Performance Analysis of Mobility Based Routing Protocols 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*— Most routing protocols in MANET adopt the popular Random Waypoint model for its simplicity and suitability for theoretical study and analysis. Recently, several entity, group and scenario based mobility models and frameworks have been proposed to model much more realistic and practical movements of mobile nodes in real scenarios. Although some work exists in evaluating routing protocols based on such specific scenarios, and some effort in adapting a protocol to suit mobility has been made, there does not exist any protocol that makes direct use of mobility information to route packets within a MANET. In this work, we first develop a practical mobility model that recognizes an orbital pattern in the sociological movement of mobile users, and then propose a novel Orbit Based Routing (OBR) protocol, that leverages the underlying orbital mobility to accurately determine a set of likely regions containing any node in the MANET. By forming a distributed location database among acquaintances and employing a scalable geographic routing to forward packets among nodes, OBR emerges as a clear choice for MANET routing in the face of practical mobility. We propose three different schemes of OBR and compare their performance against an Acquaintance Based Soft Location Management (ABSoLoM) protocol.**

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

I. INTRODUCTION

Routing in MANET is a challenging problem, and the task of locating a node and maintaining a path to it becomes increasingly difficult in the face of node mobility. Literature has proposed several routing protocols for MANET, but due to the adoption of Random Waypoint model in the performance study of these protocols, no useful assumptions about the underlying mobility were made in the protocol design. Within 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. However, common reactive protocols such as Dynamic Source Routing (DSR) [1] and Location Aided Routing (LAR) [2] either attempt to route based on cached paths and suffer terribly in the face of node mobility, or resort to repeated flooding thereby incurring a high control overhead.

To that end, position based routing has been suggested as an alternative to conventional routing approaches in mobile ad hoc networks in lieu of routing scalability. It is assumed that mobile nodes are aware of their own location via the use of a GPS receiver [3] or other localization schemes. A localized periodic broadcast protocol enables all nodes to have approximate knowledge of their neighbors' locations. Several proposals have been described in literature that make use of this neighborhood position knowledge to route packets to a known location of the destination.

One of the major attractions to position based routing is its localized nature of operation. While existing ad hoc routing protocols make use of source routes (e.g. DSR) or state based route construction/ maintenance (e.g. DSDV [4], AODV [5]), these routes are highly error prone due to node mobility or varying nature of the wireless channel. On the other hand, in position based routing, a mobile node only needs to know the destination's geographic location, location of neighbor nodes in its locality (radio range) and its own location in order to make a sensible routing decision. Since the position of a node's neighbors is conveyed through periodic broadcast messages, this information is readily available at each node. Thus, position based routing is a choice candidate for routing in mobile ad hoc networks.

However, a destination node's current location needs to be discovered before packets can be routed via position based routing. The problem of distributively managing locations such that the location of any node can be discovered prior to routing is known as the location management problem, and has been studied exhaustively by researchers. Several efficient solutions for this problem are available in literature. In one of our earlier work [6] we suggested the use of acquaintances to form a distributed location database that aids in a form of soft location management. We proposed a routing protocol called ABSoLoM that uniquely combined the power of distributed database with position based routing to outperform both conventional routing protocols (e.g. DSR, LAR) and grid based location management techniques (e.g. SLALoM [7]) in terms of control overhead and network topology assumptions (i.e. ABSoLoM does not need dense networks to perform well in contrast to most location management schemes).

In this study, we develop a practical mobility model and propose an improved protocol that takes advantage of the opportunities presented by such a model. To account for the inappropriateness of the Random Waypoint model in capturing realistic user mobility, literature has proposed several entity, group and scenario based mobility models. In a study of the sociological user movement, we have observed an orbital pattern that is not captured in any of the suggested mobility models or frameworks. Hence, we suggest our own *Random Orbit* mobility model that is not only practical, but is also capable of integrating other mobility models into a single domain. Most importantly, in this paper we propose an Orbit Based Routing (OBR) protocol that leverages the underlying orbital movement to determine a set of likely regions containing any node, thereby outperforming existing ad hoc routing protocols.

The rest of the paper is outlines as follows. In Section II we look at some related mobility models, frameworks and studies on the impact of mobility on routing protocols. In Section III we discuss the sociological user movement and introduce the Random Orbit model. In Section IV we describe the different schemes of the proposed OBR protocol. In Section V we compare the performance of the different OBR schemes against ABSoLoM, and in Section VI we conclude this work.

II. Related Work

Random Waypoint is the most popular *Entity based* mobility model in literature. In [8], the authors studied an average speed decay problem in Random Waypoint and in [9], the author enhanced the model by using acceleration to smoothen changes in speed and direction. To account for obstacles, the authors in [10] proposed a mobility model based on voronoi graphs. In [11], the authors integrated three sub-models: perception, behavioral and movement, to simulate the mobility of each node individually as a close interaction of simple behavioral traits. In [12], the authors used *renewal theory* to guarantee a steady state in node movement distributions, while those in [13] introduced *stochastic correlation* in their VUM (variable user mobility) model for cellular systems. However, all these models focus on the mobility in a flat network.

In [14], [15], the authors first proposed a *Group based* 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, smoothening changes in speed and direction. In [16], the authors surveyed several *Entity based* (e.g., Boundless Area, Gauss-Markov) and *Group based* (e.g., Column, Nomadic, Pursue) mobility models for ad hoc networks. In [17], the authors proposed a framework for analyzing mobility models in terms of protocol independent metrics. They also suggested the *Manhattan* and *Freeway* models to suit city traffic. These models can all be incorporated within the ORBIT framework at different levels to generate more realistic models.

In [18], the authors suggested two hierarchical layers for a wireless ATM network: a deterministic Global Mobility Model to describe inter-cell movements, and a stochastic Local Mobility Model to describe intra-cell movements. In [19], the authors applied *transportation theory* to model: *City Area*, *Area Zone*, and *Street Unit*, at three hierarchical levels of detail. Similarly, the authors in [20] 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. Although the proposed ORBIT hierarchy closely resembles these hierarchies, our main contribution lies in the recognition of the 'orbital' pattern that exists around these hierarchies.

The authors in [21] proposed a framework for graph based modeling of mobility and traffic in large scale MANETs, while in [22], the authors developed a tool for modeling scenarios like *Airport, Highway, and Conference* using visualized user interface. ORBIT differs from these frameworks in its generality, by which it can integrate such tools within its black boxes at different levels to generate more practical models for real life scenarios.

In [23], [24], the authors analyzed existing MANET routing protocols based on suggested mobility models and scenarios, but did not propose any specific protocol to suit them. In [25], [26], the setup of a connected virtual backbone was suggested within MANETs to help routing protocols adapt to node mobility, while in [27], the authors applied link expiration prediction based on neighbor velocity information to several existing routing protocols. On the same note, the authors in [28] used an adaptive algorithm to predict mobility to help location tracking of mobile nodes. However, no prior work has been done to design routing schemes that take direct advantage of the overall underlying mobility, like our proposed OBR protocol, which leverages the ORBIT mobility framework to outperform other position based routing protocols like ABSoLoM.

III. Sociological Movement based Random Orbit Model

In the real world, people live within societies, where their movement is subject to social constraints (e.g., following traffic regulations). Accordingly, although it is hard to determine the exact route taken by an individual at every turn, from a high level perspective, any person's movement exhibits a certain pattern that is repeated in some sequence. For example, an employee in an office may not always take the same path from his seat to a shared printer, but he is likely to repeat that movement a number of times during a day. Thus, even when we cannot determine the employee's location at a specific time, by studying his daily job routine, we can identify a list of possible places (e.g. cubicle, cafeteria) for locating him. In other words, there is an 'orbital' movement between these points of interest for this person.

This orbital movement pattern is also observed in a larger context. For example, on an average weekday, the employee could leave home for office in the morning, visit the gymnasium in the evening, and return home at night. Although we cannot predict the exact time or route taken by the person from one point to another on any given day, there is a number of fixed points of interest that are visited in some order, day after day, forming a high level 'orbit'. Similarly, the employee might 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'.

In short, the sociological movement pattern of humans is observed to be a collection of orbits at different levels of hierarchy, where each orbit comprises of a list of areas of interest and the movement in between them. Each such area along a high level orbit in turn contains a low level orbit consisting of a movement among a list of smaller areas of interest and so on, as illustrated in Figure 1. At each level of the hierarchy, the mobility along the orbits differs in terms of speed from one area to another, and the pause time in each area.

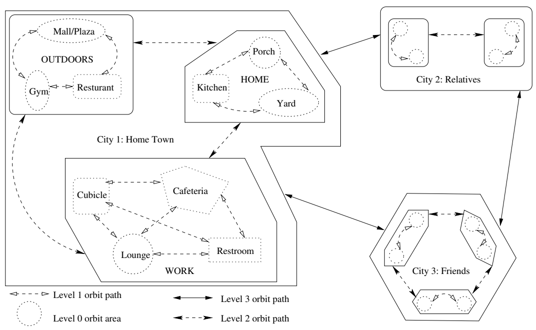

Fig. 1. The sociological orbit

Keeping the sociological orbit in mind, we consider a Random Orbit model with three hierarchical orbital levels. At the lowest level, we assume without loss of generality a rectangular area of interest, referred to as a *Hub*. For simplicity, we choose the Random Waypoint model within this area, but modify it slightly to fix the average speed decay problem by setting only non-zero minimum speed, as suggested in [8]. We refer to the movement inside the Hub as a *Local Area Orbit (LAO)*. For the next higher level, we consider a random selection process from a list of given Hubs. To move from one Hub to another, we choose to implement a simple model where a node picks a random point inside the new Hub and moves linearly towards it from its current location. We call this model as *P2P Linear*, and refer to this level of mobility as a *Medium Area Orbit (MAO)*. For the highest orbital level called the *Global Orbit (GO)*, we just consider a change in the list of Hubs given to the lower MAO. In this simplified framework, the MAOs may either overlap with a common Hub as shown in Figure 2, or may also remain disjoint as in Figure 1.

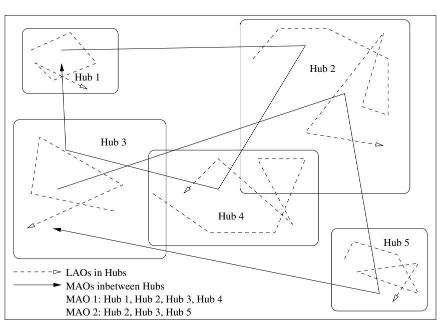

Fig. 2. Random Orbit: City Model

This model is useful for representing regular city traffic. Each Hub represents an office or residential area, where people move around in their sociological orbits. We observe pedestrian traffic inside Hubs, and faster vehicular traffic in between Hubs. The speed ranges for the LAO and the MAO are chosen according to real life speeds summarized in Table II. The parameters for this model are as described in Table I.

TABLE I
ORBIT PARAMETERS

Category	Parameter
Global Attributes	Total Hubs Hub Size (min, max) Hub Stay (min, max) Global Pause (min, max)
MAO Specific	Node Hubs (min, max) Node Speed (min, max)
LAO Specific	Hub Pause Hub Speed (min, max)

A Hub is assumed to be a rectangular area within the simulation terrain, with sides bounded by *Hub Size*. Initially, a specific number (bounded by *Node Hubs*) of Hubs is assigned to each node as part of its MAO. Nodes travel along their MAO from one Hub to another with speeds bounded by *Node Speed*. On reaching a Hub, a node moves according to the Random Waypoint model with speeds bounded by *Hub Speed* and pauses for *Hub Pause* amount. Each Hub requires a visiting node to stay for a time bounded by *Hub Stay*, which is also referred to as the *LAO Timeout*. When this timeout occurs, the node randomly selects another Hub from its list and moves towards it along its MAO, and initiates a fresh LAO upon reaching it. The MAO itself expires after a duration bounded by *Global Pause*, also referred to as the *MAO Timeout*, whence a fresh list of Hubs are assigned to the node to start a new MAO. Successive MAOs form the GO for the node. The actual speed limits in the LAO and the MAO will depend on the type of scenario being modeled.

IV. ORBIT BASED ROUTING (OBR) PROTOCOL

So far, we have described the realistic modeling capability of the proposed ORBIT framework. The orbital movement pattern also provides new opportunities to design efficient routing protocols. In this section, we describe our Orbit Based Routing (OBR) routing protocol that is among the first of its kind to the best of our knowledge, to make use of mobility information at the network layer.

Routing in MANET is a challenging problem, and the task of locating a node and maintaining a path to it becomes increasingly difficult in the face of node mobility. Literature has proposed several routing protocols for MANET, but due to the adoption of Random Waypoint model in the performance study of these protocols, no useful assumptions about the underlying mobility were made in the protocol design. In contrast, OBR tries to make use of the orbital mobility pattern in determining a set of likely regions containing a destination, as is described in detail below.

A. Protocol Overview

In continuation with our simplified analysis, we focus on a Hub level routing in the simplified ORBIT. Several motivations and advantages of *peer collaboration* were discussed by the authors in [29]. Accordingly, one of the basic concepts of OBR is to form a distributed location database among all nodes, where each node makes some acquaintances, and keeps track of their Hub lists within itself. This facilitates easy discovery of a destination with an unknown Hub list by a node via a network of its acquaintances, the acquaintances of each of its acquaintance, and so on. This concept is similar to that described in one of our earlier work [6], except that in OBR we take advantage of the underlying mobility information available through the ORBIT framework. This allows nodes to maintain Hub lists (that remain valid for a long time) of their acquaintances instead of their exact position, thereby reducing the overhead in *location updates* in the face of node mobility. Another key difference is that in ABSoLoM, nodes kept their acquaintances updated about their current location, whereas in OBR, these acquaintances are more like *logical neighbors*, where nodes do not update acquaintances when they change Hubs (within the same list) or when their MAO times out and their Hub list changes. Such updates are however performed only for active data sessions as explained later in Section IV-E. More specifically, it is assumed that each node has a specific knowledge of the terrain in terms of the Hubs and their corresponding coordinates. It is also assumed that the mobile nodes are aware of their own location via the use of a GPS receiver or other localization schemes. Each node periodically broadcasts its own coordinates and Hub list, and listens to the broadcasts made by other nodes, thereby learning of its neighbors. Each new neighbor becomes a new acquaintance and its corresponding Hub list is cached. Depending on the general value of the MAO Timeout observed in the scenario being modeled, an appropriate cache timeout value is chosen. The details of routing a packet in OBR is as follows.

B. Information Query Propagation and Response

When a source has *data* to send, it is directly transmitted to the destination if it is a neighbor. However, if it is not a neighbor, but an acquaintance with a valid Hub list in the source's cache, the *data* packet is forwarded towards that Hub list, as described in Section IV-D. If no information about the destination's Hub list is available, a *query* is sent out towards the Hub lists of a subset of acquaintances, chosen as described in Section IV-C. Such a transmission from a node to its acquaintance is referred to as a *logical hop*, which comprises of multiple physical hops determined by 'greedy geographic forwarding' [30], where each intermediate node chooses its next hop from amongst its neighbors who is closest to the destination's location than itself. An acquaintance responds to this *query* packet if it knows of a valid Hub list for the destination. If not, it forwards the *query* to a subset of its own acquaintances, carefully chosen as before. However, if the packet's logical hops exceed a specified threshold, it is dropped by the acquaintance instead of being forwarded to its own acquaintances. As an optimization, intermediate nodes are allowed to snoop into *query* packets and respond to them if possible. On receiving a *response*, the source caches the information and sends the *data* packet out towards the destination's Hub list.

C. Querying a Subset of Acquaintances

A node makes a lot of acquaintances over its life time. Hence, to reduce the control overhead it needs to limit the number of acquaintances it will *query* at any given time. However, a subset of its acquaintances has to be carefully chosen to cover all the Hubs it learned of from all its acquaintances.

Let H be a collection of subsets $\{H1, H2, ..., Hn\}$ of Hubs covered by the Hub lists of the acquaintances. Let C be the set of all the Hubs that a node learns of from all its acquaintances. Hence, $C = \bigcup\{H1, H2, ..., Hn\}$. Our problem is to find a minimum subset, $H' \subseteq H \;\; s.t.$:

$$\forall h \in C, \;\; \exists Hi \in H', \;\; s.t. \;\; h \in Hi$$

This is a minimum Set Cover problem and is known to be NP Complete [31]. To find an approximate solution, we have adopted the Quine-McCluskey optimization technique [32], [33] used widely in Boolean Algebra for minimization of boolean expressions. To describe this method, we define a few terms.

1) Prime Acquaintance: This acquaintance is not completely covered by any other. That means, there is no other single acquaintance whose Hub list covers all of the Hubs in this node's Hub list. However, more than one other acquaintances may together cover all the Hubs in this node's Hub list. Formally, a node A with Hub list Hj would be a *Prime* acquaintance *iff*:

$$\nexists Hi, \;\; s.t. \;\; h \in Hj \Rightarrow h \in Hi, \;\; \forall h \in Hj$$

2) 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 be the set of all the *Prime* acquaintances. Then, a *Prime* acquaintance A with Hub list Hj would be an *Essential Prime* acquaintance *iff*:

$$\exists h \in Hj, \;\; s.t. \;\; h \notin Hi, \;\; \forall Hi \in P \; (i \neq j)$$

For example, if $A = \{1, 2\}, \;\; B = \{2, 3, 4\}, \;\; and \;\; C = \{1, 3\}$, then B or C alone cannot cover all the Hubs of A. So A is a *Prime* acquaintance. However, A does not cover any Hub that is not already covered by either B or C. So A is not an *Essential Prime* acquaintance. On the other hand, no single node covers all Hubs of B, and B covers Hub 4 that is not covered by anyone else. Thus, B is an *Essential Prime* acquaintance.

To query the optimal subset of acquaintances, a node first determines its *Prime* and *Essential Prime* acquaintances. All the *Essential Prime* acquaintances are chosen, and all the Hubs in C that they cover are marked. If any Hub is left unmarked, the non-essential *Prime* acquaintance covering the maximum number of unmarked Hubs is chosen and the corresponding Hubs are marked. This procedure is repeated with the remaining non-essential *Prime* acquaintances, until all the Hubs in C get marked.

This entire procedure is performed first at the source. The source also records all the Hubs it has learnt of, in the packet header. Once this query reaches an acquaintance, if the acquaintance in turn needs to forward the query to its own acquaintances, it does so only if two criteria are met:

- the maximum allowed logical hop has not reached
- there are acquaintances that cover Hubs not already seen by source

If the above conditions are satisfied, the acquaintance in turn repeats the minimum subset selection process above to find out a list of acquaintances to query that cover Hubs not already seen by the source.

D. Packet Transmission to a Hub List

In OBR, all packets (*query*, *response*, *data*, *update*) are sent from one node (source) towards the Hub list of another node (destination) that is contained in the packet header. The exact procedure of transmitting a packet to a Hub list has multiple possibilities leading to different impact on protocol performance. We study three such methods as different schemes of OBR.

1) Scheme 1 (Sequential): In this scheme, the source tries to forward the packet towards a Hub in the list which is geographically nearest to its own Hub. From then on, each intermediate node performs greedy geographic forwarding to push the packet to the neighboring node that is closest to the intended Hub's center coordinates. When a local maxima occurs, the packet is redirected towards the next unvisited Hub in the destination's Hub list that is closest to the current Hub. If no neighbor can be found that is closer to this intended Hub than the current node, the next nearest unvisited Hub is tried. Thus, the Hub that could not be forwarded to from this node, shall be considered again later by another node during another local maxima, till no such node exists. If the node responsible for this redirection was within the previously intended Hub, that Hub is marked inside the packet header as visited by the packet. This process is now repeated to forward the packet towards the center of the new Hub. In this way, a packet traverses from one Hub to another in the list, until either the destination is found, or all the Hubs in the list are visited. We shall also refer to this scheme as *OBR Sequential*, which is illustrated in Figure 3(i). As shown in the figure, a node in Hub 2 failed to forward to any other node closer to Hub 3 than itself and so redirected the packet to the next nearest Hub 4. At a later stage, a node in Hub 5 finds Hub 3 still not visited and is able to redirect the packet towards Hub 3.

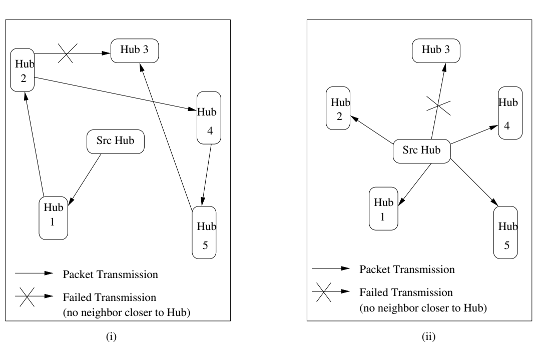

Fig. 3. (i) OBR Sequential (Scheme 1) (ii) OBR Simulcast (Scheme 2)

2) Scheme 2 (Simulcast): In this scheme, the source Hub sends out a copy of the original packet to each Hub in the destination node's Hub list. If any such packet encounters a geographic hole on the way to its destined Hub, then that Hub may never get that packet from anywhere else. However, even if the destination node is in the Hub that failed to get the packet's copy, the node shall eventually receive the packet from the cache of some other node in some other Hub in its list when it visits them. This scheme shall also be referred to as *OBR Simulcast*, since it simultaneously unicasts a copy of the packet to all Hubs in the list. This scheme is illustrated in Figure 3(ii). As shown in the figure, if the packet towards any Hub encounters a geographic hole (e.g. the packet toward Hub 3 encountered a local maxima before reaching a node within the Hub 3), then no more attempts are made at any later stage to rectify this failure.

3) Scheme 3 (Multicast): In this scheme, the source node forms a *minimum spanning tree (MST)* based on all the Hubs in the destination node's Hub list and the Euclidean distances between them. The Hub in which the source resides is the root of the tree. Once the tree is formed, the source node simultaneously sends a copy of the original packet to all Hubs that are the children of the source's Hub in the MST formed. This MST is also added to the packet in the header. Once the packet reaches a node that is the closest to the center of an intended Hub, that node then retransmits copies of that message simultaneously to all Hubs that are the children of the current Hub in the MST contained in the packet header. If the node fails to forward a copy to any neighbor who is closer to an intended Hub than itself, then this node also tries to forward copies to all children (if any) of that failed Hub. However, the failed Hub shall never receive a copy of the packet again. Figure 4 illustrates this *multicasting* down the MST. As shown, a node in Hub 0 could not find any other node closer to Hub 1 than itself, hence redirected the packet copy towards Hub 2, which was a child of Hub 1 in the MST. This scheme shall also be referred to as the *OBR Multicast*.

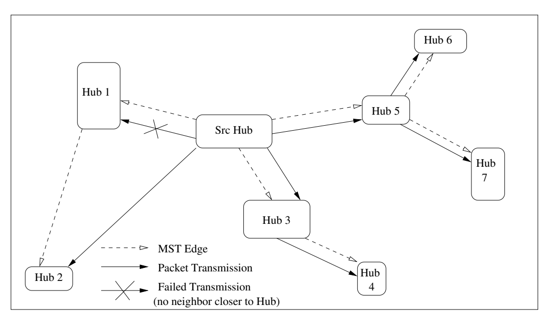

Fig. 4. OBR Multicast (Scheme 3)

E. Connection Maintenance

A session between a source and a destination becomes active when the first *data* packet is sent out from the source to the destination. This session expires when the inter-arrival time of any data packet in the same session exceeds a given threshold at the source. Once an active session is in place, the source puts its current Hub information along with its Hub list in each data packet. The destination reciprocates with similar information on getting the first data packet. From then onward, the source forwards data packets of the same session to the specified current Hub of the destination first, in order to reduce delay. Similarly, if the destination suffers an LAO or MAO Timeout, it notifies the source of the change by sending an *update* packet towards the current Hub of the source first. Such *update* packets are restricted between the two ends of an active session only.

V. Performance Analysis

In this section, we compare the different schemes of our proposed OBR protocol against each other and ABSoLoM. We perform our simulations in GloMoSim [34] where we set up 150 random CBR connections, each sending ten packets with a 512 byte data payload. To assume realistic speeds we refer to the work done in [35], [36], [37], as summarized in Table II. Accordingly, we fix our LAO parameters (i.e.

TABLE II
Real Life Speed

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

Hub Speed (min, max)) to 1 m/s and 10 m/s, and the MAO parameters (i.e. Node Speed (min, max)) to faster speeds of 10 m/s $(23\ mph)$ and 30 m/s $(67\ mph)$. We vary the two global attributes of our framework (i.e., *Hub Size* and *Hub Stay*) and the number of Hubs to study their effect on the *data throughput*, *control overhead* and *end-to-end delay* of OBR Sequential, OBR Simulcast, OBR Multicast, and ABSoLoM. We use five simulation runs with varying random seed values to plot each point in our results, which are as follows.

A. Variation in Hub Size

The Hub size is significant on three fronts in the Random Orbit model. First, for a fixed radio range, a larger Hub means less coverage of each node in a Hub. Second, for a fixed terrain size, a change in the Hub size affects the amount of terrain covered by these Hubs. Third, it means increased overlap among the Hubs. In the following simulations, the Hubs were considered to be square regions with the common size of the sides being varied. The number of Hubs are fixed to 10 and Hub Stay (min, max) is set to 10 $s\ and$ 25 s.

1) Data Throughput: The data throughput is measured in terms of the fraction of the total number of data packets generated that were received successfully. In OBR, a source learns of a destination by first making acquaintances with nodes that are within its radio range, and then using the distributed location database formed by the network of these acquaintances. Thus the Hub Size does not seem to affect the throughput significantly. More specifically however, OBR Simulcast fairs the best since it aggressively sends out multiple copies towards all the Hubs in the list, thereby increasing the chances of the destination node receiving the packet. In OBR Multicast, due to the formation of a minimum spanning tree, when a packet encounters a geographic hole on the way to a Hub in the tree, more often than not it becomes difficult to redirect it to any other Hubs that are the children of this unreachable Hub in the tree, since they are usually farther away with respect to the Euclidean distance of their center coordinates. Moreover, no further attempt is made to forward the packet to the unreachable Hub. Thus, in this scheme a single local maxima can prevent a lot of Hubs in the list from getting the packet, thereby displaying the worst data throughput amongst all the OBR variations. In OBR Sequential, if a packet cannot be forwarded to a Hub, that Hub remains marked as unvisited in the packet header. Thus, at later times, this Hub is re-considered when the packet is traversing the other unvisited Hubs in the list, thereby displaying data throughput in between the other two OBR schemes. ABSoLoM on the other hand, does not make use of the underlying mobility. The basis of making acquaintances is via neighbors that move out of the neighborhood within a specified time. Unfortunately, according to the Random Orbit model, even though nodes keep moving from one Hub to another, nodes that have common Hubs in their list may move together, thereby having a lot of acquaintance-ship formation and breakage. This affects the data throughput negatively overall, leading to lowest values for this metric. Moreover, with larger Hubs, overlap increases, causing nodes that move out to other Hubs to still be in the same neighborhood, which as discussed above affects the performance of ABSoLoM, that degrades with increasing Hub Size as shown in Figure 5.

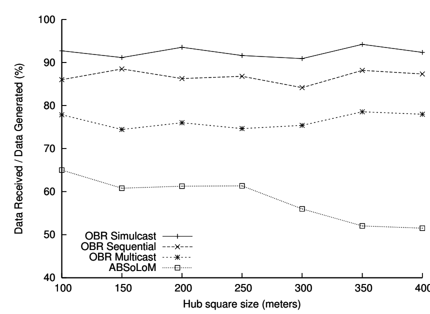

Fig. 5. Data Throughput vs. Hub Size

2) Control Overhead: The control overhead in OBR is measured in terms of the number of *hello, query, response* and *update* packets that are sent. In ABSoLoM, this overhead also includes the *acquaintance-ship request* and *acquaintance-ship accept* packets. ABSoLoM has a periodic overhead of forming and maintaining acquaintances. Thus the overall control overhead is significantly higher than OBR. More specifically, with increasing Hub size, Hub overlaps increase and thus nodes may stay close to each other even when they are in different Hubs. This results in an increased amount of acquaintance-ship breakage, leading to increased rate of acquaintance-ship formation. Thus, the control overhead is seen to increase with increasing Hub Size in Figure 6. In OBR, nodes periodically check for new neighbors to form new acquaintances. But, once an acquaintance is made, its information usually stays valid for a long time (due to a relatively large MAO value), leading to a much lower control overhead than ABSoLoM signifying higher energy efficiency. However, with increasing Hub size, the number of new neighbors increase, resulting in increasing overhead. Among the different schemes of OBR, the Simulcast displays highest overhead as expected due to its aggressive simultaneous unicasting of packets to all the Hubs in the list. Sequential performs at worst similar to the Simulcast when the packet reaches the destination node in the last Hub that it visits in the list. However, on an average its marginally lower than Simulcast. Multicast displays the lowest overhead due to the fact that a single local maxima leads to a number of Hubs not receiving the packet at all.

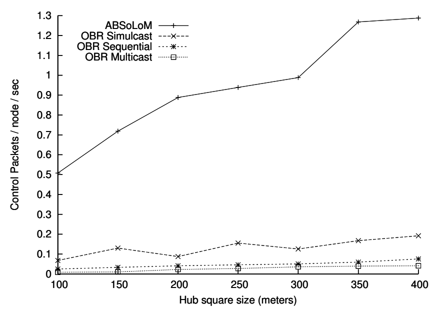

Fig. 6. Control Overhead vs. Hub Size

3) End-to-End Delay: The end-to-end delay is defined to be the time interval between the generation of a data packet at the source, and the reception of that data packet at the destination (including query and response delays, if they were required). As seen in Figure 7, OBR shows marginally higher delay (in the order of a millisecond) than ABSoLoM. More specifically, with increasing Hub size, when a packet enters a Hub it may have to take more hops towards the Hub center, resulting in the overall delay in all the schemes of OBR increasing. In ABSoLoM, due to nodes keeping track of exact location coordinates of its acquaintances, and querying for the same with regards to the destination node instead of Hub list, the Hub Size does not affect the data delay. Among the different schemes of OBR, since in Multicast a number of Hubs may end up not receiving the packet, the destination may physically need to move into a Hub that has either cached the data packet, or overlaps with another Hub that has done the same. Thus OBR Multicast displays highest end-to-end delay. OBR sequential comes next since the packet has to move through the Hubs sequentially to reach the one containing the destination node. OBR Simulcast has the lowest delay due to the simultaneous unicasting of multiple copies of the packet to all the Hubs.

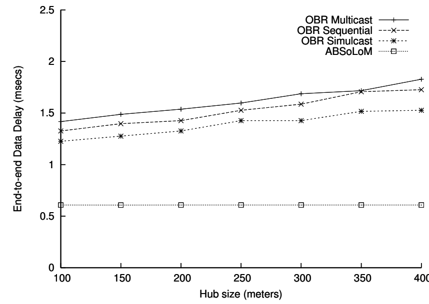

Fig. 7. End-to-end Data Delay vs. Hub Size

B. Variation in Hub Stay (LAO Timeout)

This parameter has a direct impact on the average node velocity. Lower LAO Timeout means shorter time spent by a node in a Hub, increasing the overall time spent in motion at higher MAO speeds. On the other hand, higher LAO Timeout signifies lesser nodes in transition between Hubs, thereby increasing the average node population in Hubs. The number of Hubs are fixed to 10 and Hub Size (min, max) is set to $200\ m\ and\ 300\ m$.

1) Data Throughput: Since nodes in OBR learn of a destination through the network of acquaintances, as long as there exists any mobility that expands this network and the distributed location database associated with it, OBR performs consistently well in terms of data throughput as seen in Figure 8. Moreover, due to the connection maintenance phase, data packets are directed first to the last known Hub of the destination. Thus increased LAO Timeout led to increasing data throughput when packets reach the destination at the first Hub they go to with a high probability. This lessens the negative effect of packet losses due to local maxima as they travel to different Hubs. The relative performance of the different schemes of OBR were as seen before. ABSoLoM is already affected negatively by the underlying mobility as described before. Moreover, with increasing LAO Timeout, as the overall node velocity decreases, ABSoLoM finds it even more difficult to know about other nodes through its acquaintances. Thus its data throughput degrades with increasing LAO Timeout.

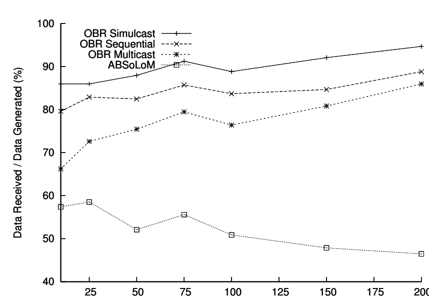

Fig. 8. Data Throughput vs. LAO Timeout

2) Control Overhead: The relative difference in control overhead shown in Figure 9 is similar to that seen in Figure 6 where OBR performs much better in general than ABSoLoM. In ABSoLoM, the higher control overhead is primarily due to the increased number of acquaintance-ship breakage due to nodes staying close to each other for a long period of time. On the other hand, lower mobility reduces the number of new neighbors a node in OBR interacts with leading to a marginal decrease in control overhead with an increase in LAO Timeout. Among the OBR variations, Simulcast has the highest and Multicast has the lowest overhead for same reasons as described before.

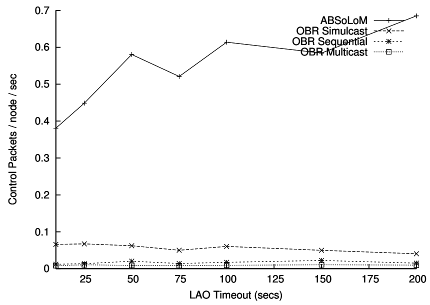

Fig. 9. Control Overhead vs. LAO Timeout

3) End-to-End Delay: As seen before in Figure 7, the delay in OBR is marginally higher overall. More specifically, an increase in the LAO Timeout increases the probability of finding a destination in its last known current Hub, where a data packet is sent first (as described in Section IV-E), causing the delay to decrease gradually with an increase in the LAO Timeout. Delay in ABSoLoM is not much affected by the change in LAO Timeout and is the lowest as seen. Among the OBR variations, Multicast has highest and Simulcast has lowest delay for the same reasons as described before.

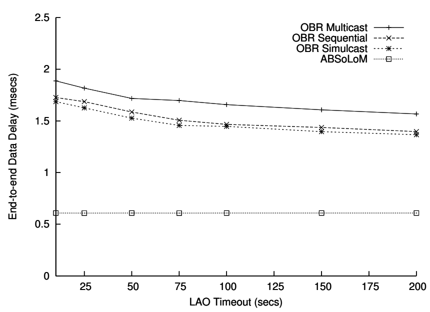

Fig. 10. End-to-end Data Delay vs. LAO Timeout

C. Variation in Number of Hubs

This parameter primarily influences the length of the average Hub list of any node. In addition, a larger number of Hubs increases the probability of Hubs overlapping with each other. The Hub Size (min, max) is set to 200 m and 300 m, and the Hub Stay (min, max) is set to 10 s and 25 s.

1) Data Throughput: As seen in Figure 11 OBR Simulcast is the only scheme not affected by the increasing number of Hubs, since for each Hub in the list, a separate copy of the message is sent. In OBR Sequential, since a single packet keeps traversing through all the Hubs, with larger number of Hubs (leading to longer Hub lists) the probability of that single packet encountering a local maxima and eventually not being able to reach all the Hubs increases. Thus the data throughput decreases gradually with an increase in the number of Hubs. A similar trend is observed for OBR Multicast. However the overall throughput in Multicast is lower than in Simulcast due to the fact that a single local maxima can prevent multiple Hubs from getting a packet, and unlike in Sequential, no more attempts are made to consider these Hubs. As for ABSoLoM, it suffers due to a number of reasons. First, with an increase in the number of Hubs, the overlapping of Hubs increase too. This causes nodes to stay in each other's neighborhood even when they move from one Hub to another, resulting in the breakage of acquaintance-ships that affect the data throughput. Also, the rate of forming new friends increase resulting in an increasing control overhead, which also leads to increased congestion resulting in packet losses that decrease the throughput.

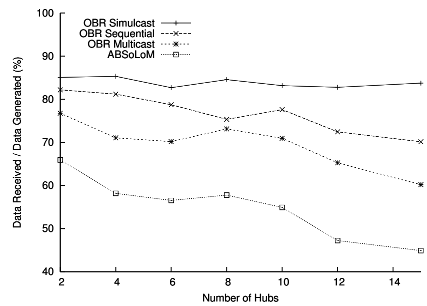

Fig. 11. Data Throughput vs. Number of Hubs

2) Control Overhead: As explained above, an increase in the number of Hubs results in loss of acquaintances in ABSoLoM and leads to higher rates of acquaintance formation. This increases the control overhead as seen in Figure 12. As far as OBR is concerned, all the schemes fair much better than ABSoLoM. Among them Simulcast has the highest and Multicast has the lowest overhead as before.

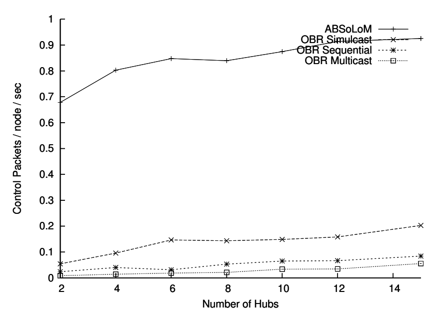

Fig. 12. Control Overhead vs. Number of Hubs

3) End-to-End Delay: As before, Multicast has the highest delay among the different OBR Schemes. With an increase in the number of Hubs, the delay in OBR Multicast also increases due to larger number of Hubs contained in a tree that might miss a packet due to a failure to transmit to some Hub in a higher level. In OBR Sequential too the delay increases with increasing number of Hubs since the packet has to traverse more number of Hubs in the list on an average. OBR Simulcast and ABSoLoM are not much affected, with the latter having the lowest delay as seen in Figure 13.

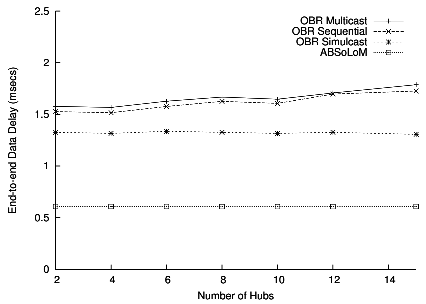

Fig. 13. End-to-end Data Delay vs. Number of Hubs

VI. Conclusion

In this work, we have developed a practical mobility model that identifies a orbital pattern in the sociological movement of mobile users. We have also proposed an Orbit Based Routing (OBR) protocol that is among the first to effectively leverage mobility information in routing packets. OBR uses the underlying orbital mobility to determine a set of likely regions containing any node, and thus outperforms other protocols like ABSoLoM. In short, this novel principle of designing routing protocols based on mobility information is useful for many applications in wireless mobile networks. Currently we are looking at other optimizations for the Simulcasting and Multicasting techniques. Unlike the Sequential scheme as mentioned, failed Hubs do not get a second chance in either Simulcast or Multicast schemes. Moreover, the connection maintenance section only helps the sequential scheme as of now via the knowledge of the last known current Hub, which not only improves throughput but also reduces delay. We hope to do similar improvisations in the other two schemes as well.

References

[1] 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.

[2] 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.

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

[4] C. Perkins and P. Bhagwat, "Highly dynamic destination-sequenced distance-vector routing (dsdv) for mobile computers," *Proceedings of ACM SIGCOMM '94 Conference on Communications Architectures, Protocols and Applications*, pp. 234–244, 1994.

[5] 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.

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

[7] 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.

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

[9] 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.

[10] 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.

[11] 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.

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

[13] 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.

[14] 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.

[15] 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.

[16] 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.

[17] 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.

[18] 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.

[19] 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.

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

[21] C. Barrett, M. Marathe, J. Smith, and S. Ravi, "A mobility and traffic generation framework for modeling and simulating ad hoc communication networks," *Proceedings of 2002 ACM Symposium on Applied Computing (SAC)*, pp. 122–126, 2002.

[22] S. Shah, E. Hernandez, and A. Helal, "Cad-hoc: A cad like tool for generating mobility benchmarks in ad-hoc networks," *Proceedings of 2nd IEEE/IPSJ Symposium on Applications and the Internet*, vol. 11, no. 1, pp. 3–30, February 2001.

[23] 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.

[24] 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.

[25] 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.

[26] 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.

[27] 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.

[28] 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.

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

[30] 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.

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

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

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

[34] 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.

[35] 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.

[36] 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*.

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