Sociological Orbit Aware Routing in MANET

Joy Ghosh, Sumesh J. Philip, 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, sumeshjp, qiao}@cse.buffalo.edu

***Abstract*— Mobility affects routing protocol performance within a Mobile Ad Hoc NETwork (MANET). Past research on mobility models and framework was motivated by the need to provide a simulation platform for evaluating routing protocol performance via realistic scenarios. Mobility information of individual nodes has also been used to improve routing decisions (e.g., choice of next hop, link break prediction) in a MANET.**

In this paper, we introduce a novel concept of integrating macro-mobility information obtained from the sociological movement pattern of mobile MANET users, into routing. The extraction of this mobility information is based on our observation that the movement of a mobile user exhibits a partially repetitive 'orbital' patten involving a set of 'Hubs' in practise. This partially deterministic movement pattern is both practical and useful in locating nodes and routing packets to them without the need for constant tracking or flooding. Leveraging this Hub-based orbital pattern, we propose a Sociological Orbit Aware Routing (SOAR) protocol. Through extensive performance analysis we show that SOAR significantly outperforms conventional routing protocols like Dynamic Source Routing (DSR) and Location Aided Routing (LAR) in terms of higher data throughput, lower control overhead, and lower end-to-end delay.

***Index Terms*— Mobility models, Routing protocol, Ad hoc wireless networks, Performance analysis**

I. Introduction

A Mobile Ad Hoc NETwork (MANET) is an infrastructure less group of wireless mobile devices that willfully cooperate to forward packets for one another. Recently, node mobility has been shown to have a significant impact on the routing protocol performance [1]. This led to numerous studies on the practicality of the different mobility models that are commonly deployed. The Random Waypoint model [2] has been a favorite for its simplicity and suitability for theoretical study and analysis. In this model, a node keeps choosing destination points randomly within a terrain and approaches it linearly with a velocity randomly selected from a specified range. In reality however, nodes (i.e. MANET users) move with some purposes in mind (e.g., going from a conference room to a cafeteria) in addition to certain restrictions (e.g., geographical constraints) resulting in certain amount of determinism in their motion.

In the light of growing need for more practical and realistic mobility modeling, a new line of research has emerged, focusing on several *Entity based* ([3], [4], [5]), *Group based* ([6], [7], [8]), and *Scenario based* ([9], [10], [11]) mobility models/frameworks. The *Entity based* models are driven by the individual node characteristics, while the *Group based* models concentrate on the collective movement of a group of nodes that deviate marginally from the characteristics of a leader node. Alternately, the *Scenario based* models account for the geographical constraints on real life movement.

Parallel to the study of the practical mobility models, work has been done on routing protocols to cope with the effects of mobility. Among the two categories of routing protocols described in literature: *Proactive* and *Reactive*, the latter is more suited for highly mobile ad hoc networks due to its ability to cope with rapidly changing network topologies. Some existing work has suggested source routing protocols that adopted various optimization techniques, such as caching of paths (e.g., [2]) to reduce path request overhead, and caching of node velocity (e.g., [12]) to approximate node locations. Others ([13], [14]) have suggested the use of virtual backbones to ease the adaptation of the routing protocols to mobility. With the advent of newer and affordable technology like the GPS receivers [15], and other localization techniques, location management schemes coupled with routing strategies (e.g., [16], [17]) have also shown to offer efficient routing solutions for MANET.

On the other hand, the authors in [18] have showed that the mobility of nodes may help increase the theoretical MANET capacity. In addition, work has also been done ([19], [20], [21]) to explore the potential of using the mobility information of individual nodes to facilitate selection of next hop or link break prediction. Most recently, work on Delay Tolerant Networks (DTN) [22] has addressed routing issues with different information oracles under completely deterministic mobility of nodes, such as satellites and buses. However, no prior work has studied 'macro-level' sociological movement patterns of MANET users and their implications on routing protocols and their performances.

In this paper, we identify a partially deterministic 'orbital' movement pattern around some specific places of social interest, called 'Hubs'. The 'macro-level' mobility refers to the fact that our abstraction does not depend on the exact movement within a Hub, or in between Hubs. Rather, our abstraction only specifies a set of Hubs where a node will visit and spend some significant amount of time, without having to follow any rigid schedule or routes (i.e., partially deterministic). We also propose an effective routing scheme for MANET called the Sociological Orbit Aware Routing (SOAR) protocol to take advantage of the spatial/temporal locality of the mobile users (nodes) around these Hubs. The orbital movement pattern is not only general enough to be realistic, but is also specific enough to be useful. At the same time, it can be practically

implemented without a need for constant location updates (or tracking) and flooding. Extensive numerical results are presented to establish the simplicity and superiority of SOAR over other conventional protocols, such as DSR and LAR in terms of higher data throughput, lower control overhead, and shorter end-to-end delay. Thus, SOAR does not make any tradeoffs with respect to these metrics unlike in DSR and LAR.

The rest of the paper is outlined as follows. In Section II, we motivate our work by discussing the sociological movement pattern of MANET users, and illustrate the 'Random Orbit' model as an example. In Section III, we provide the details of the proposed Sociological Orbit Aware Routing (SOAR) protocol and illustrate its use in a common and realistic MANET scenario. In Section IV, we evaluate the performance of SOAR through simulations, and show that it is superior to DSR and LAR. In Section V, we present a more detailed description of other related work on both practical mobility models, and mobility sensitive routing protocols to highlight the novelty of our work. We conclude this work in Section VI.

II. Sociological Movement Pattern

In the real world, mobile users move with certain purposes in mind (e.g., going from a conference room to a cafeteria) in addition to being subjected to other restrictions (e.g., geographical constraints). One of the important implications of the above observation is that 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 attending a technical convention may visit and spend some significant amount of time in different rooms hosting say Conference Track 1, Workshop 2, Tutorials 2, Cafeteria, etc. on any given day. Although it is hard to keep track of an individual at all times, and may even be against some personal privacy policies to track him continuously, from a high level perspective, most user's movement are within and in between a list of Hubs. These Hubs may be visited by the individual in some probabilistic sequence, and constitute a part of the user's partially deterministic *mobility profile*. Even if we do not know the exact location of the graduate student at any given time, given his/her mobility profile we can identify a list of possible places (e.g. workshop 2, cafeteria) for locating him/her.

This orbital movement pattern is also observed in a time and space based hierarchy. For example, on a typical weekday, the graduate student could leave home for school in the morning, visit the gymnasium in the evening, and return home at night. Similarly, the student may stay in his home town for a few weeks and visit friends and family in other cities over some weekend, forming yet another higher level nation-wide 'orbit'. This hierarchical concept is illustrated in Figure 1.

A. Natural Orbits

Interestingly, an orbit is one of the most natural forms of motion observed in the microscopic world of molecules, as well as in the planetary universe. Although, such natural orbits are mostly deterministic, and their continuous motion does not have the notion of special places like Hubs.

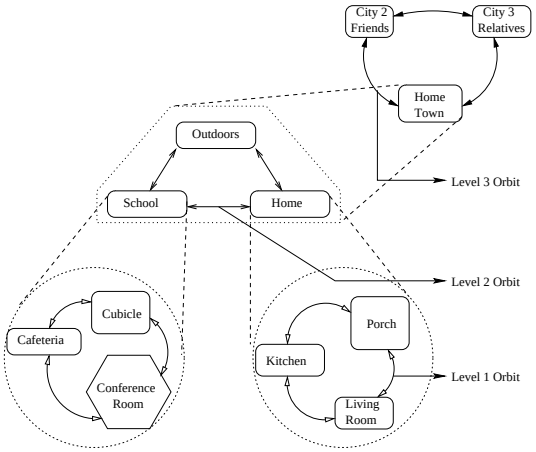


Fig. 1. A hierarchical view of sociological orbits

Electron Mobility: An electron movement is primarily an orbital motion around a nucleus. Whenever the number of electrons orbiting a particular nucleus changes, the atom (containing the electrons and the nucleus) becomes reactive and exchange electrons with other reactive atoms, thereby attaining stability. Thus for a particular electron, its a series of orbits around different nuclei over time. In society, job opportunities and inexpensive accommodation may cause an influx of people into a city. This may slowly saturate the place, leading to scarcity of jobs, high cost of living, etc. which in turn pushes people to other cities with better opportunities. In this way, some people also exhibit a series of orbital movements around different cities over time, as society tries to maintain some stability across its city based social nuclei.

Planetary Motion: All planets along with their satellites display a time and space based hierarchical orbital model, which is as follows. The moon revolves around the earth in a small orbit, lasting a month. The earth revolves around the sun in a larger orbit, lasting a year. The sun itself revolves around the milky way in a huge orbit of its own, lasting around 226 million years (a cosmic year). Although the mobility profile of mobile users do not exhibit a pattern as deterministic as the planetary system, it does possess a similar time and space based hierarchy (as shown in Figure 1).

Note that our orbital movement pattern differs from existing mobility patterns studied in the literature, 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 ([19], [20], [21]). It also differs from the deterministic mobility patterns assumed within DTN, where either exact locations of a node can be predicted with an appropriate 'oracle', or no location information is available. To the best of our knowledge, no prior work has explored the implication of such a macro-level partially deterministic sociological mobility pattern and its application to routing in MANET, despite its practicality.

B. A Random Orbit Model

To facilitate our discussion of Sociological Orbit Aware Routing (SOAR) within MANET, we first construct a simple

yet practical orbital model called the Random Orbit. As mentioned in Section II, sociological movement pattern associates an individual node's movement with several special regions, or Hubs within a terrain that the individual node visits. The Random 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 *Total Hubs*. These Hubs are located at random places within the terrain, and as a result they may or may not overlap with each other. Each node can visit a subset of randomly chosen Hubs creating a *Random Orbit*. The list of Hubs a node visits is bounded by *Hub List Size*, and the time it spends in each Hub is specified by *Hub Stay Time*. Together, these two parameters define an Inter-Hub Orbit (IHO). We also allow for an occasional change in the specific list of Hubs assigned to a node in its IHO by defining an *IHO Timeout*, upon which a node is assigned a fresh list of Hubs to visit.

Without loss of generality, each Hub is considered to be a rectangular, with the length on each side bounded by *Hub Size*. The mobility pattern of individual nodes shall comprise of two parts: movement inside a Hub, and movement in between Hubs. The point to be noted here is that for each of the two parts, any known practical mobility models may be chosen, as suggested in Table I. For detailed reference to these models, please see Section V.

TABLE I
EXAMPLE MOBILITY MODELS FOR ORBIT FRAMEWORK

Inside Hub	*Between* Hubs	Reference
LMM	GMM	[9]
Area Zone	City Area	[10]
METMOD	NATMOD	[11]
Manhattan	Freeway	[1]
Random Waypoint	P2P Linear	[2]

In our work, the movement inside each Hub, which shall also be referred to as the Intra-Hub Movement (IHM), was chosen to follow the Random Waypoint mobility model, whose speed range is denoted by *Intra-Hub Speed*, and whose pause time is denoted by *Intra-Hub Pause*. Note that a non-zero minimum for the Hub Speed should be chosen to overcome the decaying average speed problem associated with Random Waypoint, as suggested in [23]. 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 randomly selects a point within the destination Hub and moves towards it linearly from its current position with a velocity defined by the range *Inter-Hub Speed*. While Figure 2, illustrates the Random Orbit model as described above, Table II summarizes all of the Random Orbit parameters mentioned. Note that, this example Random Orbit model does not simply integrate two common mobility models (Random Waypoint, and P2P Linear) into our hierarchical orbital framework, but most importantly also introduces the practical orbital movement amongst Hubs.

The above Random Orbit model is suitable for modeling wireless devices carried by users working in an office building, attending a convention, or around a campus. As users move around, devices either automatically, or with the user's permission/assistance may record the Hubs visited most often, and share the Hub-based orbital mobility profile with trusted 'acquaintances'. Such mobility profile can then help improve routing as described next.

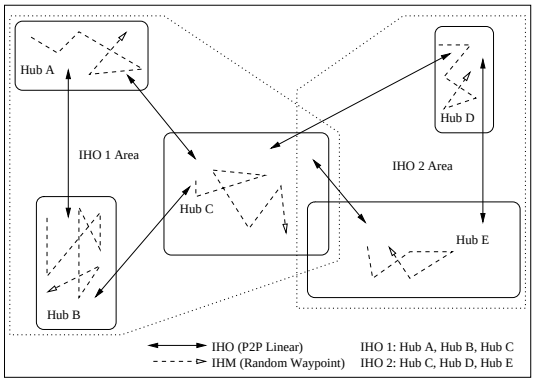


Fig. 2. The Random ORBIT Model

TABLE II
ORBIT PARAMETERS

Category	Parameter
General	Total Hubs
	Hub List Size (min, max)
	Hub Size (min, max)
	Hub Stay Time (min, max)
	IHO Timeout (min, max)
Inter-Hub	Inter-Hub Speed (min, max)
Intra-Hub (Rand Waypt)	Intra-Hub Pause
	Intra-Hub Speed (min, max)

III. SOCIOLOGICAL ORBIT AWARE ROUTING (SOAR)

In this section, we describe our Sociological Orbit Aware Routing (SOAR) protocol, which to the best of our knowledge, is among the first to make use of macro-level sociological mobility profiles of MANET users in obtaining approximate location information of mobile users as well as in improving routing.

A. Protocol Overview

Several motivations for *peer collaboration* (among 'acquaintances') were discussed by the authors in [24]. In [25], we proposed an Acquaintance Based Soft Location Management (ABSoLoM) technique, and showed the advantages of using 'acquaintances' to form a distributed location database. Although one of the basic concepts in SOAR is also the use of acquaintances, it differs from ABSoLoM in at least 2 significant ways. First, nodes in ABSoLoM use a formal acquaintanceship request and response among a few selected nodes that ensure the acquaintanceship to be mutual. Second, acquaintances not only cached each other's exact location coordinates, but also kept each other informed of their current coordinates through frequent location updates. In SOAR

however, acquaintanceship need not be mutual. As soon as one node gets to know of another node's Hub list, it will treat that other node as an 'acquaintance' and cache its Hub list information. This knowledge may be gained either by exchanging Hub list information with that node directly when they are within radio range of each other, or through a trusted third party (e.g., common acquaintance). Also, since the orbital mobility profile or, the Hub list information stays valid for a much longer time when compared to the exact location coordinates, SOAR can significantly reduce overhead in terms of *location updates* in the face of node mobility. More specifically, in SOAR, each node only needs to know the terrain in terms of the Hubs (i.e., their coordinates) in addition to its own location. Each node periodically sends its own coordinates and Hub list in a *Hello* message to its immediate neighbors within radio range to facilitate both neighbor discovery (required for Inter-Hub geographic routing of packets) and Hub list sharing (for formation of new acquaintances). Only acquaintances with an active data connection in between them need to notify each other by 'location updates' when there is a change in any of their Hub lists (as a result of an occasional IHO Timeout). A more detailed description of packet routing in SOAR is as follows.

B. Information Query Propagation and Response

In SOAR, all packets (*query*, *response*, *data*, *update*) are sent from one node (e.g., source) towards the Hub list of another node (e.g., destination) that is contained in the packet header. When a source has *data* to send and the destination is a neighbor within radio range, the *data* is directly transmitted. However, if the destination is not a neighbor, but an acquaintance whose Hub list is cached by the source, the *data* packet is forwarded to the (center point of) Hub(s) in that Hub list (see Section III-C for more details). Routing from the source to a Hub is accomplished by 'greedy geographic forwarding' [16], where each intermediate node chooses its next hop from amongst its neighbors who is closest to the destination than itself (see Section III-D for more details).

If no information about the destination's Hub list is available, the source first selects a subset of its acquaintances in a way to be described in Section III-E. For each chosen acquaintance, a separate *query* is sent to the Hubs in that acquaintance's Hub list. Such a transmission from a node to its acquaintance will be referred to as a *logical hop* here after, which often comprises of multiple physical hops. An acquaintance responds to this *query* packet if it knows of a valid Hub list of the destination. As an optimization, intermediate nodes (that are not acquaintances of the source) are also allowed to snoop into *query* packets and respond to them if possible.

If the acquaintance does not know the destination's Hub list, it forwards the *query* to a subset of its own acquaintances, chosen appropriately as before. However, if the number of logical hops exceeds a specified threshold, the *query* packet is dropped by the acquaintance. If all the *query* packets are similarly dropped, the source will time out and may either drop the *data* packet, or retry sending new *query* packets to a different subset of its acquaintances, or resort to simple flooding of *query* packets.

If the *query* reaches the destination itself, it not only responds with its own Hub list, but also indicates the current Hub it is in. The current Hub information is cached by the source and used for subsequent delivery of data in the same session. The cache timeout value for this current Hub information will be based on the average Hub Stay Time of a node. Similarly, the Hub list information itself will be cached at the source for a time proportional to the average IHO Timeout.

C. Packet Transmission to a Hub List

Once the source of any packet (*query, response, data, update*) knows the Hub list information for that packet's destination, it first checks to see if the current Hub information for that destination is available. If that information is unavailable, one copy of the packet is geographically forwarded towards (the center of) each of the Hubs (i.e., simulcast) in the Hub list of the destination. However, if the information is available, a single packet is geographically forwarded to (the center of) that current Hub. In either case, the source inserts its own Hub list and current Hub information into the packet header, that helps the destination of that packet to respond back to the source.

Specially for *data* packets, when the first *data* packet is sent, a 'data session' is considered active, which expires when no data is generated/sent within a specified interval. Throughout that active data session, the source keeps inserting his current Hub and Hub list information into each *data* packet to keep its location information updated at the destination. The destination of that active data session reciprocates with its current Hub and Hub list information in an *update* packet (which can double as an ACK) on getting the first *data* packet. From then on, whenever the destination moves out of its current Hub, or starts to orbit a different Hub list (on an IHO Timeout), it notifies the source of the change by sending a *update* packet towards the current Hub of the source. Since such *update* packets are restricted between the two ends of an active session only, they are sent out infrequently and incur little overhead.

Note that sending to the current Hub is just an optimization attempt. If the destination is not in that current Hub when the *data* arrives, the *data* can be cached by nodes in that Hub for a limited amount of time. This will allow the destination to retrieve it later, if it visits that Hub as part of its orbital movement. Just before the cached data is to be purged, the node that is closest to the center of the Hub may simulcast copies of that data to the other Hubs in the list of the destination. Of course, the source may also time out and decide to take an appropriate action (as discussed above).

D. Use of Geographic forwarding for Packet Delivery

When the source of any packet wishes to send that packet to a Hub (possibly containing the destination of that packet), it uses greedy geographic forwarding as mentioned before. As each intermediate node performs greedy geographic forwarding to push the packet closer to the intended Hub's center

coordinates, if a local maxima (also called a 'geographic hole') occurs (i.e., no neighbor closer to the Hub than itself), this node broadcasts the packet to all its neighbors. A neighbor in turn checks if any of its neighbors is closer to the Hub than the intermediate node which started this broadcast. If it finds any such neighbor, it forwards the packet to it. Otherwise, it may either drop the data packet, or employ techniques to route around 'geographic holes' as suggested in literature [17].

If the *Hub Size* is fairly large compared to the *Radio Transmission Range* of the nodes, once any packet reaches (i.e. enters) a Hub before reaching the destination, an *Intra-Hub flooding* (of the packet by the nodes within the Hub) is performed (as the exact coordinates of the destination may not be available). This is also done if the source itself lies in one of the Hubs in the destination's Hub list, in which case the source itself initiates the Intra-Hub flooding in an attempt to reach the destination. Such Intra-Hub flooding is not required when the Hub Size is comparable (or smaller) to the Radio Range since a packet can be overheard by all the nodes in the Hub as it is geographically forwarded to the center of the Hub.

In addition, an Intra-Hub flooding (if required) will introduce marginal overhead since a packet will only require to be flooded across a couple of radio hops to effectively cover the entire Hub. To support such limited flooding, all packets are uniquely identified by a tuple *(source, destination, sequence id)*, which enables nodes to identify and drop duplicate packets.

E. Querying a Subset of Acquaintances

A node may make a lot of acquaintances over its life time. Hence, to reduce the overhead due to the *query/response* packets, it needs to minimize the number of acquaintances it will query at any given time. On the other hand, since each acquaintance A_i covers (i.e., visits) a list of Hubs H_i, this minimum subset of acquaintances need to be carefully chosen to maximize the coverage of Hubs, thereby increasing the chances of obtaining the destination's information.

Let the Hub list of an acquaintance A_i be denoted by $H_i = \{h_1, h_2, ..., h_m\}$, where each h_i is a particular Hub. Let H be the set of Hub lists $\{H_1, H_2, ..., H_n\}$ covered by a node's acquaintances $A_1, A_2, ..., A_n$. Let C be the set of Hubs covered by all its acquaintances. That is, $C = H_1 \bigcup H_2 \bigcup ... \bigcup H_n$. Our problem is to find a minimum subset, $H' \subseteq H \quad s.t.$:

$$\forall h_i \in C, \quad \exists H_j \in H', \quad s.t. \quad h_i \in H_j$$

This is a minimum Set Cover problem, which is known to be NP Complete [26]. To find an heuristic solution, we have adopted the Quine-McCluskey technique [27], [28] used widely in Boolean Algebra for minimization of boolean expressions. To describe this method, we first define a few terms as follows.

Prime Acquaintance: An acquaintance A_i with Hub list H_i is a *Prime* acquaintance if there is no other single acquaintance A_j whose Hub list H_j includes H_i (i.e., $\nexists A_j, s.t. H_j \supseteq H_i$). Formally, A_i (with H_i) is a *Prime* acquaintance *iff*:

$$\nexists A_j (with H_j), \ s.t. \ \forall h_k \in H_i, \ h_k \in H_i \Rightarrow h_k \in H_j$$

For example, let $H_1 = \{1, 2\}, \ H_2 = \{2, 3, 4\}, \ H_3 = \{1, 4\}$, and $H_4 = \{3, 4\}$, be the Hub lists of acquaintances A_1, A_2, A_3, and A_4. Since none of A_2, A_3 or A_4 alone covers all the Hubs of A_1, A_1 is a *Prime* acquaintance. Following the same principle, both A_2 and A_3 are also *Prime* acquaintances, whereas A_4 is not (since $H_2 \supseteq H_4$).

Essential Prime Acquaintance: This is a *Prime* acquaintance that covers at least one Hub that is not covered by any other *Prime* acquaintance. Let $P = \{H_{p_1}, H_{p_2}, ...\}$ be the set of all the Hub lists of *Prime* acquaintances $\{A_{p_1}, A_{p_2}, ...\}$. Then, a *Prime* acquaintance A_{p_i} with Hub list H_{p_i} would be an *Essential Prime* acquaintance *iff*:

$$\exists h_k \in H_{p_i}, \quad s.t. \ \forall H_{p_j} \in P \ (j \neq i), \ h_k \notin H_{p_j}$$

Continuing with the previous example, even though A_1 is a *Prime* acquaintance, it does not cover any Hub that is not already covered by either A_2 or A_3. So A_1 is not an *Essential Prime* acquaintance. Following the same principle, A_3 is not an *Essential Prime* acquaintance either. However, A_2 covers Hub 3 that is not covered by any other *Prime* acquaintance (i.e., A_1 and A_3). Although, A_4 covered Hub 3, A_4 is not a *Prime* acquaintance, and hence ignored. Thus, A_2 is the only *Essential Prime* acquaintance in our example.

To query the optimal subset of acquaintances, a node first examines the Hub lists of its acquaintances and determines its *Prime* and *Essential Prime* acquaintances. All the *Essential Prime* acquaintances are then chosen, and all the Hubs in C that they cover are marked. If any Hub in C is left unmarked, a non-essential *Prime* acquaintance covering the maximum number of unmarked Hubs is chosen next, and the corresponding Hubs are marked. This procedure is repeated by adding one more non-essential *Prime* acquaintance at a time, until all the Hubs in C get marked. In the above example, first A_2 will get chosen (being an *Essential Prime* acquaintance), following which any one of the other *Prime* acquaintances (A_1 or A_3) will be chosen to cover Hub 1, that is not covered by A_2. Moreover, to minimize the number of responses generated for a particular query, the source may 'anycast' (send to any one of a list of destinations) query packets to Hubs that are common to the list of multiple acquaintances. Thus, in our example if eventually A_1 and A_2 get selected, separate query packets will go for A_1 to Hub 1 and for A_2 to Hubs 3 and 4, but a single 'anycast' packet destined for any of A_1 or A_2 will be sent to the common Hub 2. In addition to reduced responses, this will also minimize the number of query packets generated, leading to lower control overhead.

F. An Example Scenario

Continuing our previous example involving mobile users attending a large technical symposium, let us assume that 3 graduate students are at the same conference. The mobile wireless devices carried by them along with those carried by other convention attendees form the MANET shown in

Figure 3. The different rooms shown in the figure are assumed to hold different conference tracks, a poster session, and an exhibition area that are held concurrently. In addition, there is a registration area, a lounge and a cafeteria. Suppose Student 1 frequents the rooms hosting the *Conference Track 1, Conference Track 3*, and *Posters* (which form Student 1' IHO), while Student 2 frequents the *Lounge* and the room for *Conference Track 4* (which form Student 2's IHO), and Student 3 frequents the rooms for *Exhibits* and *Conference Track 4* (which form Student 3's IHO). When Student 1 is in the *Posters* section and Student 2 is in the *Lounge*, (note that these 2 Hubs/Rooms overlap), they came within each other's radio range and share their own Hub lists. Later, say Students 2 and 3 meet at *Conference Track 4* and also share their own Hub lists. Later, if Student 3 wishes to locate Student 1 (whom he/she has not met yet), he/she can query his/her acquaintance (Student 2) for information related to Student 1's possible locations (as shown in Figure 3(a)). Once Student 3 learns of Student 1's Hub list, he/she can then simulcast messages geographically towards the Hub list of Student 1 (as seen in Figure 3(b)).

IV. Performance Analysis

In this section, we describe our extensive simulation study to compare the performance of the SOAR protocol with that of DSR and LAR scheme 1 (LAR1) using GloMoSim [29]. We implement two versions of the SOAR protocol, SOAR-1 and SOAR-2. In SOAR-1, a node sends *Hello* packets containing its own Hub list to its 1-hop neighbors (i.e., nodes within its radio range), and only caches the Hub lists of those neighbors. In SOAR-2, each *Hello* packet also contains the Hub lists of the 1-hop neighbors in addition to the node's own Hub list. This allows nodes to cache the Hub lists of the nodes that are either 1 or 2 radio hops away. In both versions of SOAR, we use 2 as the threshold value for the number of *logical hops* any *query* packet may take before it is dropped. In this way, the query packets will only be sent to source's acquaintances, and their acquaintances. For comparison, we borrow the DSR and LAR1 implementations already available in the GloMoSim distribution.

For the simulation scenario, we consider a MANET 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 MANET, we considered the work in [30], [31], [32]. We summarize real life speed values for various activities in Table III and fix the Orbit Inter-Hub and Intra-Hub speed parameters accordingly. 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.

Relative Control Overhead: This metric is defined as the amount of control information (measured in bytes) that each node sends for each successfully received data packet in the network. For both LAR and DSR, we consider the *Route Request, Route Reply,* and *Route Error* packets as the control packets. In SOAR, the control packets are *Hello, Hub List Query, Hub List Response,* and *Location Update* packets. Although, in SOAR, the control packets have larger size (in bytes) due to the mobility information contained in them, we show via simulations that both the overhead and delay are lower than those in DSR and LAR.

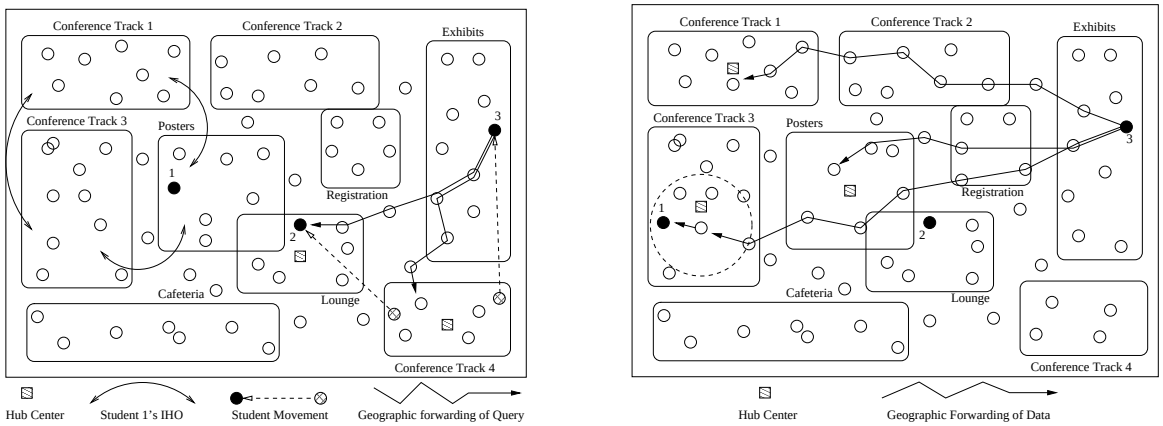
TABLE III
Real Life Speed

Category	Type	Range
Walking	Average	$= 1.34\ m/s$
	Olympic Record	$\approx 4.02\ m/s$
Running	Average	$= 4.00\ m/s$
	Olympic Record	$\approx 10.00\ m/s$
Cycling	Average	$= 8.94\ m/s$
	Olympic Record	$\approx 13.89\ m/s$

Approximation Factor for End-to-End Delay: The end-to-end delay measures the time from when a data packet is generated at the source, to the time when it is correctly received by the destination. Thus, this delay incorporates the time taken to discover a path to the destination (in DSR and LAR), or the time taken to discover the destination's Hub list (in SOAR). Packets not delivered by any protocol are excluded from the calculation for that protocol, which may raise 'fairness' concerns as to be discussed later. To account for this, we calculate the ratio of the delay observed in simulation and the minimum possible delay for a data packet in an ideal case, and call it the approximation factor for end-to-end delay. The latter is the time taken by a packet, right after being generated, to make its way to the destination via minimum number of radio hops without any MAC contention, network queuing delay, etc. This minimum number of radio hops is in turn obtained by dividing the straight line distance between the source and the destination by the radio range. We use the same minimum ideal delay per radio hop while calculating the approximation factor for each of the three protocols.

The main reason for dividing the observed delay of each packet by the ideal delay of that packet is to account for the fact that different protocols can successfully deliver different data packets, and thus may incur different end-to-end delays. If a protocol (e.g., DSR) only delivers packets to nearby destinations and drops packets to far away ones, the average end-to-end delay per received packet would inevitably be lower than that in a protocol (e.g., SOAR), which delivers to destinations both near and far. The approximation factor measures the end-to-end delay relative to the 'optimum' delay and thus introduces a sense of 'fairness' in the delay performance comparisons.

Table IV lists the major parameters used in the simulations. In what follows, we will examine how different parameters such as Total number of Hubs (given a fixed terrain), Hub Size, Inter-Hub Speed, Radio Range, and the total Number of Nodes affect the protocol performance. To that end, we vary one of these five factors while fixing all others parameters.



(a) Student 3 queries acquaintance (Student 2) for Student 1 (b) Student 3 sends data to Student 1's Hub list

Fig. 3. Use of Sociological Mobility Profiles of MANET Users for Location Approximation and Routing

TABLE IV
SIMULATION PARAMETERS

GENERAL PARAMETERS			
Simulation Duration (each run)	1000s	Terrain Size	1000m x 1000m
Number of Nodes (*Users*)	*Vary*, (Default= 100)	Radio Range	*Vary*, (Default= 250m)
MAC Protocol	IEEE 802.11	Mobility Model	Random Orbit (RW + P2P)
ORBIT PARAMETERS			
Total Hubs (*Rooms*)	*Vary*, (Default= 15)	Hub Size	*Vary*, (Default= 200m-300m)
Hub Stay Time	50s-100s	IHO Timeout	250s - 500s
Hub List Size	2 to Total Hubs	Inter-Hub Speed	*Vary*, (Default= 10m/s-30m/s)
Intra-Hub Pause	1s	Intra-Hub Speed	1m/s-10m/s
TRAFFIC PARAMETERS			
CBR connections	200 (5 packets each) Random	Data Payload	512 bytes per packet

A. Variation in Total 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, and the Hub list sizes of each node, thereby affecting the protocol performance as described below.

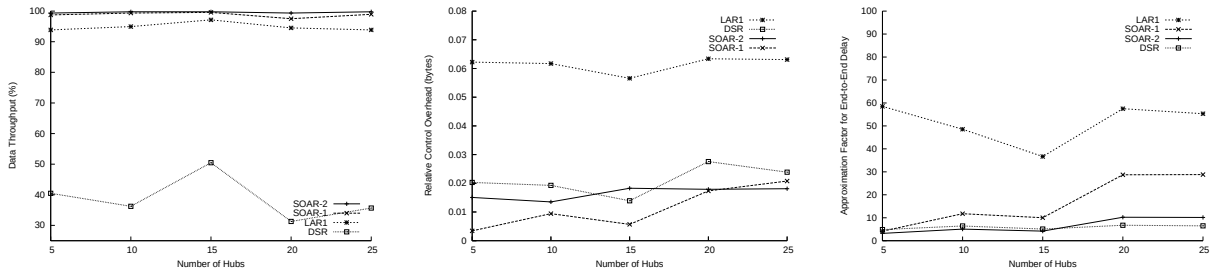
Data Throughput: Figure 4(a) shows the data throughput of all the protocols with varying number of Hubs. SOAR-2 and SOAR-1 perform the best with LAR1 showing comparable results. DSR has the lowest values for this metric.

The number of Hubs seems to have little impact on SOAR-2, SOAR-1 and LAR1 but has an interesting impact on DSR. With a very few Hubs, the number of nodes that happen to stay within each Hub at any given time can be very large. This elevates the *broadcast storm* problem (increased MAC layer contention) in DSR when flooding of discovery packets is attempted by any node, leading to unsuccessful route discovery and poor throughput. The performance of DSR improves with the number of Hubs, but after a point, it deteriorates once again. This is because the Hub list sizes of nodes also increases with the number of Hubs, and as a result, the nodes enjoy greater freedom of movement within the terrain, adversely affecting DSR by increasing the chances of route failures.

LAR1 employs the caching of velocity and location information that helps in limiting the amount of flooding required, thereby resulting in much better performance. In the SOAR protocols, as long as there is Inter-Hub movement whence the Hub list information is shared amongst nodes, there is sufficient means to locate nodes and route packets to them, irrespective of the number of Hubs.

Relative Control Overhead: From Figure 4(b), we note that LAR1 has the highest overhead, followed by SOAR-1, DSR and SOAR-2 respectively. The majority of the overhead in flooding based protocols such as LAR1 and DSR is due to the route discovery process. Specifically, in LAR1, routes are discovered iteratively by increasing the size of the region where a destination is expected to be found. When the number of Hubs is very low, they may be located far apart, requiring nodes to travel long distances as part of their IHO. This leads to nodes moving out of LAR1's estimated region, causing repeated flooding and consequently increases the control overhead. On the other hand, if the number of Hubs (and the Hub list size along with it) is very large, nodes enjoy greater freedom of movement within the terrain. This too is not favorable for LAR1 for a similar reason as above. This is why a moderate number of Hubs seems to result in a lower control overhead in LAR1.

DSR adopts a less aggressive flooding scheme and is shown to have a lower overhead than LAR1. In the case of SOAR protocols, the periodic HELLO beacon in SOAR-2 contains



(a) (b) (c)

Fig. 4. (a) Data Throughput (b) Relative Control Overhead (c) Approximation Factor for End-to-End Delay, vs. Total Hubs

more information than that of SOAR-1. Thus, the overhead in SOAR-2 is more than that of SOAR-1. More specifically, in SOAR, Hub lists stay valid for a longer time (than location coordinates, or routes), minimizing the number of query/response packets. In addition, the location update packets are also limited and infrequent. Thus, SOAR protocols are able to maintain the lowest overhead among its competitors.

Approximation Factor for End-to-End Delay: The reasons given above also explain the approximation factor for delay of all the protocols as seen in Figure 4(c). LAR1 has the highest delay due to its iterative estimation of node location, and increased control overhead. In SOAR-1, as the Hub list size grows with the number of Hubs, it takes a longer time to get the Hub list of a destination with the assistance of only 1-hop neighbor information. Thus, the delay in SOAR-1 increases marginally with increasing number of Hubs. SOAR-2, with more information, does considerably better than SOAR-1, and is comparable to DSR. However, a point to note is that this delay in DSR is only averaged over the data packets it successfully received, which is far less than any other protocol. Overall, all protocols seem to perform the best with a moderate number of Hubs for the default simulation terrain, Hub size, and number of nodes. Accordingly, we set the default value of the number of Hubs to 15 (see Table IV).

Note that the results on the relative performances of the three protocols shown in Figure 4 are generally applicable to all other four cases to be described below where the Hub size is fixed but one of the other four parameters varies, although the explanations may be slightly different in those four cases.

B. Variation in Hub Size

We study the effects of the Hub size on the protocol performance in this section. In the following simulations, the Hubs were considered to be square regions with equal sizes.

Data Throughput: Figure 5(a) shows that SOAR-2, SOAR-1, and LAR1 perform consistently well across all Hub sizes, with the SOAR protocols doing the best. On the other hand, since small Hubs force nodes to stay very close to one another within a Hub, DSR is adversely affected by the *broadcast storm* problem mentioned before, and hence does not perform well with small Hub sizes. On the other hand, Hub size has minimal effect on the throughput performance of SOAR and LAR.

Relative Control Overhead: Once again, LAR has the highest control overhead, followed by DSR, SOAR-2 and SOAR-1. The reasons are similar to those given in Section IV-A.

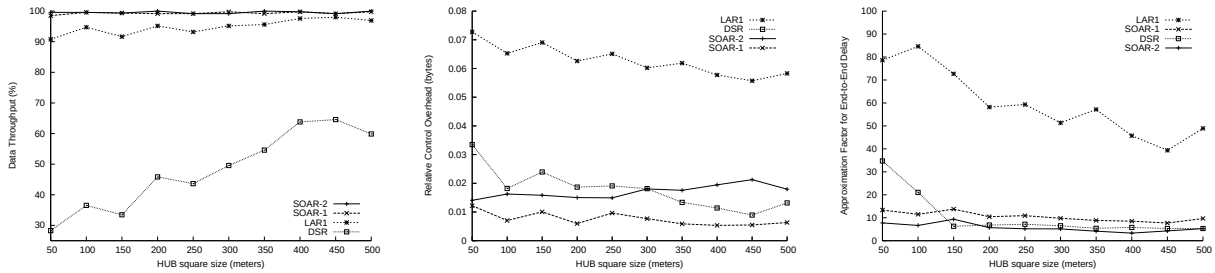
Approximation Factor for End-to-End Delay: Figure 5(c) shows LAR1 to have the highest approximation factor for the end-to-end delay, with DSR and SOAR-2 at a comparable minimum. When the Hubs are small, there is hardly any overlap amongst them. Thus, if a node moves out of a Hub, it most likely has to move a relatively long distance before reaching another Hub. This has a negative impact on the location estimation of LAR1. On other hand, when the Hubs are larger, there is a greater chance for Hubs to overlap. Thus, even if a node moves to a new Hub, its locality with respect to the terrain remains the same. This aids LAR1 in estimating node locations more accurately, and leads to a lower discovery delay with increasing Hub size.

DSR's delay can also be negatively affected by MAC layer contention with a very small Hub sizes, while SOAR protocols enjoy a low Hub list discovery latency as before, due to the use of the distributed location database within a network of acquaintances.

C. Variation in Inter-Hub Speed

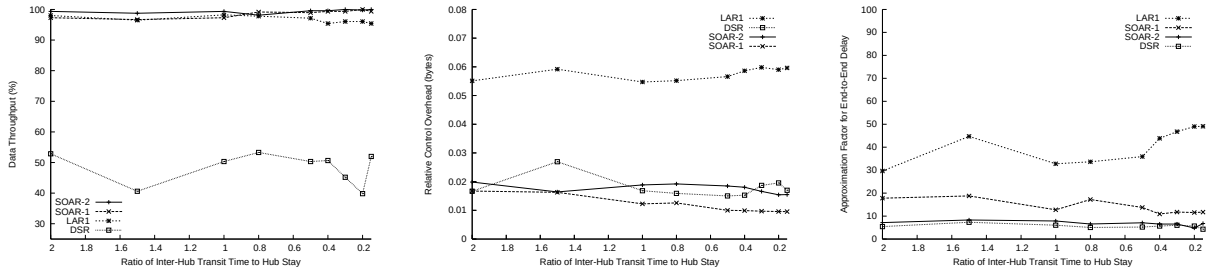
By varying the *Inter-Hub Speed* we varied the amount of time nodes spend on average transiting in between Hubs, with respect to their average Hub Stay Time. For the default *Inter-Hub Speed* range given in Table IV, the ratio of the Inter-Hub Transit Time to Hub Stay Time is in between 0.3 to 0.15. We varied this speed from 2m/s to 30m/s so as to the change the value of this ratio from 2 to 0.15.

Data Throughput: Figure 6(a), shows that SOAR-2, SOAR-1, and LAR1 do consistently well for the entire range, while DSR performance fluctuates at several points. LAR1 manages to maintain a high throughput only at the cost of higher overhead and higher delay as confirmed by our earlier observations. In the SOAR protocols, high values of the ratio (i.e. nodes spending a large amount of time traveling in between Hubs) does not have a significant impact on the throughput performance. This is because in SOAR, intermediate nodes also respond to queries, and cache data packets at each Hub in the destination's Hub list, in addition to being able to reach the destination outside a Hub during geographic forwarding. On the other hand, lower values of the ratio (i.e., nodes spend considerable amount of time within Hubs) only substantiates



(a) (b) (c)

Fig. 5. (a) Data Throughput (b) Relative Control Overhead (c) Approximation Factor for End-to-End Delay, vs. Hub Size



(a) (b) (c)

Fig. 6. (a) Data Throughput (b) Relative Control Overhead (c) Approximation Factor for End-to-End Delay, vs. Ratio of Inter-Hub Transit Time to Hub Stay

the practicality of the Hub list information. DSR seems to be doing relative well for three different cases. First, when nodes spend more time outside Hubs than inside, they have a low Inter-Hub speed, in addition to the default low Intra-Hub speed. This overall reduction in node velocity increases route stability in DSR, leading to good throughput. Second, when nodes spend most of their time within Hubs, they move with low (default) Intra-Hub speed leading once again to increased route stability. Third, DSR also seems to be doing well when nodes spend an equal amount of time inside and outside Hubs, which leads to a more uniform node distribution that ultimately increases the chances of route discovery via flooding.

Relative Control Overhead: The relative performances of LAR1, SOAR-2, and SOAR-1 in Figure 6(b) are similar to that observed in Figure 5(b), and for similar reasons. While LAR1 and SOAR protocols show more or less consistent performances, DSR performance directly reflects the impact of the three cases mentioned in the discussions of *Data Throughput* in this same section, on the control overhead.

Approximation Factor for End-to-End Delay: In terms of this metric, LAR1 has the highest values while both SOAR-2 and DSR show comparable minimum results in Figure 6(c). DSR however, achieves this average approximation factor over a much smaller set of successfully delivered data packets when compared to the other three protocols. SOAR-2 does better than SOAR-1 as expected due to a higher amount of Hub list caching. As node speed increases however, and they stay in Hubs more often and travel very quickly in between Hubs (i.e. lower ratio values), LAR may cache the lower Intra-Hub Speed and estimate a region around the last known Hub. Thus, anytime a node moves out of a Hub, LAR1 may fail to estimate the location correctly, thereby incurring higher delay with decreasing values of the ratio of Inter-Hub time to Hub stay time. This is also supported by marginal increase in LAR1 overhead, and marginal decrease in LAR1 throughput for smaller ratio values.

D. Variation in Radio Range (and Hub size)

The effect of a fixed radio range on varying Hub sizes has already been discussed in Section IV-B. In this section, we scale the terrain up by varying the Hub size and the radio range simultaneously, while retaining the default number of nodes and data connections.

Data Throughput: As seen in Figure 7(a), all protocols perform poorly with a smaller radio range. This can be explained as follows. In general, the average path length (in radio hops) between a source-destination pair increases with a smaller radio range. For LAR1 and DSR, the main impact of this effect is to increase the probability of a link failure, and ultimately leading to route failures. In the SOAR protocols, a reduced radio range implies a lower number of radio neighbors who can continue greedy forwarding. This in turn increases the probability of failure due to the occurrence of local maxima in greedy forwarding. With larger radio ranges, all protocols perform much better as expected.

Relative Control Overhead: Due to an increased average path length caused by a smaller radio range, flooding based protocols will incur higher overhead and delays. This is confirmed in Figure 7(b), which shows that both LAR1 and DSR have significant amount of control overhead for lower radio range values. However, in the SOAR protocols this effect is not as significant as in DSR or LAR1. For smaller radio ranges, nodes in SOAR have a lower number of neighbors that implies a lower number of acquaintances and lower protocol maintenance overhead on average. For higher radio ranges, nodes get to know of many other nodes' Hub lists, leading to

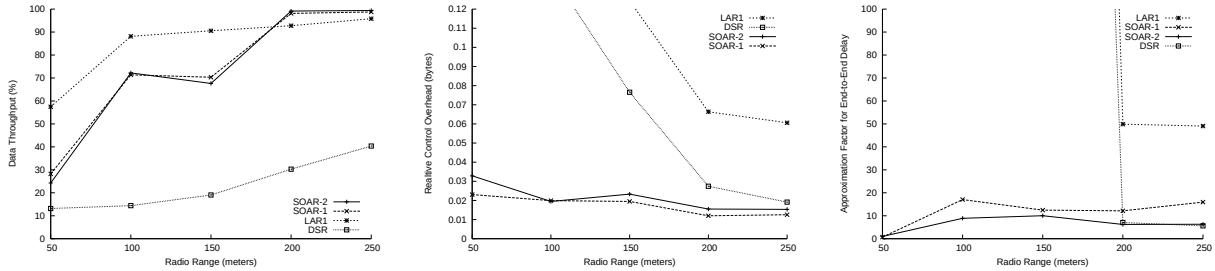


Fig. 7. (a) Data Throughput (b) Relative Control Overhead (c) Approximation Factor for End-to-End Delay, vs. Radio Range (Hub Size = Radio Range)

reduced overhead in terms of request/response packets. Thus, the overhead in SOAR is seen to be consistent across varying radio ranges.

Approximation Factor for End-to-End Delay: Figure 7(c) shows the delay for both LAR1 and DSR to be significantly higher for low radio range values. This is because longer paths in source routing schemes break more often, causing retransmissions resulting in higher delays. In the SOAR protocols, since nodes move in an orbit, they continue to share Hub lists with other nodes in Hubs and is unaffected by the varying radio range. Thus, the delay performance remains consistently lower with respect to DSR and LAR1.

An interesting point worth mentioning here is the relation of the node/route discovery latency in each protocol with the path length (in radio hops) between the source and the destination. In both LAR1 and DSR, the discovery latency for a previously unknown node may be directly proportional to the distance (in radio hops) between the source and the destination. Even if the caches within nodes are considered, longer routes have a higher probability of link breaks, leading to higher delay. However, in SOAR protocols, this relation is not that intuitive. For example, in SOAR-1, a source node may need to learn about a destination 2 hops away by querying an acquaintance that is say 4 hops away, thereby increasing the approximation factor for the end-to-end delay (i.e., the observed delay with respect to the ideal delay based on the distance between the source and the destination). On the other hand, it is equally likely that a source node may learn about a destination that is 4 hops away by simply querying a 1 hop neighbor, that happens to be an acquaintance of the destination. More specifically, since the Hub lists stay valid much longer than route caches, longer source-destination distances may still have end-to-end delays close to the ideal case due to reduced discovery latency. Thus, in SOAR the discovery latency is tightly coupled with the knowledge of each node about other nodes' Hub lists, and not as much dependent on the radio hop distance between the source and the destination as in LAR1 and DSR.

E. Variation in Number of Nodes (and Data Connections)

Finally, we study the effect of network load on our routing protocols by varying the number of nodes while keeping the number of connections per user constant, resulting in a varying total number of connections.

Data Throughput: With a small number of nodes (and connections), LAR1 performs the best as shown in Figure 8(a). In this case, DSR also benefits considerably and in fact, performs as well as SOAR protocols. As for the SOAR protocols, a very small number of nodes increases the chances of encountering a local maxima (or routing hole) while performing geographic forwarding. Additionally, with a fewer connections, SOAR protocols can no longer benefit much by allowing nodes to learn other node's Hub lists as they forward data packets for other nodes. Nonetheless, as the number of nodes (and data connections) increases beyond 40, SOAR achieves highest throughput while DSR begins to get increasingly affected by the *broadcast storm* problem as discussed earlier.

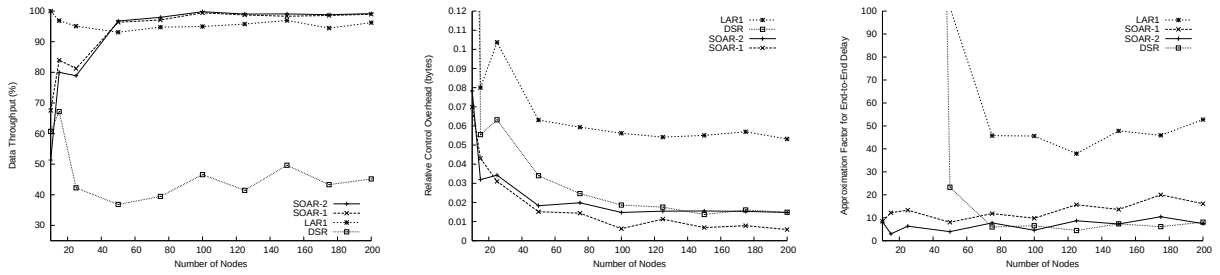
Relative Control Overhead: As shown in Figure 8(b), for all the protocols, the relative overhead reduces with increased number of nodes as they can make better use of the different information (path, location, velocity, Hub list) cached in the intermediate nodes. The relative performance of the three protocols remains unchanged.

Approximation Factor for End-to-End Delay: Figure 8(c) shows that both LAR1 and DSR have a significantly higher delays with a small number of nodes as in the case of having a small radio range (though not as bad). This is because flooding becomes ineffective when there is a only small number of nodes that are restricted to move and stay within fixed Hubs. On the other hand, in SOAR protocols, the orbital mobility information of the nodes is still effective enough to keep the node discovery latency to a consistently low value.

To summarize, based on the above study, we can firmly claim that while DSR and LAR make tradeoffs between throughput, control overhead and delay, SOAR is far superior to any one of these protocols in terms of higher data throughput, lower control overhead, and shorter end-to-end delay.

V. Comparison with Related Work

As discussed earlier in the introduction, the growing awareness of the significant effect of node mobility on protocol performance led to the parallel evolution of practical mobility models, and mobility pattern adapting routing schemes. The most popular Random Waypoint model was clearly not too realistic, and its limitations were critically analyzed by the authors in [23]. In [33], the authors suggested enhancements to the model itself, like the use of acceleration to smoothen changes in speed and direction. At the same time, others started looking at different practical aspects of realistic mobility. While the authors in [3] focused on the application of voronoi graphs to model mobility in face of obstacles, those



(a) (b) (c)

Fig. 8. (a) Data Throughput (b) Relative Control Overhead (c) Approximation Factor for End-to-End Delay, vs. Number of Nodes (Data Connections = 2 x Number of Nodes)

in [4], integrated *3* sub-models: perception, behavioral and movement, to simulate the mobility of each individual node as a close interaction of simple behavioral traits. To guarantee a steady state in node movement distributions, the authors in [5] used *renewal theory*, while authors in [34] introduced the concept of *stochastic correlation* in their VUM (variable user mobility) model for cellular systems.

Modeling of mobility was not however restricted to the individual nodes. The importance of group mobility was realized to suit several realistic scenarios like in military drills, disaster recovery, search party, etc. Accordingly, the authors in [6], [7] proposed a mobility model called Reference Point Group Mobility, where an existing group leader determines a group's collective movement, while other members move independently within a small speed and angle deviation from that of the leader. Later they extended the mobility vector model into a framework, softening changes in speed and direction. In [8], the authors surveyed several such *Entity based* (e.g., Boundless Area, Gauss-Markov) and *Group based* (e.g., Column, Nomadic, Pursue) mobility models for ad hoc networks. In [1], the authors formalized a framework for analyzing mobility models in terms of protocol independent metrics, and also proposed the *Manhattan* and *Freeway* models to suit city-wide traffic.

Literature also suggested work that observed mobility in different hierarchical levels. In [9], the authors suggested two hierarchical layers for a wireless ATM network: a deterministic Global Mobility Model (GMM) to describe inter-cell movements, and a stochastic Local Mobility Model (LMM) to describe intra-cell movements. In [10], the authors applied *transportation theory* to model: *City Area*, *Area Zone*, and *Street Unit*, at three hierarchical levels of detail. On the same note, the authors in [11] proposed the Metropolitan (*METMOD*), National (*NATMOD*) and International (*INTMOD*) mobility models to respectively suit movements within metropolitan areas, in between them and in between countries.

Despite this wide range of study on practical mobility, no prior work has admitted the realization of sociological impact on the mobility of MANET users within a society at a higher level of abstraction. Our contribution to this end is the observation of the close association of user (node) mobility with a pattern of visits to geographic regions of some social significance.

On the other hand, work was also done to aid routing protocols in countering the ill effects of node mobility in various ways. The initial flooding based source routing schemes (e.g., [2], [35]) turned to aggressive caching of routes to reduce the route discovery delay. The authors in [12] also cached the node velocity and were able to restrict the flooding required by computing an expected zone containing a node. With the advent of localization techniques and GPS technology, several location management schemes ([16], [36], [37], [38] coupled with geographic routing provided with effective routing solutions in the face of node mobility. However, such accurate knowledge of node locations comes at the high price of control overhead in terms of frequent location updates, and other protocol maintenance.

It was not until the awareness of mobility impact on protocol performance started growing, that researchers started studying the effects of various realistic scenarios on the existing MANET routing protocols (e.g., [39], [40]). Moving one step further, the authors in [13], [14], suggested methods like a connected virtual backbone to help routing protocols adapt to node mobility. Around the time that literature started suggesting practical mobility models, some researchers (e.g., [20], [21], [19] started to focus on mobility pattern/information awareness in routing protocols. Their motivation however, was the use of such mobility pattern identification, through continuous location tracking, in micro level mobility prediction that helps take low level routing decisions like the best choice of next hop in a path, link expiry prediction for QOS routing, etc.

Our work does not contend the efforts mentioned above. While they maintained their focus on the lower level of routing issues, we concentrated on the macro level mobility information that may be extracted from the observation of sociological movement pattern of MANET users within a social environment. Just as they used their micro level information for lower level mobility prediction, we use our information for a higher Hub level location management. The authors in [41] acknowledged the higher probability of a node visiting a location which it has frequented the most in the past. They developed a *delivery predictability* metric to perform probabilistic routing for intermittently connected networks like the Delay Tolerant Networks (DTN). However, unlike us, they did not attempt to profile the user mobility based on a collection of such regions that are visited by it most often in some periodic sequence. To the best of our knowledge, we are among the first to both imply the existence of such a

sociological mobility profile for a MANET user, and exploit it in making Hub or, macro level location management and routing decisions.

VI. Conclusion

In this work, we have exploited a higher level of mobility information abstraction. Specifically, we have observed the social influence on the macro-mobility of each MANET user and suggested an orbital movement pattern for each user based on a list of places or Hubs that they frequently visit. We have used this simple yet practical mobility information to perform intelligent routing. In particular, we have proposed a Sociological Orbit Aware Routing (SOAR) protocol for MANET and established the advantages of SOAR over conventional MANET routing protocols like LAR and DSR in terms of higher data throughput, lower control overhead, and lower end-to-end delay.

References

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

[2] J. Broch, D. B. Johnson, and D. A. Maltz, "Dynamic Source Routing in ad hoc wireless networks," *Mobile Computing, Kluwer Academic Publishers*, pp. 153–181, 1996.

[3] A. Jardosh, E. M. Belding-Royer, K. C. Almeroth, and S. Suri, "Towards Realistic Mobility Models for Mobile Ad hoc Networks," *Proceedings of ACM/IEEE MobiCom '03*, September 2003.

[4] D. Tan, S. Zhou, J. Ho, J. Mehta, and H. Tanabe, "Design and evaluation of an individually simulated mobility model in wireless ad hoc networks," *Communication Networks and Distributed Systems Modeling and Simulation Conference (CNDSMSC)*, San Antonio, TX 2002.

[5] G. Lin, G. Noubir, and R. Rajaraman, "Mobility Models for Ad hoc Network Simulation," *Proceedings of IEEE INFOCOM '04*, March 2004.

[6] X. Hong, M. Gerla, G. Pei, and C.-C. Chiang, "A Group Mobility Model for Ad hoc Wireless Networks," *Proceedings of ACM/IEEE MSWIM '99*, August 1999.

[7] X. Hong, T. Kwon, M. Gerla, D. Gu, and G. Pei, "A mobility framework for ad hoc wireless networks," *In ACM 2nd International Conference on Mobile Data Management (MDM)*, January 2001.

[8] T. Camp, J. Boleng, and V. Davies, "A Survey of Mobility Models for Ad Hoc Network Research," *Wireless Communications and Mobile Computing (WCMC): Special issue on Mobile Ad Hoc Networking: Research, Trends and Applications*, vol. 2, no. 5, pp. 483–502, 2002.

[9] T. Liu, P. Bahl, and I. Chlamtac, "Mobility modeling, location tracking, and trajectory prediction in wireless atm networks," *IEEE Journal on Selected Areas in Communications*, vol. 16, no. 6, pp. 922–936, August 1998.

[10] J. Markoulidakis, G. Lyberopoulos, D. Tsirkas, and E. Sykas, "Mobility Modeling in Third-Generation Mobile Telecommunications Systems," *IEEE Personal Communications*, vol. 4, no. 4, pp. 41–56, August 1997.

[11] D. Lam, D. Cox, and J. Widom, "Teletraffic Modeling for Personal Communications Services," *IEEE Communications Magazine*, pp. 79–87, February 1997.

[12] Y. B. Ko and N. H. Vaidya, "Location-Aided Routing (LAR) in mobile Ad-Hoc networks," *Proceedings of IEEE INFOCOM '98*, pp. 66–75, October 1998.

[13] S. Basagni, D. Turgut, and S. Das, "Mobility-adaptive protocols for managing large ad hoc networks," *Proceedings of IEEE International Conference on Communications (ICC)*, pp. 1539–1543, June 2001.

[14] S. Butenko, X. Cheng, D.-Z. Du, and P. Pardalos, "On the construction of virtual backbone for ad hoc wireless networks," *Cooperative Control: Models, Applications and Algorithms*, pp. 43–54, 2003.

[15] P. Enge and P. Misra, "Special Issue on Global Positioning System," *Proceedings of the IEEE*, vol. 87, no. 1, pp. 3–15, January 1999.

[16] J. Li, J. Janotti, D. S. J. D. Couto, D. R. Karger, and R. Morris, "A Scalable Location Service for Geographic Ad Hoc Routing," *Proceedings of ACM/IEEE MobiCom '00*, pp. 120–130, August 2000.

[17] B. Karp and H. T. Kung, "GPSR : Greedy perimeter stateless routing for wireless networks," *Proceedings of ACM/IEEE MobiCom '00*, pp. 243–254, August 2000.

[18] M. Grossglauser and D. N. C. Tse, "Mobility increases the capacity of ad hoc wireless networks," *IEEE/ACM Transactions on Networking*, vol. 10, no. 4, pp. 477–486, August 2002.

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

[20] W. Su, S.-J. Lee, and M. Gerla, "Mobility prediction and routing in ad hoc wireless networks," *International Journal of Network Management*, vol. 11, no. 1, pp. 3–30, February 2001.

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

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

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

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

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

[26] M. Garey and D. Johnson, "Computers and intractability: A guide to the theory of np-completeness," *W.H. Freeman and Company*, June 1979.

[27] W. V. Quine, "A way to simplify truth functions," *American Mathematical Monthly*, vol. 62, no. 9, pp. 627–631, 1955.

[28] E. J. McCluskey, "Minimization of boolean functions," *Bell Systems Technical Journal*, vol. 35, no. 5, pp. 1417–1444, 1956.

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

[30] R. L. Knoblauch, M. T. Pietrucha, and M. Nitzburg, "Field Studies of Pedestrian Walking Speed and Start-Up Time," *Transportation Research Board Records*, no. 1538, 1996.

[31] R. M. Queen, H.-Y. Lin, and M. T. Gross, "Standardized Running Speed versus Self-selected Running Speed: A Between-Trial and Between-Day Reliability Study," *The Center for Human Movement Science - UNC Chapel Hill*.

[32] A. Williams, "Cycling Speed," *Article in Peak Performance Online (http://www.pponline.co.uk/encyc/0065.htm)*, no. 65.

[33] C. Bettstetter, "Smooth is better than sharp: a random mobility model for simulation of wireless networks," *Proceedings of the 4th ACM international workshop on Modeling, analysis and simulation of wireless and mobile systems*, pp. 19–27, 2001.

[34] K. Lee and S. Kim, "Modeling variable user mobility with stochastic correlation concept," *Computer Networks: The International Journal of Computer and Telecommunications Networking*, vol. 38, no. 5, pp. 603–612, April 2002.

[35] C. E. Perkins and E. M. Royer, "Ad hoc On-Demand Distance Vector Routing," *Proceedings of the 2nd IEEE Workshop on Mobile Computing Systems and Applications*, pp. 90–100, February 1999.

[36] S.-C. Woo and S. Singh, "Scalable Routing Protocol for Ad Hoc Networks," *Wireless Networks*, vol. 7, pp. 513–529, January 2001.

[37] C. T. Cheng, H. L. Lemberg, S. J. Philip, E. van den Berg, and T. Zhang, "SLALoM: A Scalable Location Management Scheme for Large Mobile Ad-hoc Networks," *Proceedings of Wireless Communications and Networking Conference*, March 2002.

[38] S. Philip and C. Qiao, "Hierarchical grid location management for large wireless ad hoc networks," *ACM SIGMOBILE Mobile Computing and Communications Review (MC2R)*, to appear.

[39] G. Ravikiran and S. Singh, "Influence of mobility models on the performance of routing protocols in ad-hoc wireless networks," *Proceedings of IEEE Vehicular Technology Conference '04*, May 2004.

[40] P. Johansson, T. Larsson, N. Hedman, B. Mielczarek, and M. Degermark, "Scenario-based performance analysis of routing protocols for mobile ad-hoc networks," *Proceedings of the 5th annual ACM/IEEE International Conference on Mobile Computing and Networking*, pp. 195–206, 1999.

[41] A. Lindgren, A. Doria, and O. Schelen, "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.