{path-index}$
    **end for**
    term $\leftarrow$ product of all probabilities of marked edges
    $P^d(s, d) \leftarrow P^d(s, d) +$ (coefficient*term)
    **if** $n = m$ **then**
      break
    **end if**
  **end for**
**end for**
print $P^d(s, d)$

## IV. The SOLAR Protocols

In a conventional MANET (as opposed to intermittently connected networks), Sociological Orbit aware Location Approximation and Routing (SOLAR) framework uses a concept of "acquaintance" similar to that in our work in Acquaintance Based Soft Location Management (ABSoLoM) protocol [10], as well as to some degree the concept of *peer collaboration* (among 'acquaintances') in [3]. For a detailed description of SOLAR and its application towards MANET the readers are




(a) $G = (V, E)$ (b) $G_k = (V, E_k), k = 2$ (c) $G_{sp} = (V, E_{sp})$ (d) $G' = (V, E')$ where dotted edges are *special edges*

Fig. 4. Steps in preparing a network graph for the application of Approximation Algorithm 1

referred to our prior work in [11], [12]. In an ICMAN however, we apply the concepts of "contact probability" and "delivery probability" in proposing a series of multi-path protocols to address the inherent network differences between a MANET and an ICMAN.

### *A. Static SOLAR-KSP Algorithm*

In this version of SOLAR, we assume that each node knows of every other node's hub list and its associated transition probability matrix. Also, as the simulation proceeds we assume nodes to be able to communicate with other nodes within radio range even while traveling from one hub to another. Under this assumption, each node distributively does the following: First, every node computes the *contact probability* with every other node, taking into consideration the communication of nodes within hubs only. Second, every node computes the *delivery probability* to all other nodes. As mentioned in Section III-A, the method for obtaining the contact probability of two nodes within a specified time interval $T$ is computationally expensive. Thus, for the sake of simplicity we estimated the contact probability in our simulation from the actual observed mobility patterns of every pairs of nodes for the duration of the entire simulation. However, it should be noted that in the real world the computation of this contact probability as suggested by our analytical model in Section III-A would need to be done only once at start (and may change only rarely), and hence is practical. For computing the *delivery subgraph* associated with this SOLAR protocol we use the Dijkstra's Shortest Path algorithm [9], and then compute the delivery probability as described next.

Hub list information of all nodes can be formally represented as a weighted graph $G = (V, E)$, where $V$ is the set of all the nodes, and $E$ is the set of weighted edges between every pair of nodes that have at least one hub in common. Let $P(u, v)$ be the contact probability of nodes $u$ and $v$. Then the weight of edge $(u, v)$ is given by:

$$w(u, v) = \log\left(1/P(u, v)\right).$$

In this weighted graph, each node applies a variation of the Dijkstra's Shortest Path algorithm to find $k$ shortest paths (KSP) to every other destination, such that:

1) a path with the minimum total weight is chosen first
2) each path has a different next hop node from source.

Note that due to condition 2, a node may have less than $k$ shortest paths. In any case, a node orders the $k$ paths for each destination in descending order of delivery probability.

Once these KSPs are constructed, a node only needs to maintain the next hops for each of the paths (maximum of $k$ entries per destination nodes). When the source has a packet to send to the destination, it first checks if the destination is within radio range, in which case the packet is directly delivered. Else, it caches a copy of the packet for a pre-determined time interval $T$, during which it may adopt one of the following "forwarding schemes":

1) *send-to-all*: send copies of the packet to all the next hops on the $k$-best paths that come within radio range
2) *send-to-best*: send a copy of the packet to the next hop node on the best path only, if it comes within radio range
3) *send-to-any*: send a copy of the packet to any of the next hop nodes on the $k$-best paths that come within radio range

After the time interval $T$, the packet is purged from the cache. Each node in the path repeats this same process as that packet gets forwarded towards the destination.

**Short Analysis of the Forwarding Schemes:** In the interest of analyzing the relative performance of the *send-to-best* forwarding scheme to that of the *send-to-all* forwarding scheme, we consider a simple example network (Figure 5(a)) where the height ($h$) of the shortest path between the source ($s$) and the destination ($d$) is 2. Let us assume that all nodes (except $d$) have an incoming degree of 1 and an outgoing degree ($k$) of 2. Let $P$ be the *contact probability* for all pairs of nodes. Let us consider all paths from $s$ to $d$ that have a maximum length of $h + 1$ (since paths longer than that have negligible delivery probabilities in our simulation environment). The ratio ($R$) of the delivery probabilities ($P^d$) of the two schemes can then be written as

$$R = \frac{P^d_{best}}{P^d_{all}} = \frac{P^2}{2P(P^2 + P - P^3) - P^2(P^2 + P - P^3)^2}$$

$$or, R = \frac{1}{2(P + 1 - P^2) - (P^2 + P - P^3)^2}$$

By neglecting the higher order term in the denominator, we have

$$R = \frac{1}{2(P + 1 - P^2)}$$

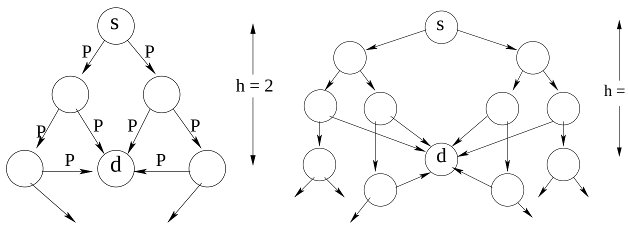
The denominator will be maximum for $P = 0.5$, giving the worst case ratio of $P^d_{best}$ to $P^d_{all}$ as 0.4. If we extend the graph to have $k = 2, h = 3$ (Figure 5(b)) and still consider all paths with maximum lengths of $h + 1$, we can get the simplified ratio

$$R = \frac{P}{4(P^2 + P - P^3)} = \frac{1}{4(P + 1 - P^2)}$$

If we always ignore all higher degree terms in the denominator in each calculation that are less than 0 (this makes the denominator larger and we get a lower bound), we can get the general form of this ratio as

$$R = \frac{P}{k^{h-1}(1-(1-P)(1-P^2)^{k-1})}$$

From the term $k^{h-1}$ in the denominator we can observe that $R$ will decrease exponentially as $h$ increases, and it will decrease polynomially as $k$ increases.




(a) $k = 2$, $h = 2$ (b) $k = 2$, $h = 3$

Fig. 5. Examples for analysis of forwarding schemes in Static SOLAR-KSP

**<u>Performance of Static SOLAR-KSP</u>:** To evaluate the effect of different $k$ values on SOLAR-KSP under each of the 3 schemes: *send-to-all, send-to-best*, and *send-to-any*, we simulated our protocol in the GloMoSim [31] simulator using the default parameter values as shown in Table I. We compare the metrics *data throughput, network byte overhead,* and *average end-to-end data delay*, all of which are defined in more detail in Section V. For the sake of our following discussion, let us define $N^{ksp}_{next}$ for each node $i$ to be the set of nodes (maximum of $k$) that form the next hops on each of the KSPs emanating from node $i$ to a given destination node.

In the results shown in Figure 6 we see that *send-to-best* is independent of the values of $k$ since it only sends data along the best path. Whereas, for the other two schemes: *send-to-any* and *send-to-all*, a value of $k = 1$ makes them similar to *send-to-best*. Hence, we only show protocol results for $k$ from 2 to 4 and results for *send-to-best* represent the performance of all 3 schemes for $k = 1$.

In Figure 6(a) we see that *send-to-all* has the highest throughput, but it also has the highest overhead as seen in Figure 6(b). This is understandable as *send-to-all* uses all of the KSPs computed at start of simulation. More specifically, for larger values of $k$ the overhead of *send-to-all* is seen to grow significantly due to the exponential increase in the number of packets forwarded at each intermediate node. Overall, we get the optimum results (high throughput, but low overhead) for *send-to-all* with $k = 2$. On the other hand, the throughput for *send-to-any* is seen to decrease with increasing values of $k$. This is because as $k$ increases, the cardinality of $N^{ksp}_{next}$ increases, and as *send-to-any* chooses the first available node in $N^{ksp}_{next}$, the difference in the delivery probability of that chosen node to that of the best choice within $N^{ksp}_{next}$ may increase. Scheme *send-to-best* has the lowest overhead as it waits for the best choice in $N^{ksp}_{next}$, and consequently has the highest dropping rate (i.e., lowest throughput) when that best choice is not available within a specified time interval $T$.

In Figure 6(c), we see that for each of the 3 schemes, the end-to-end data delay is proportional to the data throughput as seen in Figure 6(a), with varying values of $k$. This is because, data packets are sent from the source uniformly to nodes that are both near and far, with destination nodes that are near (shorter paths) having a higher probability of receiving a packet correctly. Hence, larger number of packets delivered also implies larger number of packets reaching destination nodes further apart (over longer paths), involving higher values of delay, which consequently increase the average end-to-end delay.

From the results above, it is evident that in our given scenario Static SOLAR-KSP performs the best with the *send-to-all* scheme with $k = 2$. As such, we shall use this variation of Static SOLAR-KSP, also referred to as ***S-SOLAR-KSP***, for our performance comparison in Section V with the other SOLAR protocols, to be described next.

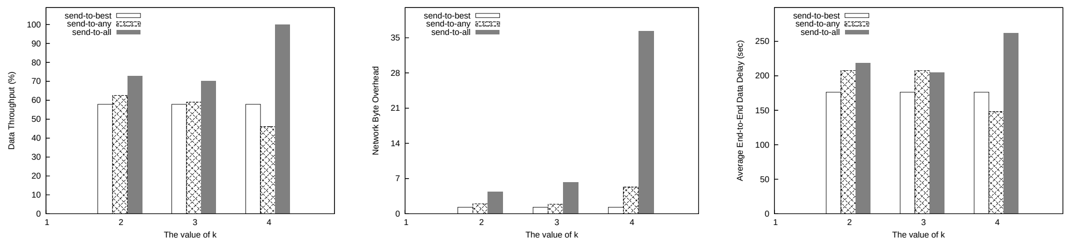
### B. Dynamic SOLAR-KSP Algorithm

In this variation of SOLAR, each node locally computes KSP to every other node in the network, but unlike Static SOLAR-KSP algorithm where nodes try and forward only to nodes within $N^{ksp}_{next}$, a node in Dynamic SOLAR-KSP algorithm shall forward to at most $k$ nodes from amongst its current neighbors with higher delivery probability to the destination. To avoid packet duplication, when a node receives any packet in a hub, *it does not try to forward to any other nodes in the same hub*. It waits till it moves to a new hub before repeating the forwarding process described above. Also, nodes are assumed to not communicate with any other node (except with the destination) when they travel from one hub to another. Thus, Dynamic SOLAR-KSP combines static hub based information with dynamic selection of next hop on the path towards the destination. In our simulations, we choose the value of $k = 2$ and refer to this version as ***D-SOLAR-KSP***.

### C. SOLAR-HUB Algorithm

This SOLAR variation is very unlike the other versions discussed so far in that nodes in this protocol do not compute KSP to every other node in an attempt to forward data along a path of intermediary nodes to the destination. Instead, the source node tries to forward data to its neighbors that have a higher *delivery probability to the hubs* visited by the destination (and not to the destination itself). To explain this next hop selection process in more detail, lets define a few terms:

- $P^d_{n_i h_j}$: The delivery probability of node $n_i$ to hub $h_j$.
- $P^t_{n_i h_j}$: The probability of node $n_i$ to travel to hub $h_j$ ever during simulation.
- $h(n_i)$: The hub that node $n_i$ is going to visit next.
- $P^c_{n_i n_k}(h_j)$: The probability for contact of nodes $n_i$ and $n_k$ in hub $h_j$ ever during simulation.
- $N(n_i)$: Neighbors of node $n_i$.



(a) Data Throughput (%) (b) Network Byte Overhead (c) Average End-to-End Data Delay (sec)

Fig. 6. Performance of Static SOLAR-KSP with varying $k$

Given this, every node $n_i$ can dynamically and distributively compute the delivery probability to every other hub $h_j$ as

$$P^d_{n_i h_j} = \max(P^t_{n_i h_j},\ \max_k(P^c_{n_i n_k}(h(n_i)) * P^t_{n_k h_j}))$$

Thus, when a node $n_s$ wants to forward a data packet to one of the hubs $h_j$ in the destination node's hub list, it will pick as the next hop the node

$$\{n_i \mid \max(P^d_{n_i h_j}),\ n_i \in N(n_s)\}\ \textit{iff}\ P^d_{n_i h_j} > P^d_{n_s h_j}$$

More specifically, nodes are assumed to know the next hub they are going to visit after they move out of their current hub (note that SOLAR-KSP versions did not make this assumption), in addition to every other node's hub list probability distribution. We believe that in the real world, this assumption may often be a realistic one, as users (nodes) move with some purpose in mind. We study 3 different strategies for routing under this framework:

Strategy-1: When source has data to send, it shall forward a copy of the data packet to a maximum of $k/2$ neighbors with higher probability of visiting the "most visited" hub of the destination, and to a maximum of $k/2$ *different* neighbors with higher probability of visiting the "second most visited" hub of the destination. If no such neighbors exist, source caches the packet for a specified timeout period. To avoid loops, each downstream node that receives a packet in a particular hub, only repeats this forwarding process when it moves into a different hub. Once a packet reaches a node that is within either the most or, the second most visited hub of the destination, it is cached for a specified timeout period for the destination node.

Strategy-2: When a source has data to send, it shall forward a copy of the data packet to a maximum of $k$ neighbors who have higher probabilities to visit the "most visited" hub of the destination when compared to all nodes (including the source) in the neighborhood of the source. Like before, each downstream node repeats this process upon moving into a new hub, till the packet arrives at a node that is within the most visited hub of the destination, where it shall stay cached for the destination for a specified timeout period.

Strategy-3: This is a mix of the 2 strategies described above, where the source of the data packet follows *Strategy-1*, while all other downstream nodes that receive a packet follow *Strategy-2*. Similar to D-SOLAR-KSP, nodes in this version are assumed to not communicate with any other node (except with the destination) as they travel in between hubs.

We compared all the 3 strategies via simulation studies, and found *Strategy-3* with $k = 2$ (referred to as ***SOLAR-HUB*** henceforth) to perform the best. However, due to space constraints we are not able to include these comparison results in this paper.

## V. Performance Analysis of SOLAR Protocols

In this section, we describe the extensive simulation study we carried out to compare the performance of the SOLAR protocols: ***S-SOLAR-KSP, D-SOLAR-KSP***, and ***SOLAR-HUB*** using the GloMoSim [31] simulator. We included the Epidemic Routing protocol [27] (referred to as ***EPIDEMIC*** in this paper) in our comparisons because of its simple yet efficient performance in face of general intermittently connected networks. For the simulation scenario, we considered an ICMAN built within a corporate campus consisting of several buildings (hubs). Corporate employees spend most of their time within the hubs and intermittently move in between hubs. To model realistic speeds of mobile users within such a network, we considered the work in [18], [29] and fixed the ORBIT Inter-Hub and Intra-Hub time/speed parameters, along with the other simulation parameters as shown in Table I. We chose three metrics to evaluate the performance of each protocol as described below:

Data Throughput: This metric is defined as the ratio of the total number of data packets received correctly by all destinations, to the total number of data packets generated by all sources, for the entire duration of the simulation.

Network Byte Overhead: This metric is defined as the ratio of the amount of control and data information (measured in bytes) that got transmitted from a node to its neighbor, to the total amount of data information (measured in bytes) that was received correctly, for the entire duration of the simulation. In the SOLAR variations, the control packets consist of only *Hello* packets, whereas in Epidemic Routing, the control overhead is due to the exchange of *Hello* and *Summary Vector* messages.

Average End-to-End Data Delay: The end-to-end data delay is defined as the time interval (measured in seconds) between the generation of a data packet at the source and its reception at the destination. This value is averaged over all packets correctly received at the destination to give the average.

In what follows, we will examine how the total number of hubs, and the total number of nodes affect the protocol performance. To this end, we vary one of these two factors

TABLE I
SIMULATION PARAMETERS

| | | | |
|---|---|---|---|
| *GENERAL PARAMETERS* | | | |
| Simulation Duration (each run) | 3000s | Terrain Size | 1000m x 1000m |
| Number of Nodes (*Users*) | *Vary*, (Default= 100) | Radio Range | 125m |
| Cache Size | *Vary*, (Default= 200 Packets) | Cache Timeout | *Vary*, (Default= 400s) |
| MAC Protocol | IEEE 802.11 | Mobility Model | Probabilistic Orbit (RW + P2P) |
| *ORBIT PARAMETERS* | | | |
| Number of Hubs | *Vary*, (Default= 15) | Hub Size | 50m x 50m |
| Hub Stay Time | Exponential (Mean= 50s) | Hub List Timeout | None |
| Hub List Size | 2 to Number of Hubs | Inter-Hub Transition Time | Exponential (Mean= 40s) |
| Intra-Hub Pause | 1s | Intra-Hub Speed | 1m/s-10m/s |
| *TRAFFIC PARAMETERS* | | | |
| CBR connections | 30 (120 packets each) Random | Data Payload | 1460 bytes per packet |

while fixing all other parameters to their default values. Each plot point in the results is averaged over 6 different simulation runs with varying random seeds.

### A. Variation in Number of Hubs

The number of hubs in the terrain affects protocol performance due to its direct impact on the expected node density within hubs (given a fixed number of nodes), and the hub list sizes of each node.

*Data Throughput*: As seen in Figure 7(a), the data throughput of SOLAR-HUB is the most consistent with a varying number of hubs. This is attributed to the assumption that each node knows of the next hub it is going to visit, which aids in the next hop selection process. In the case of the KSP based SOLAR protocols: D-SOLAR-KSP and S-SOLAR-KSP, increasing number of hubs decreases the node density within hubs and reduces the likelihood of finding suitable next hops with higher delivery probability to the destination. Amongst the two however, nodes in D-SOLAR-KSP have more freedom of choice in selecting the next hop, and can make best use of whichever nodes are available within radio range. Thus, the throughput of D-SOLAR-KSP is comparable to that of SOLAR-HUB. In S-SOLAR-KSP however, since nodes only forward to other nodes within their $N_{next}^{ksp}$ set, decreasing node densities significantly decrease the probability of finding such nodes in the same hub within a specified time, thereby adversely affecting their throughput with increasing number of hubs. Similarly in EPIDEMIC, since the only way of data dissemination is via data exchange amongst neighbors, decreasing node densities within hubs causes the throughput to decrease. Moreover, since in our Probabilistic Orbit model, hubs do not overlap, nodes in EPIDEMIC cannot communicate with other nodes in other hubs, as they are far apart.

*Network Byte Overhead*: From Figure 7(b), we note that the network byte overhead incurred by SOLAR-HUB is the highest and it keeps increasing with increasing number of hubs. This is easy to understand since in SOLAR-HUB nodes try and forward a copy of a packet whenever they enter a new hub. Thus, as the number of hubs increases, the hub list size of each node grows and more packets get forwarded in the network. In D-SOLAR-KSP, the overhead is seen to be less than SOLAR-HUB primarily due to the reason that in D-SOLAR-KSP, it is possible for two nodes to forward different copies of the same packet for a specific destination to the same neighbor in a hub, with highest delivery probability to the destination. In this scenario, that neighbor will drop one of the packets and only forward one copy onward, reducing the total number of packet transmissions in the network. In SOLAR-HUB however, since packets are forwarded to the "most visited" and "second most visited" hubs of a destination, such dropping of duplicate packets at intermediate nodes are less frequent. S-SOLAR-KSP is more or less consistent with varying number of hubs since it only tries to forward to nodes within $N_{next}^{ksp}$, which is independent of the hub list size of nodes. However, with large number of hubs, the probability of meeting with favorable neighbors reduce considerably, reducing the the total data packets forwarded, which in turn lowers the overhead. The overhead in EPIDEMIC could not be included in this same graph as we found it to be an order of 100 more than any of the SOLAR protocols. In EPIDEMIC, a pair of nodes exchange data whenever they have space in their buffer and each has some data packet that the other does not. Thus, the number of data exchanges are enormous, leading to such high overhead.

*Average End-to-End Data Delay*: As seen in Figure 7(c), the end-to-end data delay in S-SOLAR-KSP is not much affected by the number of hubs. This is due to the fact that nodes in S-SOLAR-KSP only use the pre-computed KSPs that do not depend on the number of hubs. In D-SOLAR-KSP however, the dynamic and greedy next hop selection favors the end-to-end data delay when there are only a small number of hubs by making sure that the packet never waits for any particular neighbor to appear. However, when the number of hubs grow larger, such a greedy selection may cause the packet to reach the destination via a longer path (when compared to S-SOLAR-KSP), or may suffer delays at some intermediate node in the dynamically formed path, leading to higher end-to-end delay. In SOLAR-HUB, longer hub lists may cause the destination to visit the "most visited hub" less frequently, leading to higher delays. EPIDEMIC displays the highest end-to-end delay since it only relies on eventual dissemination of the data packet to the destination, which may take a long time when number of hubs are large and nodes meet in hubs less frequently.

### B. Variation in Number of Nodes

In this section, we study the effect of varying the number of nodes in the network on the performance of the protocols. Given a fixed terrain size and radio transmission range, the total number of nodes directly impact the network connectivity.

*Data Throughput*: From Figure 8(a), it is clear that as the number of nodes increases, all protocols benefit, for their own respective reasons. With a fixed number of hubs, an increase in the number of nodes also increases the node density within hubs, which as noted in the discussions in Section V-A positively affect all protocol performance. The relative performance amongst the protocols are similar to that seen in Figure 7(a) with SOLAR-HUB and D-SOLAR-KSP performing much better than S-SOLAR-KSP and EPIDEMIC. Since nodes in S-SOLAR-KSP only wait for $k$ of its neighbors in $N_{next}^{ksp}$ to come within radio range, it cannot fully leverage the increase in the number of nodes. On the contrary, higher node density provides the greedy approach in D-SOLAR-KSP with more choices of next hop, leading to better performance than S-SOLAR-KSP. In EPIDEMIC, larger number of nodes will cause a larger number of data exchanges. Given a fixed cache size this can lead to eventual dropping of packets, limiting the throughput to a low value with increasing number of nodes.

*Network Byte Overhead*: The relative performance of the overhead of all the protocols shown in Figure 8(b) is also similar to that seen in Figure 7(b), and much for the same reasons. With an increased node density within hubs, nodes in SOLAR-HUB almost always find $k$ neighbors to forward to, leading to an increased overhead. D-SOLAR-KSP displays lower overhead than SOLAR-HUB for reasons discussed for *Network Byte Overhead* in Section V-A. S-SOLAR-KSP has the lowest overhead since it only forwards to at most $k$ preselected nodes, that may or may not appear within radio range in a specified timeout period. EPIDEMIC once again was omitted from this graph due to its much larger values of overhead.

*Average End-to-End Data Delay*: The end-to-end delay for all protocols decrease on an average with a larger number of nodes, as seen in Figure 8(c). Although, EPIDEMIC protocol eventually delivers data, due to its lack of direction (toward the destination node or hub) it suffers high end-to-end delay. S-SOLAR-KSP shows consistent performance by only selecting nodes from $N_{next}^{ksp}$, which is still bounded by $k$, and thus independent of the total number of nodes. On the other hand, increasing node density within hubs appear to cause significant improvements in the end-to-end delay for nodes in both SOLAR-HUB and D-SOLAR-KSP, as nodes find larger number of suitable nodes to efficiently forward their data packets.

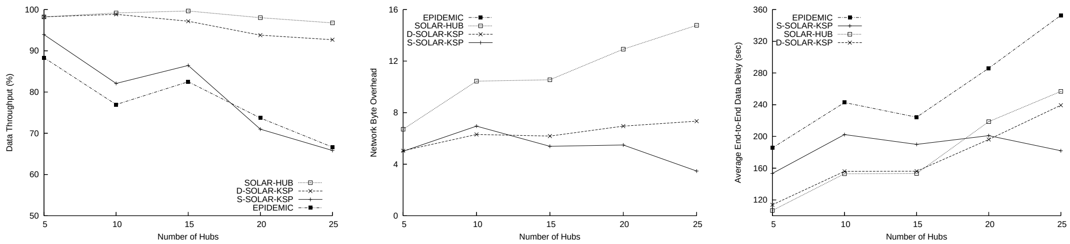
We also studied the effect of varying hub sizes while keeping the radio range a constant, but found similar relative results for each hub size considered. Due to space constraints we are unable to include those results in this paper. However, it is evident from the above performance comparison that all the SOLAR variations perform much better than the conventional approach of Epidemic Routing, without having to compromise on any of data throughput, network overhead or end-to-end data delay.

## VI. Effect of Cache Size and Cache Timeout

We further study the effect of two vital routing parameters on our SOLAR protocols. First is the maximum number of packets that a node can hold on to simultaneously, both for itself and for other nodes, referred to as the *Cache Size*. Second is the maximum time for which any packet is cached within a node, referred to as the *Cache Timeout*. All other simulation parameters have the default values mentioned in Table I.

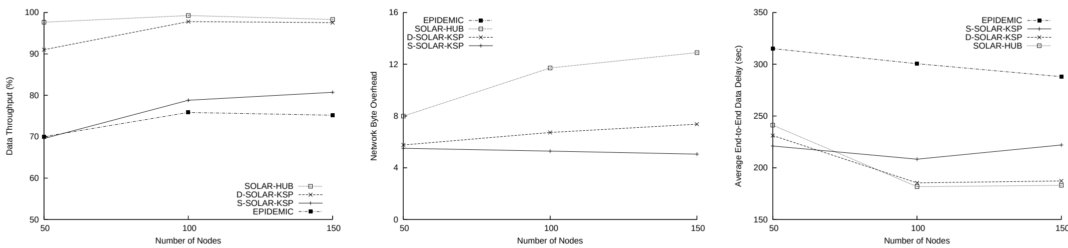
**Cache Size:** In this scenario, the *Cache Timeout* was set to the simulation time to effectively have no timeout, while the *Cache Size* was varied. As seen in Figure 9(a), the performance of all our SOLAR protocols improve with increasing cache size. With sufficient amount of storage space in all nodes, we see that both D-SOLAR-KSP and SOLAR-HUB are able to attain $100\%$ throughput via their dynamic selection of next hop from amongst the available neighbors. Although, the throughput in S-SOLAR-KSP increases with larger cache sizes, it is still shy of $100\%$ indicating some packet loss. This is easy to understand given the fact that in S-SOLAR-KSP, nodes wait and forward only to nodes in their $N_{next}^{ksp}$ set. In most cases, the *contact probability* $p$ of a node with those nodes in $N_{next}^{ksp}$ within the simulation time period is less than $1$. Thus, with probability $1 - p$ packets will be dropped at a node when none of the expected nodes come within radio range, preventing the throughput from reaching $100\%$. It is obvious that given infinite time, S-SOLAR-KSP will be able to forward all packets eventually. This is the same reason why in Figure 9(c) we find the average end-to-end delay in the dynamic SOLAR protocols: D-SOLAR-KSP and SOLAR-HUB, to be far less than that in S-SOLAR-KSP, which waits to forward only to specific nodes. Also, as mentioned earlier in our performance analysis of *Static SOLAR-KSP* in Section IV-A, nodes uniformly send data to other nodes over both short and long paths, with the shorter paths having more probability of success. Hence, an increase in the throughput (with increasing cache size) also indicates more data delivered over longer paths, which in turn leads to an increase in the average end-to-end data delay for S-SOLAR-KSP, as seen in Figure 9(c).

As the cache size increases, each node can store more packets and can in turn forward more packets, leading to increased overhead as shown in Figure 9(b). Since, nodes in S-SOLAR-KSP wait to forward to specific nodes, the number of transmissions is lower for small cache sizes, leading to lower overhead than the other two SOLAR protocols. However, as the cache size increases to hold more packets, S-SOLAR-KSP may have multiple transmissions of the same packet within




(a) Data Throughput (%) (b) Network Byte Overhead (c) End-to-End Data Delay (sec)

Fig. 7. Protocol Performance vs. Number of Hubs




(a) Data Throughput (%) (b) Network Byte Overhead (c) End-to-End Data Delay (sec)

Fig. 8. Protocol Performance vs. Number of Nodes

the same hub, unlike in D-SOLAR-KSP and SOLAR-HUB, where nodes forward packets only when they move into a new hub. This explains the sudden rise of overhead in S-SOLAR-KSP with increasing cache size. However, this increase levels off beyond a certain point when the probability of finding a node within $N_{next}^{ksp}$ provides an upper bound to the number of transmissions in a hub, making it independent of the cache size.

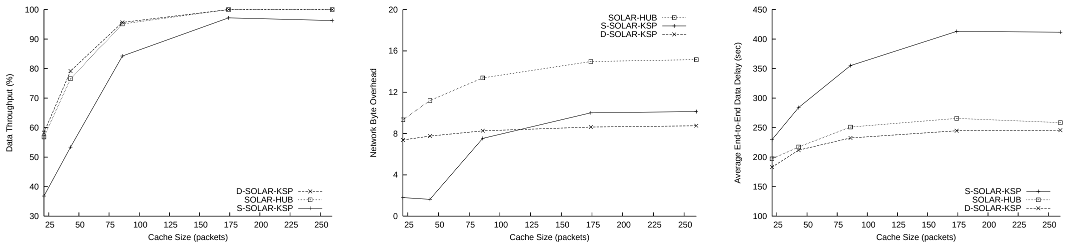
**Cache Timeout:** In this scenario, the *Cache Size* was set to $650$ packets, while the *Cache Timeout* was varied. As seen in Figure 10(a), even with a moderate cache timeout D-SOLAR-KSP and SOLAR-HUB perform well with around $80\%$ throughput. This is because both these protocols keep selecting their next hops from the available set of neighbors, unlike S-SOLAR-KSP, which has to wait for specific nodes in $N_{next}^{ksp}$, and consequently has a much lower throughput with a low cache timeout value. However, with increasing cache timeout, the probability of nodes in S-SOLAR-KSP to find nodes in $N_{next}^{ksp}$ within that timeout period increases, thereby increasing the throughput significantly. The corresponding average end-to-end data delay incurred by the protocols also show proportional results in Figure 10(c), with higher average delay for higher throughput. The reason is similar to that described before, where more packets getting delivered indicate greater number of packets being delivered over longer paths (and hence with longer delay), which increase the average end-to-end delay. More specifically, for nodes in S-SOLAR-KSP this effect of increase in cache timeout is more significant since each node waits for that time period to try and forward packets. Just as this improves the probability of finding expected nodes, thereby increasing the throughput as seen before, it also increases the average end-to-end delay by an appreciable amount.

The relative overhead performances of the $3$ protocols in Figure 10(b) is seen to be similar to that shown in Figure 9(b), and for similar reasons. SOLAR-HUB has more number of transmissions at the intermediate nodes than D-SOLAR-KSP due to reasons discussed for *Network Byte Overhead* in Section V-B. The overhead in S-SOLAR-KSP however, shows less drastic changes with varying cache timeout than that seen for varying cache size in Figure 9(b). This is because the lowest cache timeout we considered ($200$ seconds) was more than the average hub stay time (exponentially distributed with mean of $50$ seconds), giving every node a fair opportunity to meet with nodes in their $N_{next}^{ksp}$. Overall, given more time every protocol has a higher probability of meeting suitable nodes for forwarding packets, thereby displaying a general increase in network overhead with increasing cache timeout.

## VII. OTHER RELATED WORK

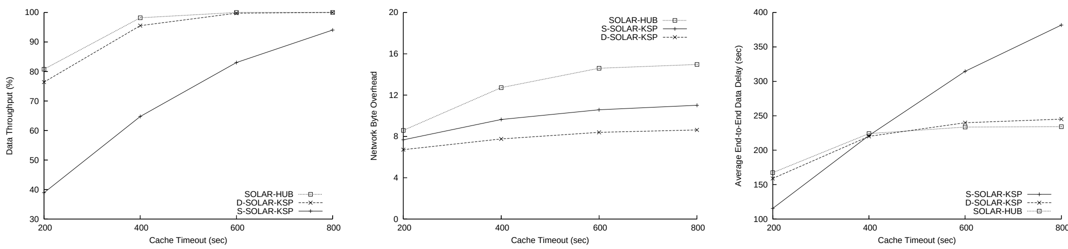
Routing in an ICN has received significant interest from the research community recently. Several routing-related issues in ICN were addressed in [17], which focused mainly on networks with known connectivity patterns, such as satellites with fixed paths, or busses with fixed routes. The authors developed several algorithms to analyze the knowledge to performance relationship in different protocols and demonstrated that their algorithms performed better with more network knowledge. However, for the availability of such global knowledge they assumed the presence of certain "knowledge oracles" that may not be applicable to most mobile ad hoc networks, where users' mobility follow much less predictable schedules.

Routing issues were also addressed in general intermittently connected networks in [27], where the authors proposed the Epidemic Routing protocol that relies on data buffering and




(a) Data Throughput (%) (b) Network Byte Overhead (c) End-to-End Data Delay (sec)

Fig. 9. SOLAR Performance vs. Cache Size




(a) Data Throughput (%) (b) Network Byte Overhead (c) End-to-End Data Delay (sec)

Fig. 10. SOLAR Performance vs. Cache Timeout

node mobility to spread messages in the network. The so-called "summary vectors" were used for nodes to selectively exchange data packets, in order to limit the number of data transmissions. Similar work ( [4], [13]) on data dissemination was also done for sensor and ad hoc networks.

The concept of Epidemic Routing was extended upon by the authors in [22], where they proposed a probabilistic routing scheme whereby each node maintains the so-called "delivery predictability" to each known destination, and uses this metric to make routing decisions. However, their delivery predictability may decay with time, unlike our contact probability that remains constant throughout the entire simulation by virtue of the hub based mobility profile of nodes extracted from the underlying orbital mobility. In [23], the authors proposed a context-aware adaptive routing algorithm that takes into account the suitability of a node for carrying a message based on context information of the node at multiple dimensions. More recently, the authors in [21] suggested an algorithm that relies on vehicles to act as mobile routers, which connect disconnected sensor networks to a known destination.

Node mobility was shown to affect routing protocol performance in [2]. The work in [14] exploited node mobility in MANET as a type of multi-user diversity, and showed that node mobility may actually help increase the theoretical capacity of a MANET. In addition, research has also been done ( [25], [26], [28]) on MANET to take advantage of mobility information obtained via continuous location tracking and "micro-level" mobility prediction. However, such methods and solutions are not applicable to ICMAN due to its inherent intermittent connectivity. Note that our orbital movement pattern differs from such existing mobility patterns in that it neither models the motion of the users at a micro-level (i.e., on small time scales or within small distances), nor simply predicts user locations via historical/statistical tracking information ( [25], [26], [28]). It also differs from the deterministic mobility patterns assumed within ICN, where either exact locations of a node can be predicted with an appropriate "oracle", or no location information is available.

The authors in [15] and [32] respectively studied the effects of controlled message flooding and controlled mobility in large scale ICN. Along the same lines of informed message passing that exploits the mobility information of network users to some extent, the authors in [7] proposed the introduction of autonomous agents that can adapt their motion within the network using multi-objective control methods to increase network efficiency. Although, the introduction of such agents with controlled/programmable motion simplifies the problem of routing, the ready availability/deployment of such agents within an ICN may not be too practical. To the best of our knowledge, our work is the first to explore the implication of the macro-level partially deterministic sociological orbits involving a list of hubs and its application to location approximation and routing in ICMAN, despite its practicality.

As part of our ongoing research, we are in the process of analyzing real time experimental data for user mobility and user network access patterns to establish the existence of hubs and orbits to validate our initial claim. Along with that, we are working on developing an efficient approximation algorithm for computing the delivery subgraph associated with one of our SOLAR strategies, and computing the delivery probability from it.

## VIII. Conclusion

Efficient location management and routing in mobile ad hoc networks has long been a challenging problem for researchers. Off late, the same problem is being studied under the additional constraints of network disconnections, common within Intermittently Connected Networks (ICN). The unavailability of contemporaneous end-to-end path from a source to a destination through intermediary peers renders most reactive and proactive routing protocols useless. Although, some work [17] has been suggested in literature that assumes deterministic mobility, the actual motion of users in real life are not as predictable. In the absence of any practical knowledge oracle to provide up-to-date network connectivity information, the routing problem remains a daunting one. To that end, researchers in [7], [15], [32] have also proposed methods to either exploit or influence the network mobility (e.g., by introduction of foreign mobile agents into the ad hoc network and controlling their motion) to improve network efficiency. However, such methods may not be too practical (e.g., foreign agents may not always be freely available and their ready deployment may not always be feasible).

In this paper, our contribution is two-fold. First, we propose our novel *Probabilistic Orbit* mobility model based on macro-level sociological orbits involving a set of hubs, that is well suited for semi-deterministic node mobility within a special class of ICN formed of mobile ad hoc users called ICMAN. Second, we propose a series of *Sociological Orbit aware Location Approximation and Routing (SOLAR)* algorithms, that leverage upon the underlying orbital mobility information to efficiently route data within an ICMAN. We present strong theoretical analysis of our SOLAR framework and support it with extensive simulation results that decisively show that SOLAR protocols outperform other conventional routing approaches (e.g., Epidemic Routing [27]) in an ICMAN, in terms of higher data throughput, lower network overhead, and lesser end-to-end data delay. We have already established the simplicity and efficiency of using our SOLAR framework in a Mobile Ad Hoc Network (MANET) [11], [12]. In this work, we present equally strong results to prove that our proposed SOLAR is as strong a candidate of choice when it comes to meeting with the challenges of routing within an ICMAN.

## References


[1] Abowd, G., and Mynatt, E. Charting past, present, and future research in ubiquitous computing. *ACM Transaction on Computer Human Interaction 7*, 1 (March 2000), 29–58.

[2] Bai, F., Sadagopan, N., and Helmy, A. Important: a framework to systematically analyze the impact of mobility on performance of routing protocols for adhoc networks. *Proceedings of IEEE INFOCOM '03 2* (March 2003), 825–835.

[3] Banerjee, S., and Chrysanthis, P. Peer support in a mobile world. *Proceedings of NSF Workshop on Context-Aware Mobile Database Management* (January 2002).

[4] Beaufour, A., Leopold, M., and Bonnet, P. Smart-tag based data dissemination. *First ACM International Workshop on Wireless Sensor Networks and Applications (WSNA02)* (June 2002).

[5] Bolch, G., Greiner, S., de Meer, H., and Trivedi, K. S. *Queueing networks and Markov chains.* A Wiley-Interscience Publication. John Wiley & Sons Inc., New York, 1998. Modeling and performance evaluation with computer science applications.

[6] Bollobás, B. *Random graphs*, second ed., vol. 73 of *Cambridge Studies in Advanced Mathematics*. Cambridge University Press, Cambridge, 2001.

[7] Burns, B., Brock, O., and Levine, B. N. Mv routing and capacity building in disruption tolerant networks. *In Proceedings of IEEE INFOCOM '05* (March 2005).

[8] Cerf, V., Burleigh, S., Hooke, A., Torgerson, L., Durst, R., Scott, K., Fall, K., and Weiss, H. Delay tolerant network architecture. *Internet Draft draftirtf-dtnrg-arch-02.txt (work in progress)* (July 2004).

[9] Dijkstra, E. W. A note on two problems in connexion with graphs. *Numerische Mathematik 1* (1959), 269–271.

[10] Ghosh, J., Philip, S., and Qiao, C. Acquaintance Based Soft Location Management (ABSoLoM) in MANET. *Proceedings of IEEE Wireless Communications and Networking Conference (WCNC) '04* (March 2004).

[11] Ghosh, J., Philip, S. J., and Qiao, C. Poster abstract: Sociological orbit aware location approximation and routing (solar) in manet. *Presented as a Poster in ACM Mobihoc '05, Champaign, IL* (May 2005).

[12] Ghosh, J., Philip, S. J., and Qiao, C. Sociological orbit aware location approximation and routing in manet. *In Proceedings of IEEE Broadnets '05, Boston, MA* (October 2005).

[13] Glance, N., Snowdon, D., and Meunier, J.-L. Pollen: using people as a communication medium. *First ACM International Workshop on Wireless Sensor Networks and Applications (WSNA02) 35*, 4 (March 2001), 429–442.

[14] Grossglauser, M., and Tse, D. N. C. Mobility increases the capacity of ad hoc wireless networks. *IEEE/ACM Transactions on Networking 10*, 4 (August 2002), 477–486.

[15] Harras, K. A., Almeroth, K. C., and Belding-Royer, E. M. Delay tolerant mobile networks (dtmns): Controlled flooding in sparse mobile networks. *To appear in Proceedings of Networking '05, Waterloo, Ontario, Canada* (May 2005).

[16] Harrison, P. G., and Knottenbelt, W. J. Passage time distributions in large markov chains. *SIGMETRICS Perform. Eval. Rev. 30*, 1 (2002), 77–85.

[17] Jain, S., Fall, K., and Patra, R. Routing in a delay tolerant network. *Proceedings of ACM SIGCOMM'04* (September 2004).

[18] Knoblauch, R. L., Pietrucha, M. T., and Nitzburg, M. Field Studies of Pedestrian Walking Speed and Start-Up Time. *Transportation Research Board Records*, 1538 (1996).

[19] Krivelevich, M., and Vu, V. H. Choosability in random hypergraphs. *J. Combin. Theory Ser. B 83*, 2 (2001), 241–257.

[20] Ladd, A., Bekris, K., Rudys, A., Wallach, D., and Kavraki, L. On the feasibility of using wireless ethernet for indoor localization. *IEEE Transactions on Robotics and Automation 20*, 3 (June 2004), 555–559.

[21] LeBrun, J., Chuah, C. N., Ghosal, D., and Zhang, H. M. Knowledge-based opportunistic forwarding in vehicular wireless ad hoc networks. *To appear in IEEE Vehicular Technology Conference* (Spring 2005).

[22] Lindgren, A., Doria, A., and Schelen, O. Poster: Probabilistic routing in intermittently connected networks. *Proceedings of The Fourth ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc 2003)* (June 2003).

[23] Musolesi, M., Hailes, S., and Mascolo, C. Adaptive routing for intermittently connected mobile ad hoc networks. *Proceedings of IEEE 6th International Symposium on a World of Wireless, Mobile and Multimedia Networks (WOWMOM'05), Taormina, Italy* (June 2005).

[24] Remagnino, P., and Foresti, G. Ambient intelligence: A new multidisciplinary paradigm. *IEEE Transactions on Systems, Man and Cybernetics 35*, 1 (Jan 2005), 1–6.

[25] Samal, S. Mobility pattern aware routing in mobile ad hoc networks. *MS Thesis, Virginia Polytechnic Institute and State University* (May 2003).

[26] Su, W., Lee, S.-J., and Gerla, M. Mobility prediction and routing in ad hoc wireless networks. *International Journal of Network Management 11*, 1 (February 2001), 3–30.

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

[28] Wang, W., and Akyildiz, I. F. On the estimation of user mobility pattern for location tracking in wireless networks. *Proceedings of IEEE Globecom '02* (November 2002), 619–623.

[29] Williams, A. Cycling Speed. *Article in Peak Performance Online (http://www.pponline.co.uk/encyc/0065.htm)*, 65.

[30] Yoon, J., Liu, M., and Noble, B. Random Waypoint Considered Harmful. *Proceedings of IEEE INFOCOM '03 2* (March 2003), 1312–1321.

[31] Zeng, X., Bagrodia, R., and Gerla, M. Glomosim: a library for parallel simulation of large-scale wireless networks. *Proceedings of the 12th Workshop on Parallel and Distributed Simulations (PADS) '98* (May 1998), 154–161.

[32] Zhao, W., Ammar, M., and Zegura, E. Controlling the mobility of multiple data transport ferries in a delay-tolerant network. *In Proceedings of IEEE INFOCOM '05* (March 2005).