SOCIOLOGICAL ORBIT BASED MOBILITY PROFILING AND ROUTING FOR WIRELESS NETWORKS

by

Joy Ghosh

(August 11, 2006)

Major Professor: Chunming Qiao, Ph.D.

A Dissertation submitted to the

Faculty of the Graduate School of

The State University of New York at Buffalo

In Partial Fulfillment of the Requirements

for the degree of

Doctor of Philosophy

Department of Computer Science and Engineering

-

-

To my mother Mrs. Parbati Ghosh, who has been my first and my longest academic counsel, and my father Mr. Himansubhushan Ghosh, who has always encouraged me to think analytically and apply logic to all my actions. It was their constant support, understanding, guidance and patience that has helped me reach where I am today.

ACKNOWLEDGEMENTS

I wish to thank my Ph.D. advisor Prof. Chunming Qiao for his incessant guidance and encouragement in not only my research, but also in my overall academic and professional career. His constant enthusiasm and genuine interest in my academic progress and research work has been a great source of moral support all along. I also thank my co-advisor Prof. Hung Q. Ngo for his immense insight into the theoretical analysis of my research, without which it would have been impossible. I would like to thank my other committee member Prof. Shambhu Upadhyaya for his valuable suggestions and comments regarding the practical aspect of my work. Thanks also goes to Dr. Minkyong Kim at Dartmouth College for agreeing to review my dissertation and for her insightful comments on my work during her visit to our University at Buffalo.

I would also like to thank Prof. Matthew J. Beal for not only helping me to understand a particular machine learning topic in a very limited time, but also for his help in formulating a critical part. Thanks to Dr. Cedric Westphal at Nokia Research Center for his valuable comments on my work and his subsequent collaboration with us.

I wish to specially thank my colleague Dr. Sumesh J. Philip for truly being a friend, philosopher and guide for the most part of my journey to this point. Thanks to Dr. Kris Schindler for all his mentoring that made my initial years at Buffalo a joy to remember. Special thanks to my colleagues Seokhoon Yoon, Nirmal Thangaraj and Duc T. Ha for their sincere collaboration with me in my research, and to Prof. Murat Demirbas, Dr. Ramkumar Chinchani, Mohit Virendra, Ajay Venkatsuresh, and other PhD students in our C.S.E. Research Suite for our occasional in-depth discussions.

I would like to extend my gratitude to Prof. Thomas Gross and Cristian Tuduce at ETH Zurich, for providing us with their mobility trace data that proved extremely crucial to my research and for Prof. Gross' comments on my work during his visit to our University at Buffalo. I also thank Mirco Musolesi at University College London UK, for our discussions during his visit to University at Buffalo.

I thank all my near and dear friends from University at Buffalo including Anantharaman Ganesh and Devika Thakkar for bearing with my philosophical streak for so long. Last but not least, I wish to thank Alpina Kulkarni, without whose loving support, encouragement, and patience, this would have been a very difficult journey.

Contents

List of Figures

List of Tables

ABSTRACT

A study of wireless users' mobility has been proven beneficial to several applications such as radio link lifetime estimation, location prediction, resource allocation, etc. However, the suggested approaches to studying mobility have varied in terms of the details of the mobility information required. In our empirical study, we show that each network user has a probabilistic list of places (e.g., buildings on campus) to visit each day, also referred to as a *mobility profile*. We further show that over a period of time (e.g., a week) a user may repeatedly follow a mixture of mobility profiles with certain probabilities, also referred to as a *sociological orbit*. Unlike earlier work, we present a novel profiling technique capable of processing sporadic mobility data (which is more commonly found) and building mobility profiles for each user, using a Mixture of Bernoulli's distribution based on the Expectation-Maximization algorithm. We demonstrate the usefulness of our mobility profiles in hub-level location predictions with 10% to 30% higher accuracy than a common statistical method.

We propose a suite of Sociological Orbit aware Location Approximation and Routing (SOLAR) protocols that leverage upon mobility profile information exchange. Within MANET, extensive performance analysis shows that our SOLAR variations significantly outperform 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. Within Intermittently Connected Mobile Ad hoc Networks (ICMAN), similar results are obtained by comparing our multi-path hub-level routing method called SOLAR-HUB, and two variations of user-level routing techniques (S-SOLAR-KSP and D-SOLAR-KSP) based on "contact probabilities" with the performance of Epidemic Routing.

Finally, we formulate a novel routing problem within probabilistic graphs of finding an optimal delivery sub-graph that maximizes the delivery probability. Our multi-fold solution proves the hardness of our optimization problem and presents an algorithm to approximate the objective function. We then model user mobility with Semi-Markov Processes to compute pair-wise user contact probabilities, and propose and highlight the superiority of an edge-constrained routing protocol over a probabilistic routing protocol, and an epidemic routing protocol proposed in literature for intermittently connected networks.

Chapter 1

Introduction

Among the most critical services required of any mobile wireless system are reliable location prediction strategies and efficient routing schemes to both locate and route data packets to users within the dynamic network topology. Although many proactive, reactive, and hybrid approaches have been suggested in the literature for various types of mobile wireless networks, mobility of nodes is still considered a big threat to achieve good protocol performance. In this dissertation, we shall address this challenge.

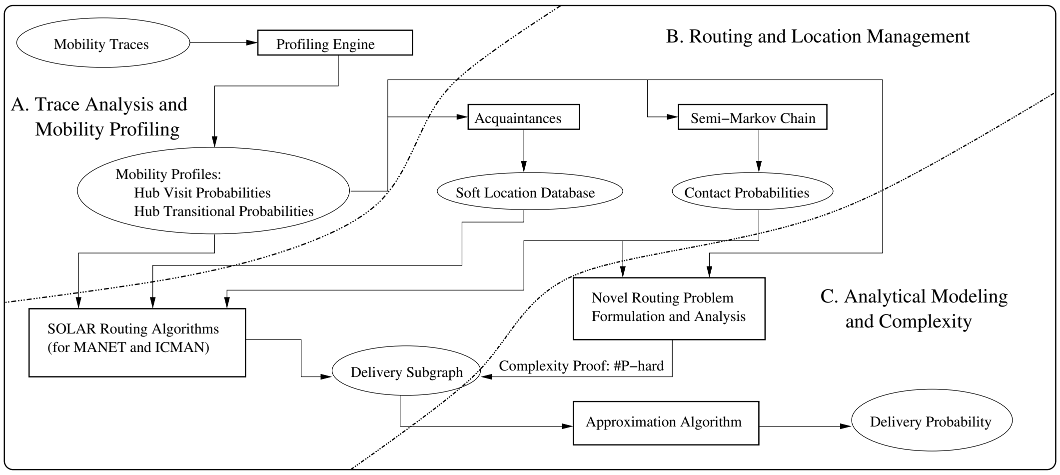


Figure 1.1: The Overall Picture

Our overall contribution is illustrated in Figure 1.1 which shows where and how our different research areas (shown as rectangular boxes) fit in. We approach the routing problem in three major areas: **A.** Trace analysis and mobility profiling; **B.** Routing and location management; and **C.** Analytical modeling and complexity. In the following sections, we shall briefly touch upon our contributions in each of these areas, which shall be described in much greater detail in the rest of this dissertation.

1.1 Trace Analysis and Mobility Profiling

We note that a key part in attempting to solve routing issues related to user mobility lies in a better understanding of the underlying user mobility itself. We have observed strong sociological influences on user movement and have attempted to incorporate that observation not only into building a practical mobility framework, but also in profiling users based on their mobility patterns, as described next.

1.1.1 A Hierarchical Mobility Framework for Routing

Mobility pattern of wireless users has been shown to affect all routing protocols in various ways [7]. The popular Random Waypoint mobility model [26] was liberally used due to its simplicity in implementation and suitability for theoretical study and analysis. In this model, a node randomly chooses a destination point within the terrain and approaches it linearly with a velocity randomly selected from a specified range. On reaching the point, it pauses for a specified time and then repeats the process. In recent times however, such randomness has been deemed unrealistic for modeling real life mobility. To that effect, several mobility models/frameworks have been proposed that can be categorized as either *Entity* (motion of each individual node), *Group* (collective motion of nodes in a group), and *Scenario* (geographic restrictions on movement) based. Our goal was to try and extract mobility information from such mobility models that can be directly incorporated into routing decisions. To this end, we proposed a macro level mobility framework called ORBIT [47], that can integrate multiple mobility models into a multi-layer hierarchy. ORBIT is a practical solution towards modeling mobility in the real world of mobile device users, since it is based on our observation of the of sociological movement patterns of users. We also designed a simple but efficient ORBIT Based Routing (OBR) protocol that makes direct use of the underlying orbital mobility pattern, along with the concept of acquaintances to perform much better than existing solutions. We analyzed the performance of multiple OBR schemes against each other and with ABSoLom [48].

1.1.2 Mobility Profiling Techniques

The mobility of users forming a mobile wireless network causes changes in the network connectivity and may even lead to intermittently connected networks. On one hand, nodal mobility may increase the overall network capacity [54]. On the other hand, it may make it challenging to locate users and route messages within the network. Researchers have analyzed mobility traces to various ends and have suggested numerous practical mobility models. However, we are the first to identify the sociological impact on user movement in the formation of what we call "sociological orbits".

In particular, we present a technique [44] that can process sporadic mobility data more generally found amongst real traces and generate mobility profiles for individual users. We validate our initial and intuitive claims to the sociological influence on a wireless user's movement pattern via an empirical study of mobility trace data. We process Access Point (AP) level mobility to building (hub) level movement, and represent each user's daily mobility as a binary vector of hub visits where a particular bit value "0" signifies non-visit, and "1" signifies a visit to a particular hub indexed by the position of the bit in the vector. We further represent this entire H-bit vector as a point in a H-dimensional space where H represents the total number of hubs in the system, and the bit values signifies the values along each corresponding dimension. We apply a Mixture of Bernoulli's distribution using the Expectation-Maximization algorithm to group each point into separate clusters represented by a cluster mean, which we call a "mobility profile".

In addition, we highlight the usefulness of our mobility profiles in performing hub level location predictions that are not only 10% to 30% more accurate than general hub visit frequency based statistical predictions, but also incur much less computational overhead. Besides predicting whether a user is going to visit a particular hub on a given day (at any time of day), we also perform hub transitional prediction. In the latter case, given a current hub we study the predictions for the next sequence of hubs visited on the same day.

1.2 Routing and Location Management

We leverage upon our understanding of the underlying user mobility in making better routing decisions and location predictions. In this section, we shall discuss the different steps taken to solve the routing challenge.

1.2.1 Location Management Schemes

A family of protocols [33], [89], [135], [105] has been proposed for MANET which makes use of *location management* and *geographic forwarding* for routing data packets. Nodes are assumed to be aware of their own location via either a GPS receiver [42], or other localization schemes [81]. When a source needs to communicate with a destination, it first tries to find the geographical coordinates of the destination using a location service. Once the location is obtained, each packet is forwarded to one of the source's "neighbors" (i.e. other nodes within radio transmission range) which is geographically closer to the destination than the source itself. The neighbor that receives the packet forwards it in a similar fashion till the packet finally reaches the destination. Geographic forwarding lends itself as an attractive candidate for routing in MANET, since the amount of network topology information that needs to be stored by each node is minimal and since the data packets need not store the entire route within them anymore. Moreover, since an end-to-end session does not depend on any specific route, the effect of link breakage on a session is greatly reduced. This forwarding technique however, suffers from lack of sufficient node density that can give rise to what is known as a "geographic hole". To that end, literature has proposed planar graph based face routing [22, 75] to forward around holes if possible. However, the advantage of having a localization scheme comes at the heavy expense of maintaining an up-to-date location database of all nodes in the network. Most approaches mentioned in literature divide the entire terrain into ordered grid based regions and either assign specific regions to nodes as their home regions, or assigns each node the task of acting as the location server for other nodes. Each node in turn is also responsible for maintaining the freshness of this distributed location database by periodically updating their own location information. Deployment of such schemes in MANET also requires a fair amount of node density to avoid empty home regions. Overall, the accuracy of such schemes come at a heavier price, when compared to the simple reactive approaches mentioned above.

1.2.2 Using Node Mobility to Routing Advantage

To bridge this divide between the low maintenance, low performance flooding based protocols and the high maintenance, high performance location management based protocols, we explored the possibility of using mobility of the nodes to our advantage. One of the main advantages of node mobility lies in the fact that even though a single node may not know all other nodes in different parts of the network, their collective knowledge as nodes spread out gives rise to an important distributed location database. Our first work was to propose a novel framework called Acquaintance Based Soft Location Management (ABSoLoM) [46], which utilizes a concept of maintaining acquaintances in order to tap into this distributed "soft" location database for the purpose of efficient geographic routing in MANET. This work differs from other grid-based location management schemes in that it makes no prior assumptions regarding either the shape or the size of the network terrain, or the density of the nodes. ABSoLoM is also informal in nature, not requiring any strict management strategies like computation of grids, assignment of location servers or home regions, location registration, etc. Our main idea is inspired by the experimental studies in social sciences by S. Milgram [99]. In this pioneering work, the author spoke of the likelihood of two individuals in a social network to be connected through a short sequence of intermediate acquaintances. On repeated experiments amongst different pairs of strangers in USA, the average length of the intermediate sequence came out to be around 5 or 6, which later became popular as the principle of "six degrees of separation" [55]. In their study, however, the source was informed about the approximate location of the destination and the source had to forward the packet to acquaintances known to itself only on a first name basis. In our case, the source has no idea regarding the location of the destination, but it uses acquaintances to search for the destination with the hope of finding them being connected via a short list of common contacts, as observed by the studies mentioned above. Thus we make direct use of an effect of mobility that geographically separates a node from its acquaintances to help the node learn of different parts of the network.

1.2.3 Mobility Profile based Routing

Within a MANET, node mobility has been shown to have a significant impact on the routing protocol performance [7]. This has led to numerous studies on the practicality of the different mobility models that are commonly used in mobile network research. The Random Waypoint model [26] has been a favorite for its simplicity and suitability for theoretical study and analysis. In reality however, nodes (i.e. MANET users) move purposefully (e.g., going from a conference room to a cafeteria) while being subject to certain restrictions (e.g., geographical constraints), resulting in certain amount of determinism in their motion as validated by our empirical studies of mobility trace data.

Based on the above, we propose a macro-level mobility framework called ORBIT [50], based on a partially deterministic orbital movement pattern of mobile users around some specific places of social interest called *hubs*. The term "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 amount of time, without specifying any rigid schedule or routes (i.e., a partially deterministic movement pattern). We also propose an effective routing scheme for MANET called the Sociological Orbit aware Location Approximation and Routing (SOLAR) protocol [49, 51] to take advantage of the spatial/temporal locality of the mobile users (nodes) around these hubs. The proposed ORBIT framework is not only general enough to be realistic, but is also specific enough to be useful. In particular, the proposed SOLAR protocol 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 SOLAR over other conventional protocols, such as DSR [26] and LAR [79] in terms of higher data throughput, lower control overhead, and lower end-to-end delay.

One of our main assumptions in the above work has been sufficient user density within the MANET to perform greedy geographic forwarding. However, in a different sort of wireless network such as an Intermittently Connected Network (ICN) this assumption does not hold strong. An ICN is often modeled as a weighted graph, where the capacities and durations of edges (radio links) between nodes (users) are time varying due to user mobility. As such, there is no guarantee of a contemporaneous end-to-end path from the source to the destination through intermediary peers. This sets an ICN [1] strictly apart from a conventional mobile ad hoc network (MANET). Consequently, traditional MANET routing protocols as well as our proposed SOLAR protocols are rendered useless within ICN. In this research, we focus on Intermittently Connected Mobile Ad hoc Networks (ICMAN), which lacks wireless networking infrastructure (similar to a regular MANET), in addition to having non-deterministic user mobility like in an ICN.

According to our findings in [44], we consider an ICMAN where each user may have a list of hubs to visit such that each user may stay in its current hub for some time before moving to another hub in the list with a certain probability. We propose ([45,52]) a sociological orbit aware location approximation and routing (SOLAR) protocol called (*SOLAR-HUB*) that takes advantage of user mobility profiles to perform "hub-level" routing. For comparison study, we also propose two different variations of multi-path SOLAR protocols (*Static SOLAR-kSP* and *Dynamic SOLAR-kSP*) that perform "user-level" routing based on user "contact probabilities" computed from our user mobility profiles. We compare the performances of the different SOLAR protocols with that of the simple and efficient Epidemic routing [128] and show that all SOLAR protocols outperform Epidemic routing in terms of higher data throughput, lower network overhead, and lesser end-to-end data delay.

[1] also referred to as Delay Tolerant Networks

1.3 Analytical Model and Complexity

In forming our SOLAR routing protocols, we make use of the pairwise user contact probabilities and also the delivery probability between a pair of source and destination users. In this part of our research, we aim to provide mathematical models for the computation of these probabilities and also analyze their complexity. We use a Semi-Markov Chain to model the user mobility, so as to obtain the pairwise user contact probabilities. Next, we focus on a novel problem of given a graph, to find the optimal delivery subgraph that can maximize the *source,destination*-connectedness problem. In other words, we aim to compute the delivery subgraph with maximum delivery probability. We analyze this problem and prove its complexity, and then propose a polynomial time approximation algorithm to compute the delivery probability of a given delivery subgraph. Our future direction would be to develop an optimal routing strategy that would route packets along the optimal delivery subgraph. In this dissertation however, we propose an edge-constrained routing algorithm (EC-SOLAR-KSP) that makes use of all the analytical models and performs better than other probabilistic routing schemes proposed in literature for intermittently connected networks.

1.4 Summary

The rest of this dissertation is organized as follows. In Chapter 2, we present some background information to motivate our research and highlight our contribution by discussing some related work. In Chapter 3, we describe the details of our ABSoLoM protocol that takes advantage of node mobility in a novel way. In Chapter 4, we describe and analyze our initial hierarchical mobility framework based on the sociological movement patterns of mobile device users. In Chapter 5, we describe our novel technique for profiling the mobility of wireless users and demonstrate its usefulness in location predictions and also suggest several applications. In Chapter 6, we describe and analyze a suite of Sociological Orbit aware Location Approximation and Routing (SOLAR) schemes for MANET, and in Chapter 7, we extend similar research to ICMAN. In Chapter 8, we present theoretical analysis of our SOLAR framework. Finally in Chapter 9, we summarize our major contributions and conclude this work by discussing future direction of research.

Chapter 2

Background and Related Work

Mobile wireless systems are becoming a norm with recent technological advances that give the users a gamut of wireless devices and application choices. Commercialization of wireless technology has seen the advent of devices like cell phones, PDAs, and Laptops, powered with various economical wireless networking solutions like the IEEE802.11 [1] and Bluetooth [56]. Such revolutionary technological progress has been made possible by equally ground-breaking research in the wireless networking domain in lieu of user mobility. The mobile wireless system infrastructure already has many challenges, the effect of node mobility being one of the most critical. We shall discuss the general issues in such networks and how they eventually affect routing decisions within them.

2.1 Routing Issues within Mobile Wireless Systems

A mobile wireless system typically consists of devices equipped with some wireless transmitter and receiver, interacting amongst themselves following some chosen technology like IEEE802.11 [1], Bluetooth [56], Sensors [63], etc. Following are the characteristics of the network and the devices most commonly observed.

2.1.1 Network Characteristics

Although the specific characteristics of a mobile wireless network will depend on the kind of technology adopted, there are some common issues in all of them, as described below.

Topology Dynamics The network topology graph obtained by considering all the nodes within communication range of each other as having a point-to-point link, varies proportionally with mobility. A link may either break due to direct mobility of two nodes that cease to be in each other's communication range, or due to the appearance of an obstacle between them that causes interference. Even for static networks, dynamic turning on and off of devices may cause the illusion of nodes joining and leaving the network at different points in time. This too contributes to the variation in the resulting network topology.

Network Partitions For certain networks like the delay tolerant networks, no end-to-end path between two nodes may ever exist at any given point in time. These networks may remain partitioned for long and the region of partition may vary with time.

Time Varying Bandwidth/Capacity The condition of wireless channels depend on a lot of factors like noise, interference, fading, node mobility, etc. Thus the link capacities or the bandwidth of the channel varies with time.

High End-to-end Latency Unlike wired networks, due to the limited bandwidth and dynamic topology, the end-to-end latency of such networks are usually more significant. Specially for situations like in delay tolerant networks, frequent network partitions lead to long queuing of packets in the intermediate nodes.

Asymmetric Data Rates Due to the time varying channel conditions and the individual node characteristics, no assumption maybe made on the capability of directional links. In some cases, like in deep space networks, a return channel may even be absent. In other cases, the up-link and down-link may have completely different data rates.

Lossy Links vs. Congestion In wired networks, loss of packets is considered a fair indication of network congestion. However, in wireless networks packets may be lost due to temporary link loss, as well as buffer overflow due to congestion.

Security Considerations Most wireless systems like sensor networks or mobile ad hoc networks function on the basis of mutual trust in between nodes. Thus, although deployment of such networks requires no prior infrastructure and is deemed as an easy process, securing it is a challenging task.

2.1.2 End System Characteristics

Portability is of great concern when designing mobile computing devices. Thus, when these same devices are empowered with wireless technology, the amount of resources available are limited. The key elements of consideration are listed below.

Power Considerations Most mobile computing devices are battery powered. Hence they have limited longevity. Some of these devices (like the sensors) often have low duty cycle to reduce the consumption of power. They may wake up at regular intervals, or may be event driven.

Buffer Management Usually the storage capacity of these nodes are limited. So they may need to be augmented by external persistent storage devices, or may have to practice some sort of effective queue management techniques to minimize buffer overflow.

Limited Memory and Processing Capabilities To be able to keep the cost of such devices at a low and affordable range, they are not memory or processor intensive. These are usually built with some specific range of applications and users in mind.

2.2 Routing Objectives and Existing Approaches

Given the network and end system characteristics of a mobile wireless system, routing becomes a really challenging problem. Nevertheless, any attempt to solve it should share some basic objectives:

1. No infrastructure requirements - ready deployment and completely distributed algorithms.
2. Correctness - exhibit some sense of direction, or some means of approaching the destination, despite of mobility.
3. Avoid looping - instability in network should not have invalid data congesting it.
4. Congestion detection - ability to distinguish loss of packets due to congestion from those due to lossy links.
5. Proactiveness - make use of all data available to proactively perform certain tasks, within an acceptable cost of protocol overhead.
6. Reactiveness - ability to respond to live traffic with maximum efficiency.
7. Energy Efficiency - incorporate power saving mechanisms for enhancing the network life time and avoid unnecessary tasks.
8. Scalability - provisions for accommodating a larger network with more devices.
9. Robustness - means to battle the effects of mobility and take advantage whenever possible.

The routing approaches suggested in literature so far can be broadly categorized as following:

Proactive Protocols Protocols like Distance Vector, or Link State Routing proactively maintain some form of a routing table that is computed based on the network topology. They resort back to flooding the network with local link information, which helps them gain global knowledge over several iterations. In case of static networks, this method works very well, greatly reducing the end-to-end routing delay. However, in the face of node mobility, the rapidly changing network topology mandates the process of building a routing table to go on forever. Even then, the freshness of the global image of the network at each node depends on the rate of mobility. This results in network congestion due to control packets, leading to a high overhead. Moreover, in networks with no end-to-end path, this approach may fail to provide any path at all.

Reactive Protocols Protocols like DSR [26] and LAR [79], reactively find a route to a destination with unknown route. The lack of any proactive routing table is complemented by a primitive flooding scheme in the absence of any knowledge. This process inherently has longer delay when compared to a purely proactive approach, but they tend to make use of any available knowledge to optimize their routing decisions. DSR works in the promiscuous mode (snooping into other node's packets) and actively makes use of route caching to minimize overhead and routing delay. LAR uses prior location and velocity information of a node to estimate the region most likely to contain it at a later time. However, all such source routing schemes that use entire route caching suffer from frequent invalidation of routes due to high probability of mobility induced link failures in that cached path. Moreover, these protocols also fail to provide any route for networks with a topology that has a time variant disconnection.

Hybrid Protocols Some approaches like ZRP [103] try and take advantage of both Proactive and Reactive methodologies. Proactive protocols may compute the most optimal route prior to the arrival of traffic, but suffer in the effort of trying to obtain a global up-to-date network topology information. Thus, in these hybrid approaches they consider only subgraphs of the entire network, called zones, and practice proactive routing within them. For inter-zone routing, they resort to reactive mechanisms. Needless to say, they still do suffer from the basic limitations of the proactive and reactive protocols in general, in the face of node mobility.

Store and Forward Protocols As mentioned above, none of the proactive, reactive or hybrid approaches can succeed in finding routes in a network topology that is disconnected at different points in time. To that end, certain protocols adapt a "store and forward" kind of philosophy, wherein they hold on to data till they can transmit to someone else at a future point in time. For MANET, [93, 128] proposed routing schemes for intermittently connected networks using this philosophy. For delay tolerant networks, recent work in [67] assumed knowledge oracles and used a link capacity function to find minimum cost paths as a cascade of time varying links. Their usefulness is however limited by the practicality of their assumptions.

2.3 Acquaintance Based Soft Location Management

In [103], the authors tried to complement pure reactive approaches by proactive strategies within zones to reduce the location discovery delay and restrict the flooding required. In [24], the reactive part of [103] was supplemented with the GPS based query optimizations. In [12, 25], the advantages of reactive nature of DSR is combined with the location-based solutions to achieve improved efficiency, but authors in [12, 25] still restrict themselves to source routes and consequently suffer more from link breaks than the protocols based on geographic forwarding. In [13], the authors introduced an efficient way of disseminating geographical position of nodes. By this mechanism, nodes far apart update each other about their own positions less frequently than those close together, based on the relative velocity. Using this concept, several multi-point communication protocols have been proposed, such as [23], [14]. In [71], the authors used a clustering algorithm to restrict the flooding required by DSR. Consequently literature describes several protocols (e.g. [5, 6, 34]) that extends this concept of clustering to propose solutions for multicasting issues in wireless ad hoc networks.

In [18], the authors did extensive analysis of an anchor based routing technique that uses a concept of "anchors" to loosely source route messages towards the destination. In their work, they tried to make geographic routing more efficient by finding some fixed geographic locations (not nodes) en-route to the destination that are called "anchors". Messages are forwarded geographically towards an anchor point until they reach a node close enough to that point. Thereafter these messages are forwarded to nodes that take them towards the next anchor point, till they reach the destination node. The authors used special nodes called Friend Assisted Path Discovery (FAPD) [19] responders that maintain friendship amongst themselves to learn about the irregular network topology and to help sources find appropriate anchor points towards their respective destinations. These responders do not take part in the forwarding path themselves. Thus, this work differs from ours in the following ways: we do not reserve nodes for specific tasks (like the responders); we make use of acquaintanceship not only to learn collectively about the network neighborhood but also to forward packets to the destination.

2.4 Mobility Models and Frameworks

Random Waypoint [26] is the most popular *Entity based* mobility model in literature. In [136], the authors studied an average speed decay problem in Random Waypoint and in [17], the author enhanced the model by using acceleration to smoothen changes in speed and direction. To account for obstacles, the authors in [69] proposed a mobility model based on voronoi graphs. In [124], 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 [92], the authors used *renewal theory* to guarantee a steady state in node movement distributions, while those in [86] 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 [61, 62], 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 [30], 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 [7], 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 (see Figure 4.2(a)) to generate more realistic models.

In [94], 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 [96], the authors applied *transportation theory* to model: *City Area*, *Area Zone*, and *Street Unit*, at three hierarchical levels of detail. Similarly, the authors in [83] 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 [11] proposed a framework for graph based modeling of mobility and traffic in large scale MANETs, while in [119], 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.

2.5 On Profiling Mobility of Wireless Users

Many researchers have tried to model practical mobility in various ways to achieve different goals. Earlier work on mobility modeling [30] was done mostly with Mobile Ad hoc NETworks (MANET) in mind. For example, some [115] used mobility pattern analysis to minimize radio link changes via appropriate selection of next hop within radio range. While the authors in [123, 130] performed physical location prediction via continuous short-term and short-range tracking of user movement, we had leveraged on our assumptions on “sociological orbits” to perform efficient routing within MANETs [50, 51].

Our study of user mobility traces is motivated by the need to extract practical mobility information, which may potentially benefit applications such as location approximation and routing within all types of networks such as MANETs, DTNs, wireless access networks, etc. More specifically, it is noted that wireless users belong to a larger social environment and as such, their movement behavior is subject to several location dependent sociological constraints (in addition to speed limits and specific walkways, as described in [7]). In particular, on any given day, each user may visit a list of places of some social importance (which we referred to as “hubs” in [50]) in some probabilistic manner, creating what we refer to in this work as a “mobility profile”.

One of the earliest attempt to understand the 802.11-based network access patterns, traffic load and user preference of wireless over wired networks was made by the authors of [82, 125, 126]. However, the main focus of these projects were not on the mobility pattern of individual users. An extensive campus-wide 802.11 based wireless network testbed was setup in Dartmouth College, and its traces has been studied by the authors in [60, 76, 77], focusing more on AP-centric parameters. In particular, they studied amongst many things, the number of periodic visits to a particular AP or building, the length of such periods, and the frequency with which any sequence of 2 locations were visited in succession. However, for such information on periodic and sequential visits to location pairs to be useful in statistical location prediction for a set of n hubs, one would need all ${}^{n}P_2 = n * (n-1)$ permutations of visit sequences. In contrast, our proposed methods for profiling mobility based on sociological orbits only requires information on up to n hubs to be collected with a much lower frequency of updates as mentioned earlier in the work.

There have been several other related work on analyzing 802.11-based wireless access patterns at ETH Zurich [127], University of North Carolina at Chapel Hill [35], University of California at San Diego [8, 98], MIT [9] and University of Saskatchewan [116], just to name a few. Most of these work focussed on topics such as: training mobility models for use in network simulation; taking wireless network load measurements for general resource allocation and capacity planning; understanding the user mobility in terms of periodic visits to APs; improving web-caching, QoS-routing, etc.

However, none of these work explored the notion of sociological orbits and in particular, mobility profiles. As a result, when it comes to predicting the next AP to be accessed for example, they often employ common statistical approaches whereby access to different APs is assumed to be independent. It is worth mentioning that much earlier work exists on understanding cellular users' mobility patterns, with primary objectives being triggering location updates based on user movement and improving paging services for example [4]. However, they dealt with a set of cells that have a much larger area than hubs, and whose physical boundary may not have special social implications as hubs do.

There also exists a few studies on the social aspects related to wireless networks. For example, the work in [3] analyzed the NTT DoCoMo's i-mode users' email partners to infer their social relationships. Similarly, the authors of [100] studied the social influence on wireless networks as we did. However, they proposed mobility models based on *Social Network Theory*. In particular, they introduced a "sociability factor" to define the social relationships between users, and used that to characterize user groups that may have correlated mobility. Our work is different in that we profile users' mobility in terms of the locations they visit, instead of modeling their physical movement in between, or within those locations in terms of velocity and direction of the movement as in [100, 127].

Authors in [85, 120], base their computational models on empirical data that are filtered to provide stable mobility data sets spanning regular intervals of consecutive days. In most real systems however, we observe lots of irregularity in terms of wireless users' usage pattern of the network, where a user may not be present in the network consistently for the entire period of observation. Our work is novel in this sense that it provides for a mechanism to profile users based on sporadic mobility data without any filtering and yet providing a leverage in performing location prediction with increased accuracy when compared to general statistical methods.

In this research, we not only validate the existence of such mobility profiles via mobility trace analysis, but also show that in practice, a user is usually associated with more than one profile. In particular, we find evidence of a probabilistic mixture of such profiles that stay valid for a long period of time (i.e., several days or weeks), after which a different mixture of profiles will be in effect. The data analyzed in this work is collected on the ETH Zurich campus that has many users appearing only for a few months, and in a very sporadic manner. Otherwise, it is similar in content to that available from the Dartmouth campus (both are obtained from Access Point system logs). However, compared to the most related (and yet much different) work in [32], this work primarily focuses on the orbital parameters, in particular on the *user-centric* parameters like the user mobility profiles and its applications, whereas [32] focuses more on AP-centric parameters. Our mobility profile based hub-level location prediction is shown not only to be more precise than a common statistical method, but also to incur a much lower overhead. Note that although this work analyzes data from a campus-wide wireless access network (instead of a MANET, as data from former is more readily available), our mobility profiling and location prediction techniques are applicable to other types of networks as well, since the movement of users is ultimately influenced by their social environment.

2.6 Routing within Mobile Ad Hoc Networks (MANET)

The growing awareness of the effect of node mobility on protocol performance has led to research on practical mobility models and mobility adaptive routing schemes. A popular mobility model which is used exhaustively in mobile network research, Random Waypoint, is clearly not very realistic and has practical limitations as pointed out in [136]. In [17], the authors suggested enhancements to the model itself, including the use of acceleration to smoothen changes in speed and direction. While the authors in [69] focused on the application of voronoi graphs to model mobility in face of obstacles, those in [124], 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 [92] used *renewal theory*, while authors in [86] introduced the concept of *stochastic correlation* in their VUM (variable user mobility) model for cellular systems.

Although, the previously described models could simulate the mobility of individual nodes, they could not capture the motion of entire groups in scenarios such as military drills, disaster recovery, search party etc. Accordingly, the authors in [61,62] 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. Their work was later extended to include a mobility vector model, which smoothened the sudden changes in speed and direction. In [30], 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 [7], 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 describes several proposals that model mobility as a hierarchy. In [94], 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 [96], the authors applied *transportation theory* to model: *City Area*, *Area Zone*, and *Street Unit*, at three hierarchical levels. On the same note, the authors in [83] proposed the Metropolitan (*METMOD*), National (*NATMOD*) and International (*INTMOD*) mobility models to model movements within metropolitan areas, in between them, and in between countries respectively. Despite this wide range of study on practical mobility, no prior work has studied the mobility of MANET users in between certain geographic regions of some social significance (which we call hubs).

Meanwhile, work has also been done on routing protocols to counter the adverse effects of node mobility. For instance, flooding based source routing schemes [26, 104] employ aggressive route caching techniques to reduce the route discovery delay. The proposed protocol in [79] also cached the node velocity to restrict the flooding of node discovery packets to a smaller geographic region containing the destination node. Observing that node locations could similarly aid in routing, several location management schemes [33, 89, 105, 135] coupled with geographic routing were proposed for stateless routing in ad hoc networks. However, these schemes require accurate knowledge of locations (via the use of a GPS receiver, for instance), and the distributed management of accurate locations result in frequent location updates and protocol maintenance packets.

While the researchers in [72, 111] studied the effects of various realistic scenarios on the existing MANET routing protocols, the authors in [15, 29] suggested methods like a connected virtual backbone to help routing protocols adapt to node mobility. Moving one step further, researchers in [115, 123, 130] started to focus on mobility pattern/information awareness in routing protocols. Their method was to use such mobility pattern information obtained either via continuous location tracking, or micro-level mobility prediction to help make low level routing decisions such as the best choice of next hop to take. On the other hand, our work focusses on the macro-level mobility information for location approximation and routing at the hub level.

2.7 Routing within Intermittently Connected Mobile Ad Hoc Networks (ICMAN)

Dealing with temporally disconnected networks, the authors in [128] proposed an intelligent means of data dissemination called the Epidemic Routing protocol that relies on data buffering and node mobility to spread messages in the network. The so-called "summary vectors" were used for nodes to selectively exchange data packets, in order to limit the number of data transmissions. Similar work ([16, 53]) on data dissemination was also done for sensor and ad hoc networks. Routing in Intermittently Connected Networks (ICN) has also received significant interest from the research community recently. Several routing-related issues in ICN, also called Delay Tolerant Networks (DTN) were addressed in [67], which focused mainly on networks with known connectivity patterns, such as satellites with fixed paths, or busses with fixed routes. The authors developed several algorithms to analyze the knowledge to performance relationship in different protocols and demonstrated that their algorithms performed better with more network knowledge. However, for the availability of such global knowledge they assumed the presence of certain “knowledge oracles” that may not be applicable to most mobile ad hoc networks, where users’ mobility follow much less predictable schedules. Literature suggests similar study on how mobility (controlled or not) affects routing protocols and network performance (e.g., network capacity) in various types of ad hoc networks including sensor networks with mobile sinks (or base stations), and delay tolerant networks [28,54,118,138]. However, they did not deal with specific user mobility patterns. In [65, 74, 88, 121], the main focus was on the so-called “contact probability” of two users, which is oblivious to the specific locations (or “hubs”) they visit.

The concept of Epidemic Routing was extended upon by the authors in [93], where they proposed a probabilistic routing scheme whereby each node maintains the so-called “delivery predictability” to each known destination, and uses this metric to make routing decisions. However, their delivery predictability may decay with time, unlike our contact probability that remains constant throughout the entire simulation by virtue of the hub based mobility profile of nodes extracted from the underlying orbital mobility. In [101], the authors proposed a context-aware adaptive routing algorithm that takes into account the suitability of a node for carrying a message based on context information of the node at multiple dimensions. More recently, the authors in [84] suggested an algorithm that relies on vehicles to act as mobile routers, which connect disconnected sensor networks to a known destination.

The authors in [58] and [138] respectively studied the effects of controlled message flooding and controlled mobility in large scale ICN. Along the same lines of informed message passing that exploits the mobility information of network users to some extent, the authors in [28] proposed the introduction of autonomous agents that can adapt their motion within the network using multi-objective control methods to increase network efficiency. Although, the introduction of such agents with controlled/programmable motion simplifies the problem of routing, the ready availability/deployment of such agents within an ICN may not be too practical. To the best of our knowledge, our work is the first to explore the implication of the macro-level partially deterministic sociological orbits involving a list of hubs and its application to location approximation and routing in ICMAN, despite its practicality.

Chapter 3

Acquaintance Based Soft Location Management (ABSoLoM) in MANET

In this chapter, we concentrate on the effect of mobility in a MANET and explore possibilities of turning it to our advantage while making routing decisions. A major challenge faced in MANETs is locating devices for communication, especially with high node mobility and sparse node density. Present solutions provided by ad hoc routing protocols range from flooding the entire network with route requests, to deploying a separate location management scheme to maintain a device location database. It is our observation that even though a single node does not get to know of all other nodes, as nodes move around, their collective knowledge may serve as an important source of location information. Based on this, we propose a novel routing scheme called Acquaintance Based Soft Location Management (ABSoLoM) in MANET. In ABSoLoM, nodes make use of the real life concept of making acquaintances and keeping in touch with them regarding each others current locations. ABSoLoM has a twofold aim: to avoid the overhead of flooding; and to use a "soft" location management setup that does not require strict location management strategies and is thus computationally less expensive than standard "hard" location management schemes. Simulation results show that ABSoLoM not only outperforms existing flooding schemes (such as DSR [26] and LAR [79]) in terms of throughput, overhead and location discovery latency, but also achieves performance comparable to “hard” grid based location management schemes (like SLALoM [33]) with a much lower control overhead. Our proposed protocol is described in great detail below.

3.1 Introduction

A MANET is an infrastructure less group of wireless nodes that forward packets for one another. The hurdles in designing routing protocols for such networks are node mobility, limited node power, and restricted bandwidth of the wireless medium. Traditional routing protocols developed for wired networks like Distance Vector and Link State routing do not consider these limitations, and hence are unsuitable. Routing protocols for MANET can be broadly classified into two categories: proactive and reactive. Proactive protocols maintain an up-to-date image of the network topology at each node by broadcasting the neighborhood information of each node frequently throughout the network. In MANET where the network changes fast, such an effort is costly in terms of bandwidth and power consumption. This also leads to congestion and poor throughput. On the other hand, discovering and maintaining a route to the destination for every new session is a major challenge for reactive protocols. For example, Dynamic Source Routing (DSR) [26] uses pure flooding technique to locate destinations. Location Aided Routing (LAR) [79] tries to fair better than DSR by estimating the approximate region containing the destination (from some prior location information) and flooding only within that expected zone. However all such source routed protocols suffer significantly from path breakage caused by node mobility since there is a high probability of nodes in the path moving out of range of transmission of the preceding node. Repeated efforts in finding a path between a source and destination pair increases the overhead to maintain a single session.

Recently, a new family of protocols [33,89,105,135] has been proposed for MANET which makes use of location management and geographic forwarding for routing data packets, where nodes are aware of their own location via the use of a GPS receiver. When a source needs to communicate with a destination, it tries to find the geographical coordinates of the destination using a location service. Once the location is obtained, each packet is forwarded to one of the sources neighbors which is geographically closer to the destination than the source itself. The neighbor that receives the packet forwards it in a similar fashion till the packet finally reaches the destination. Geographic forwarding lends itself as an attractive candidate for routing in MANET, since the amount of network topology information that needs to be stored by each node is minimal and since the data packets need not store the entire route within them anymore. Moreover, since an end-to-end session does not depend on any specific route, the effect of link breakage on a session is greatly reduced. However, the challenge now lies in effectively maintaining a location database of all nodes in the network. SLALoM [33] is one such location management scheme that operates by partitioning the network terrain into well numbered grids. A hierarchy is then setup to have bigger regions comprising of smaller ones. All nodes are then assigned multiple home regions distributed evenly throughout the network such that a given nodes location is known to all nodes in that nodes home region. Each node is also responsible for updating its home regions with its current location information. Home regions that are near are updated more frequently than those that are far. Thus a source node just needs to query one of the home regions of the destination (that can be computed by the source locally) which is closest to the source, to get the destination location information. Deployment of such schemes in MANET requires a fair amount of node density to prevent the occurrence of empty home regions. In networks with highly mobile nodes, these schemes will incur considerable overhead in terms of location updates and maintenance of a distributed location database even with low traffic. Thus there is a wide gap between the low maintenance, low performance flooding based protocols and the high maintenance, high performance location management based protocols.

In this chapter, we propose a novel framework called Acquaintance Based Soft Location Management (ABSoLoM) that makes use of a simple heuristic to maintain a distributed location database for efficient geographic routing in MANET. A main difference of our scheme from the location management schemes mentioned above is that we make no assumptions regarding either the shape or size of the network terrain or the density of the nodes. By the aid of acquaintances, our scheme informally builds a "soft" location management setup and eliminates the need for strict management strategies (like computation of grids, assignment of location servers or home regions, location registration, etc). Our main idea is inspired by the experimental studies in social sciences by S. Milgram [99]. Unlike their study however, in our case, the source has no idea regarding the location of the destination, but it uses acquaintances to search for the destination with the hope of finding them being connected via a short list of common contacts, as observed by the studies mentioned above. Additionally, while other protocols (like DSR and LAR) struggle to cope with the challenges imposed by mobility, ABSoLoM uses the mobility of the nodes to its advantage for achieving better performance. Thus on one hand, ABSoLoM uses geographic forwarding to avoid the limitations of flooding protocols, and on the other hand due to the "soft" location services provided, it achieves performance comparable to location management schemes (like SLALoM) with a significantly lower control overhead. To the best of our knowledge, no other solution has been proposed in this capacity that tries to utilize this concept of acquaintanceship amongst nodes, to bring about a soft location management based routing protocol for wireless ad hoc networks.

The rest of this chapter is outlined as follows. In Section 3.2, we describe the details of the ABSoLoM protocol. In Section 3.3, we analyze ABSoLoM and compare it with DSR, LAR and SLALoM, and present our simulation results. Finally in Section 3.4, we conclude this work.

3.2 ABSoLoM Protocol Details

The basic idea of ABSoLoM is to make use of the real life concept of making acquaintances and keeping in touch with them regarding each other's current locations to bring about a "soft" location management system. When a source needs to obtain the location of a destination, it queries its acquaintances about it. If the destination is unknown to the acquaintances, they in turn ask their acquaintances and the process is repeated until the destination's location is discovered. Thus, ABSoLoM makes use of a location database which is maintained among the nodes in a distributed fashion. In the following subsections, we outline the different phases that constitute the ABSoLoM protocol.

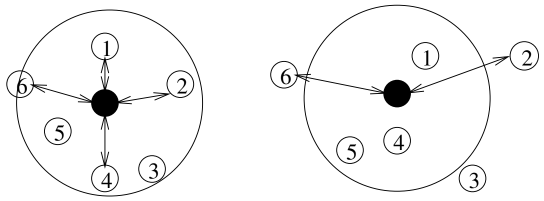
3.2.1 Acquaintance Selection

Each node periodically broadcasts its current location so that all nodes are aware of their neighbors and their respective locations. At start, each node randomly selects a neighbor who is not already an acquaintance, and sends an acquaintanceship request (*ACQ_REQ*). Simultaneously, it adds this neighbor in its acquaintance table as a *pending* entry. If the node does not get an acceptance from the *pending* acquaintance after a specific period of time, it deletes the entry from its acquaintance table, and retries the acquaintance selection process with a new neighbor. If the neighbor does accept the invitation (*ACQ_ACC*), the node marks that entry in its acquaintance table as *accepted*. As far as using acquaintances to locate a destination is concerned, a node does not distinguish between its *pending* and *accepted* acquaintances. The node however, sends location updates (*LOC_UPD*) containing its up-to-date location only to its *accepted* acquaintances.

We fix the maximum number of acquaintances that a node can have so that the memory requirements to keep track of acquaintances and the overhead to maintain acquaintanceship and discover destinations are practically manageable. Thus, any node that receives an acquaintanceship request replies with an acceptance only if it can accommodate a new acquaintance. In the event that a node receives an acquaintanceship request from a node to which it had send an acquaintanceship request previously, it marks the corresponding entry in its acquaintance table to *accepted*. In case of rejection, the node just ignores the request and lets the requesting node time out.

3.2.2 Acquaintanceship Termination

If a node finds any of its acquaintances to be in its neighborhood for a period of time greater than a timeout interval, it deletes that entry from its acquaintance table. Since both acquaintances will sense each other in their neighborhoods, the acquaintanceship will terminate mutually without the need for explicit termination messages. The timeout interval is chosen to be the time it takes for a node to move in and out of its acquaintance's transmission range at a speed equal to the average velocity of the nodes in the network. This method of initiating and maintaining acquaintanceship is illustrated in Figure 3.1. The intuition behind this form of acquaintanceship termination is for nodes to have acquaintances spread all over the network which, in turn, leads to faster discovery of a destination (see Section 3.2.5 for more detailed description).



(a) Acquaintance initiation (b) Acquaintance after timeout

Figure 3.1: (a)Initially make acquaintances with neighbors (b)After a timeout, only those nodes that move out of the neighborhood remain as acquaintances

3.2.3 Location Update

Each node checks its own location periodically, and on moving a distance equal to its transmission radius, sends out a location update (*LOC_UPD*) to each of its *accepted* acquaintances. Additionally, the node sends a location update to each node with whom it has an ongoing data connection. On getting location updates from its acquaintances, a node updates its acquaintance table with the location information contained within the updates.

3.2.4 Nosy Neighbors

As an optimization attempt, each node also maintains a location cache whose entries consist of the locations of the nodes (other than its acquaintances) this node learned of, while forwarding packets for other nodes in promiscuous mode. However, the information contained by only two types of control packets are relevant:

Location Update While forwarding a *LOC_UPD* packet, each node inserts (or updates) the location information contained within the packet into its cache. Additionally, if the node has room to make a new acquaintance, it adds a new entry for this node in its acquaintance table, and sends an acquaintanceship request to it. Intuitively, the cache allows for faster discovery of destination nodes during the discovery phase. To prevent nodes from using stale cache entries, entries from the cache are deleted after a timeout interval.

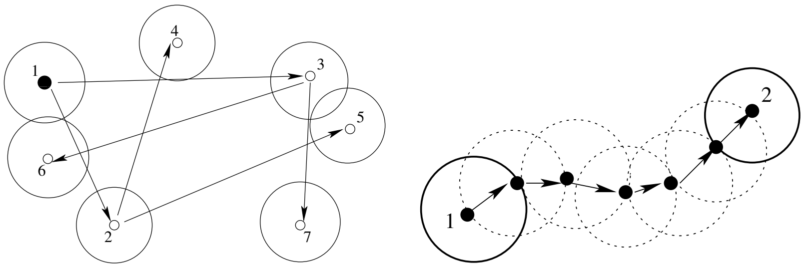
Location Response If a node knows about a queried destination, it geographically forwards a response (*LOC_RESP*) back to the source containing the location of the destination. This node may know of the destination for one of the following reasons:

- the destination is a neighbor or an acquaintance
- there is a live connection with the destination
- destination's information was in the location cache

An intermediate node while forwarding this *LOC_RESP* packet looks inside and may cache the location information of both the source and the destination.

3.2.5 Acquaintance Based Location Discovery and Data Transfer

At the start of communication, the source checks its neighbors, acquaintances, and location cache in that order for information regarding the destination. If there is no such information, it buffers the data packet and creates a location query (*LOC_QRY*) with a unique sequence number. It also specifies the maximum logical hop count in the query, which specifies the number of times a query packet can be forwarded by acquaintances to other acquaintances during the discovery process. The query is then geographically forwarded to all its *pending* and *accepted* acquaintances. Figure 3.2 illustrates the case where the maximum logical hop is 2. If the node has no acquaintances, it broadcasts the query to all its neighbors. This type of broadcasting in the absence of acquaintances is done only at the source node to enable a location query to be initiated properly. The neighbors in turn forward the query to all their acquaintances. In the absence of acquaintances, a neighbor (other than the source) will drop the query. Thus, ABSoLoM avoids full flooding of location query packets.



(a) Querying 2 hops away (b) 1 logical hop

Figure 3.2: (a) Node 1 queries 2 and 3 by one logical hop each. Node 2 queries 4 and 5, and node 3 queries 6 and 7 by one logical hop each (b) single logical hop from 1 to 2 by geographic forwarding (the larger circles in both figures, represent the transmission range of each node)

An intermediate node (nosy neighbor), while forwarding query packets from a node to its acquaintance, sniffs into the packet to see if it knows of the destination. If it does, it creates a *LOC_RESP* containing the location of the destination, and geographically forwards the response back to the source (as described earlier). When an acquaintance gets a *LOC_QRY*, if it knows of the destination, a *LOC_RESP* containing the location of the destination is sent back to the source. Otherwise, it decrements the logical hop count by unity, and if it is non-zero, geographically forwards the query to all its acquaintances. However, a node does not forward the query to an acquaintance if:

- that acquaintance forwarded this node this query
- that acquaintance is the source of this query

On receiving a *LOC_RESP*, the source retrieves all the buffered data for that destination and forwards them geographically to the destination. Duplicate location responses are ignored. It also adds the destination to a list of ongoing connections and appends its location information to each data packet that it forwards to the destination. The destination also adds the source to its list of live connections, and from then on, both nodes update each other of their current location periodically for the duration of the communication session. To further optimize this discovery process, a source while waiting for a *LOC_RESP* keeps a check to see if the destination itself entered the neighborhood. In that case all buffered packets are readily forwarded to that destination and all *LOC_RESP* packets for that particular query are ignored.

3.3 Performance Analysis

To evaluate the efficiency of the protocol, we implemented ABSoLoM in GloMoSim [137] and compared it with three protocols: DSR [26] (as it represents the most rudimentary of all source routed flooding techniques); LAR [79] (since it utilizes approximate destination location information to restrict flooding); and SLALoM [33] (since we found it to be a good representative of the grid based location management protocols). For additional performance comparison between various other existing MANET routing protocols (including the ones chosen in this paper) the readers are referred to: [39], [31], [87], [73], [106].

We used the standard distribution of DSR and LAR Scheme 1 provided with GloMoSim and implemented SLALoM as described in [33]. The simulation parameters were chosen as follows:

- simulation time = 2000s
- terrain size = 2000 m x 1000 m
- number of nodes = 100
- radio range = 350 m
  (a transmission covers 19.25% of the terrain)
- mobility = Random Waypoint Model
  (velocity from 0 m/s to 10 m/s with pause of 15s)

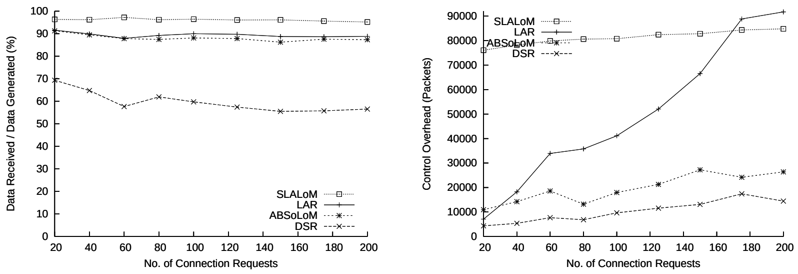
- traffic = Random CBR

  (packets of size 1 K ; 50 packets/connection; inter packet arrival time uniformly distributed between 1s to 3s)

We performed some simulations of ABSoLoM alone with the above parameters in an effort to see the effect of maximum number of acquaintances and the maximum number of logical hops on the throughput and control overhead. When the maximum number of acquaintances is greater than 3, we saw a dramatic increase in overhead whereas the throughput remains same. Similarly, allowing query packets to be forwarded for more than 2 logical hops increases the overhead unnecessarily without substantially increasing the throughput. So we empirically fixed the maximum number of acquaintances to be 3, and the query hop count to be 2 for the subsequent experiments.

Figure 3.3(a) shows the average throughput for each protocol as a function of traffic load. While both LAR and ABSoLoM achieve high and stable throughput with increasing traffic load, DSR performs poorly. This is mainly because DSR cannot deal with link failures as efficiently as LAR, whereas ABSoLoM is not affected much by link breaks due to geographic forwarding as discussed earlier. LAR is much more aggressive in searching for new routes when compared to DSR since LAR continually expands the estimated region of the destination and searches for a path to the destination in that region. SLALoM performs the best since it has the advantage of both geographic forwarding and the complete location database that allows the destination location to be quickly ascertained. ABSoLoM did not fare as well as SLALoM since it only offers a "soft" location management setup.

Figure 3.3(b) shows the impact of traffic load on the control overhead in each protocol measured in terms of the number of control packets received at all nodes. Control packets in DSR and LAR consist of "route requests", "route responses" and "route errors". In ABSoLoM we consider: periodic *HELLO*, *ACQ_REQ*, *ACQ_ACC*, *LOC_QRY*,

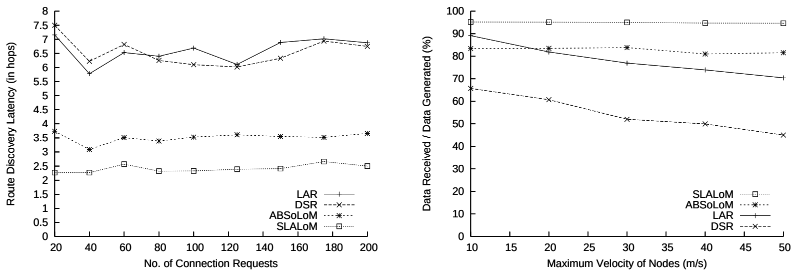


(a) (b)

Figure 3.3: (a) Data Throughput vs. Load (b) Control Overhead vs. Load

LOC_RESP and *LOC_UPD*. In SLALoM we have: periodic *HELLO*, *location queries, responses, updates, notifications*, and protocol *maintenance* packets that are needed to inform a node entering a new region, of all the nodes that have that region as their home. In Figure 3.3(b) we find DSR to have the lowest overhead since it is purely reactive in that it only needs to broadcast a "route request" at the start of each session, or if a source route fails. On the other hand, LAR has a very high control overhead which increases with traffic load. This is because in LAR, if a source had heard from a destination previously, it tries to estimate the region where the destination is expected to be found, and floods route requests in that area. Worse, if the request times out, this procedure is carried out repeatedly by increasing the size of the flooding zone which may grow to the entire terrain. This results in many redundant route requests. SLALoM has a high control overhead even for lower traffics just to maintain the location database. From the figure 3.3(b), it's clear that the overhead in ABSoLoM, while much lesser than that of LAR and SLALoM, is a bit more than that of DSR (note that ABSoLoM has a much better throughput than DSR).

Figure 3.4(a) shows the route discovery latency(in *physical* hops) in each protocol. This metric gives us an estimate of the time a source has to wait before it can initiate data transfer after the request for a session arrives. Since the route request has to reach



(a) (b)

Figure 3.4: (a) Route Discovery Latency vs. Load (b) Data Throughput vs. Node Mobility

the destination before a reply can be sent (in the absence of a cached location), route discovery latency is higher in both LAR and DSR on average. In ABSoLoM, since a *LOC_RESP* can be sent back to the source by an acquaintance (or an intermediate node), node discovery can be much faster, as evident from the figure. SLALoM performs best by virtue of its complete location database, where each node can query a destination's home region for its location. In our simulation scenario almost all queries in SLALoM were serviced successfully by the "near" home regions, thereby reducing the overall discovery latency considerably.

As a final test to the robustness of each protocol, we also simulated the performance by increasing the mobility of each node. We kept all other simulation parameters same as before but went on to increase only the maximum velocity of nodes in the Random Waypoint model from 10 m/s to 50 m/s and fixed the pause time to 0 s. The traffic was fixed to 200 connection requests. Figure 3.4(b) shows the throughput achieved by each protocol with varying mobility. Both LAR and DSR suffer from increased mobility because of increased number of source route failures, which considerably affect their throughput. On the other hand, ABSoLoM fares much better, since higher mobility enables a node to have acquaintance nodes spread all over the network, thereby ensuring efficient destination discovery. SLALoM is not much affected by mobility either, due to its dependence on geographic forwarding that shields it from link breaks. Overall, ABSoLoM emerges as a clear winner over LAR and DSR in terms of higher throughput, lower control overhead, lower route discovery latency and performs very well in highly mobile networks. At the same time ABSoLoM is a viable alternative to extensive location management schemes as it performs nearly as well with a significantly lower control overhead that would lead to lower power consumption and lesser congestion.

3.4 Summary

In this chapter, we have described a few popular schemes for routing in mobile ad hoc networks, and have showed that these schemes perform rather poorly in networks with high traffic volume and node mobility. Geographic routing has been shown to be a possible candidate for routing in such networks, and we have proposed a novel scheme known as Acquaintance Based Soft Location Management (ABSoLoM), in which nodes use the real life concept of making acquaintances and keeping in touch with them regarding their current locations, for efficient geographic routing. ABSoLoM, inspired by the "small-world phenomenon" concept, combines the advantages of purely reactive protocols with the concept of location databases to offer a cost effective solution that is intended to bridge the gap between the two extremes of flooding and complete location management schemes. The number of acquaintances that a node can have, and the number of query steps are the two parameters that need to be tuned for the efficient operation of ABSoLoM. By empirically selecting appropriate values for these parameters, we have shown that ABSoLoM not only outperforms flooding protocols like DSR and LAR in terms of throughput, discovery latency and control overhead, but also performs as well as "hard" location management schemes like SLALoM at a significantly lower control overhead.

Chapter 4

A Hierarchical Framework for Practical Mobility Modeling

In this chapter, we present our hierarchical mobility framework inspired by the sociological influence on the wireless users' movement. Mobility has been shown to affect protocol performance quite severely. In this light, the main challenge in effective protocol evaluation lies in the practical modeling of realistic mobility. Literature has already proposed a gamut of practical mobility models and frameworks to suit different real life scenarios. Our goal is to identify and extract mobility information that may come into use for efficient routing. Mobility can be bound by two observable extremes: *completely non-deterministic* - sensors dropped at random, moved randomly by blowing wind; and *completely deterministic* - Low Earth Orbiting (LEO) satellites that have a fixed scheduled motion. The more practical and commonly observed mobility however, lies somewhere in the middle of these two extremes and forms the center of our research interest.

4.1 Sociological Movement Pattern of Mobile Device Users

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. cubicles, 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 7.1. 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.

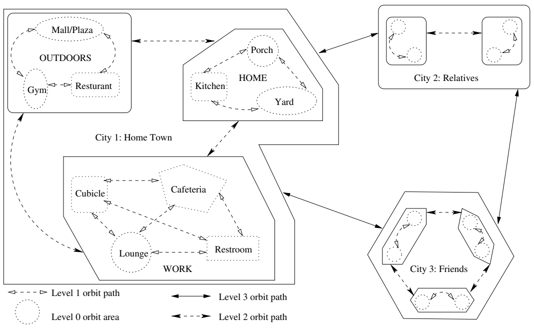


Figure 4.1: The sociological orbit

Interestingly, an “orbit” is one of the most natural form of motion observed in the microscopic world of molecules, and in the planetary universe as well. In fact, one may map certain characteristics of the natural orbits in these two extremes, to that of the sociological orbit described above.

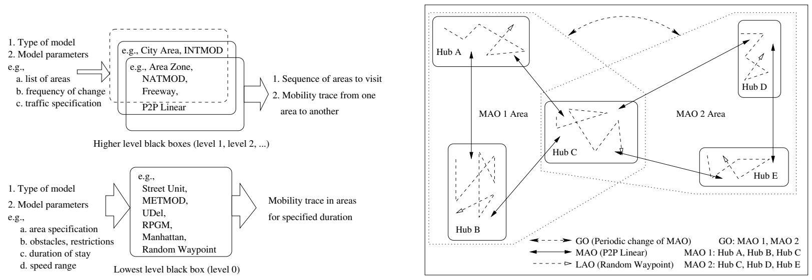
Electron Mobility Electrons of an atom orbit the atomic nucleus in different energy levels. Each atom is considered stable with a specific number of electrons in them. If this number increases or decreases, it becomes reactive, in which case electrons are exchanged in between oppositely charged atoms to attain stability. In our society, job opportunities and inexpensive accommodation may cause an influx of people into a city. This may slowly saturate the place, leading to a scarcity of jobs or a high cost of living which in turn prompts people to move out to other cities that offer better opportunities. In this way, over a long period of time, our 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. 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). Similarly, we find people moving within a small area (at home, work, etc.) for a few hours, forming small orbits at different parts of the day. A high level orbit, along which a person moves from one such area to another lasts for days or weeks. Over a period of months or years, a person may travel between cities along yet another higher level orbit. To the best of our knowledge, there exists no mobility model or framework that captures this hierarchical orbital movement pattern, despite its practicality.

The rest of this chapter is outlined as follows. In Section 4.2 we outline our mobility framework in greater details. In Section 4.3 we demonstrate the usefulness of our framework to generate practical mobility models and then analyze these models in Section 4.4. Finally we conclude our work in Section 4.5.

4.2 ORBIT Mobility Framework and Parameters

Keeping the sociological orbit in mind, we developed a mobility framework called ORBIT that incorporates the orbital movement pattern to easily generate practical models suiting various scenarios. The basic building blocks in ORBIT are the different levels of orbit. The versatility of ORBIT is in the fact that the mobility modeler in each orbital level may be viewed as a black box outputting node mobility traces, given inputs for that specific level. We model the lowest level orbit (level 0 in Figure 7.1) a bit differently than the higher level ones in terms of the output generated. More specifically, our black box for level 0 orbit takes as input a smallest area of interest and a given duration, and generates mobility traces within the area for that duration. In contrast, at the higher levels, the black box takes as input a list of areas of interest and generates a visiting sequence, as well as the mobility trace from one area to another. The exact nature of the traces generated by the black boxes at any level depend on the mobility model implemented within it, and hence is user defined. Figure 4.2(a) suggests a few of the existing mobility models that can be used in our framework at different hierarchical levels (references for these models are in Chapter 2 Section 2.4). For example, at level 0, we could use the Manhattan model to generate mobility traces within a city, and at level 1, apart from generating a sequence of cities to visit, we could use the Freeway model to generate inter-city movement. This is an additional strength of the proposed ORBIT framework, which can integrate all other models into a single domain providing practical models that are not only highly detailed, but are also capable of accommodating geographic hierarchies of cities and nations. Moreover, by appropriately fixing the area and duration of the lowest level 0 orbit, it is possible to simulate both small scale and large scale networks within the same framework.



(a) (b)

Figure 4.2: (a) The ORBIT framework (see Chapter 2 Section 2.4 for model references) (b) Our simplified ORBIT mobility framework

<u>**The Simplified ORBIT Framework:**</u> To facilitate our discussion, we consider a simplified ORBIT with three hierarchical orbital levels. It is worth noting that given the generality of the ORBIT framework as discussed earlier (and in Figure 4.2(a)), the following choices and assumptions made, serve only to simplify our quantitative analysis.

At the lowest level, we assume 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 [136]. 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 4.2(b), or may also remain disjoint as in Figure 7.1.

Simplified ORBIT Parameters: Considering the simplified ORBIT framework as an example, the parameters required to describe the framework could be divided into *3* sections, as depicted in Table 4.1.

Table 4.1: 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.

4.3 Orbit Based Mobility Models

In this section, we demonstrate the applicability of the models generated by our framework. We show that common mobility models like Random Waypoint [26] and Random Walk [64] can be trivially produced by our simplified framework by appropriately choosing values for our ORBIT parameters. In addition, we generate several new models as examples, which may be used to simulate realistic scenarios.

Random Waypoint and Random Walk If we let a single Hub cover the entire terrain, and set the LAO Timeout to the simulation time, all the nodes will follow Random Waypoint in the single Hub. On the other hand, the entire terrain can be tiled into Hubs in such a way that a node can go from one Hub center to any adjacent Hub center in a single simulation step i.e. unit time). By setting the LAO Timeout to zero (i.e. no pause), and selecting the visiting sequence to go through only adjacent Hubs that are either to the right of the current hub with probability *p*, or to the left with probability *1-p*, a Random Walk is generated. Figure 4.3 illustrates both these scenarios. Although these examples are trivial, they serve to illustrate the capacity of ORBIT to emulate existing mobility models that are commonly chosen for evaluating MANET protocols.

Random Orbit Among all the new models to be described, this is the most general. Figure 4.4 illustrates this Random Orbit model, where we assume each Hub to be a rectangular region with varying sizes. In each Hub, nodes move in an LAO with LAO Specific Parameters, and use the MAO Specific Parameters to travel from one Hub to

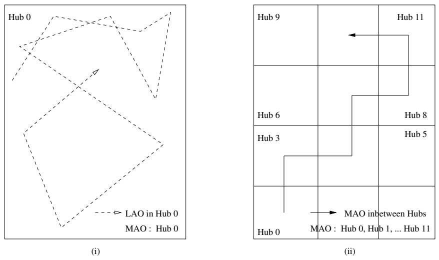


Figure 4.3: (i) Random Waypoint (ii) Random Walk

Table 4.2: 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	$< 13.89\ m/s$

another in the same MAO. On an MAO Timeout, a new set of Hubs are chosen to form a new MAO. While in an MAO, nodes visit their Hubs in a random sequence. 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. To assume realistic speeds we refer to the work done in [78, 107, 134], as summarized in Table 4.2. Accordingly, we fix our LAO parameters (i.e. Hub Speed (min, max)) to 1m/s and 10m/s, and the MAO parameters (i.e. Node Speed (min, max)) to faster speeds of 10m/s (23mph) and 30m/s (67mph).

Uniform Orbit This model is similar to Random Orbit, except for the setup of the Hubs. More specifically, unlike in the previous model, the entire terrain is divided into a grid of Hubs, with no Hubs overlapping with any other as seen in Figure 4.5(i).

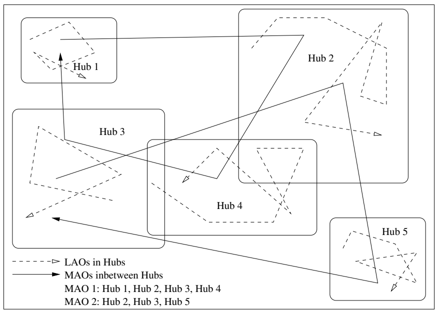


Figure 4.4: Random Orbit: City Model

Such a model may be used to simulate smaller scenarios like a School building, which is divided into a set of non overlapping classrooms. Students keep moving from one classroom to another along an MAO, and spend some time in each room along an LAO. The MAO might change after weeks/months along a GO.

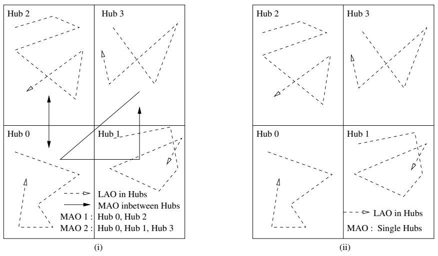


Figure 4.5: (i) Uniform Orbit: School (ii) Restricted Orbit: Office

Restricted Orbit This model is similar to Uniform Orbit, except that in this model, each MAO consists of a single Hub. In effect, an MAO is identical to an LAO, and there is no inter-Hub movement as shown in Figure 4.5(ii). This model is useful in simulating an office building scenario, that is made up of several non overlapping departments (or Hubs). Employees belonging to a particular department generally restrict themselves to a particular office space, thereby having no inter-Hub movements along the same MAO. Over time, their work might require them to shift to a different department, causing their

MAO to change along their GO.

Overlay Orbit In this model, we have the same setup as the Restricted Orbit. However, we overlay an extra Hub on top of the grid of Hubs, that spans the entire terrain. Movement in this model's MAO is also restricted to a single Hub. However, due to the overlaying nature of the extra Hub, nodes in this Hub move through all other smaller Hubs as seen in Figure 4.6. Such a model is useful for simulating exhibition/convention scenarios where a fixed number of organizers/presenters set their stalls up in small non-overlapping sections, while general people/attendees move across from one stall to another.

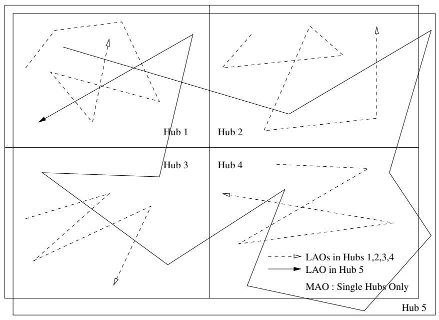


Figure 4.6: Overlay Orbit: Exhibition/Convention Area

4.4 Analysis of Model Characteristics

In this section, we analyze the models generated by the proposed ORBIT framework using a few of the protocol independent metrics defined in [7], which were shown to affect the basic building blocks of different routing protocols, thus accounting for the effect of mobility models on protocol performance. Our motivation for this analysis is to illustrate that our models generated as examples, differ with respect to these metrics when any ORBIT parameter is varied, providing multiple choices for modeling mobility in different scenarios. We choose to vary the number of Hubs in our study, since this also causes the Hub sizes to vary in all our grid based models (i.e. all except Random Orbit). Although our modified Random Waypoint model has a single Hub and yields a constant result when the number of Hubs varies, it is included as a reference point in our simulations. We perform our simulations in GloMoSim [137] with 100 nodes in a 1000 x 1000 sq. meter area for 1000 seconds. Each node is assumed to have a Radio Range of 250 meters. ORBIT parameters are used as follows.

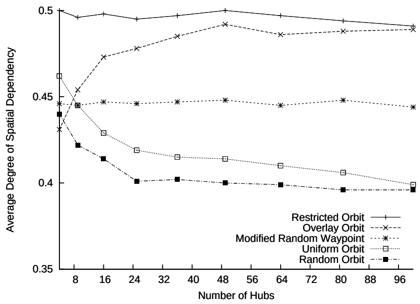
- Hub Size (min/max) = (150/250)m (Random Orbit)
- Hub Stay (min/max) = (50/100)s
- Global Pause (min/max) = (250/500)s
- Node Hubs (min/max) = (1/Total Hubs)
- Node Speed (min/max) = (10/30)m/s
- Hub Speed (min/max); Pause = (1/10)m/s; 1s

To analyze our models, we use the average degrees of spatial and temporal dependency as the mobility metrics, and the average number of link changes and link duration as the connectivity graph metrics. For a detailed description of the metrics, the reader is referred to [7]. No single metric serves as the determining factor in choosing one model over another.

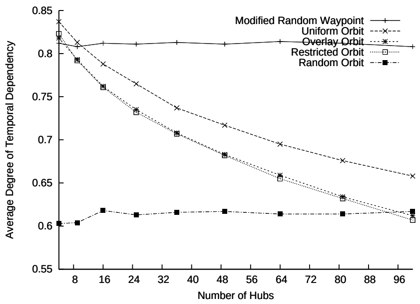
4.4.1 Mobility Metrics

For this particular metric, we consider only the average degrees of spatial and temporal dependencies.

Average Degree of Spatial Dependence Spatial Dependence indicates the similarity in the velocities of two nodes that are within a specific range from each other, which is chosen to be $2 * (RadioRange)$ in our simulations. Since Restricted Orbit and Overlay Orbit do not allow inter-Hub movements, nodes in the same Hub follow the same Hub parameters for a long time, resulting in a high spatial dependence as seen in Figure 4.7(a). Random Orbit and Uniform Orbit support inter-Hub movements and hence have a lower value for this metric, while our modified Random Waypoint has an intermediate value.



(a)



(b)

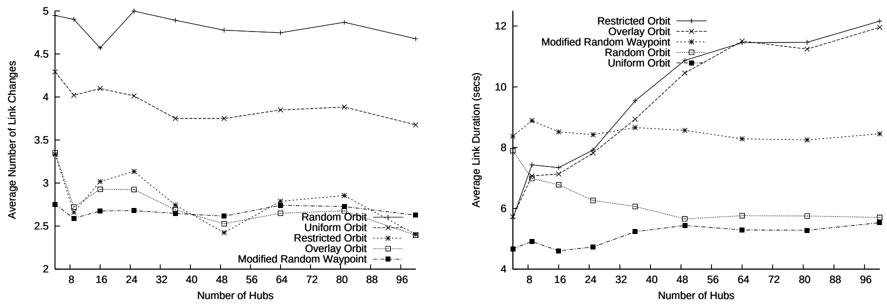
Figure 4.7: (a) Spatial Dependence vs. Number of Hubs (b) Temporal Dependence vs. Number of Hubs

Average Degree of Temporal Dependence Temporal Dependence indicates the similarity in the velocities of a node within a specific time interval, which was taken to be 20 seconds in our simulations. In Random Orbit, the Hub sizes are not affected by the number of Hubs, unlike in the other models, but the amount of overlap among Hubs increases sharply. Due to this overlapping, the inter-Hub movements end up being short (a node quickly reaches one Hub from another), similar to the intra-Hub movements causing frequent changes in mobility and leading to a low temporal dependence as seen in Figure 4.7(b). In our modified Random Waypoint, nodes have a single LAO where a slower speed change results in a high value for this metric. For the remaining models, a larger number of Hubs means a larger number of different Hub parameters that a node is subject to, leading to a steady decrease in the temporal dependence with an increase in the number of Hubs.

4.4.2 Connectivity Graph Metrics

For this particular metric, we consider only the average number of link changes and link duration.

Average Number of Link Changes This is the average number of times a link between two nodes (that ever existed during the entire simulation) comes up from being down. Since in Restricted Orbit and Overlay Orbit nodes are confined to particular Hubs, the probability of a link between two nodes in the same Hub breaking (and then coming up later) is small. In our modified Random Waypoint, links that would break when two nodes move away, have a low probability of coming up later as nodes move slowly in the entire terrain. In Random Orbit and Uniform Orbit, a link that is formed when two nodes visit a common Hub may break when one of them moves away, but has a high probability of coming up again when they meet later in the common Hub. In Random Orbit, these nodes may re-form the link even if they are in different but overlapping Hubs, thus showing the highest value for this metric in Figure 4.8(a).



(a) (b)

Figure 4.8: (a) Link Changes vs. Number of Hubs (b) Link Duration vs. Number of Hubs

Average Link Duration This is the average time a link between two nodes stays up. The restricting nature of Restricted Orbit and Overlay Orbit proves beneficial to link stability. Moreover, with an increase in the number of Hubs, the Hub sizes decrease causing the nodes to huddle even closer, increasing link duration as seen in Figure 4.8(b). The inter-Hub movements supported by Random Orbit and Uniform Orbit causes link breaks to happen more often, while our modified Random Waypoint shows an intermediate value.

4.5 Summary

In this chapter, we have focused on the effect of mobility on protocol performance and realized the importance of practical mobility modeling for realistic scenarios. Although several solutions are available, none identifies the orbital mobility pattern inherent within sociological movement of mobile device users. We have proposed the ORBIT framework that is capable of integrating multiple existing mobility models into a time and space based hierarchy, thereby creating much more practical mobility patterns.

Chapter 5

On Profiling Mobility of Wireless Users

In this chapter, we present a novel profiling technique capable of processing sporadic mobility trace data without any filtering and yet providing a leverage in performing location prediction with increased accuracy when compared to general statistical methods based on simple visit frequency to multiple locations. Profiling wireless users based on their mobility within wireless networks has recently been proven quite beneficial to several applications such as location prediction, resource allocation, etc. However, the suggested approaches to profiling have varied, as well as the details of the mobility information required to build the profiles. Authors in [85, 120], base their computational models on empirical data that are filtered to provide stable mobility data sets spanning regular intervals of consecutive days. However, in most real systems we observe lots of irregularity in terms of wireless users' usage pattern of the network, where a user may not be present in the network consistently for the entire period of observation.

We analyze a year long wireless network users' mobility trace data collected on the ETH Zurich campus that has many users appearing only for a few months, and in a very sporadic manner. Unlike earlier work in [32, 77, 127], we profile the wireless users based on sociological influence on their movement behavior and predict their locations. More specifically, we show that each network user regularly visits a list of places, such as a building (also referred to as "hubs") with some probability. The daily list of hubs, along with their corresponding visit probabilities, are referred to as a *mobility profile*. We also show that over a period of time (e.g., a week), a user may repeatedly follow a mixture of mobility profiles with certain probabilities associated with each of the profiles. Our analysis of the mobility trace data not only validate the existence of our so-called sociological orbits [50], but also demonstrate the advantages of exploiting it in performing *hub-level location predictions*. Our mobility profile based hub-level location prediction is shown not only to be more precise than a common statistical method, but also to incur a much lower overhead. We further illustrate the benefit of profiling users' mobility by discussing relevant work and suggesting applications in different types of wireless networks, including mobile ad hoc networks.

The rest of this chapter is outlined as follows. In Section 5.1 we redefine our modified sociological mobility framework and discuss different types of mobility trace data sets in Section 5.2. In Section 5.3 we present a study of hub-based parameters and analyze the user-based parameters in Section 5.4. In Section 5.5 we demonstrate the usefulness of mobility profiles in performing hub level location predictions and suggest other applications in Section 5.6. Finally, we conclude this chapter in Section 5.7.

5.1 Modified Sociological Orbit Framework

In this section, we briefly describe and enhance the sociological orbit framework we first proposed in [50]. In the real world, it is observed that users routinely spend a considerable amount of time at a few specific place(s), referred to as hub(s). For example, in a WLAN scenario a hub may be a floor within a building or, the entire building itself, depending on the scale of the network model. Although, it is hard (and may be even against privacy policies) to keep track of an individual at all times, one can still take advantage of the fact that most users' movements are within, and in between, a list of hubs.

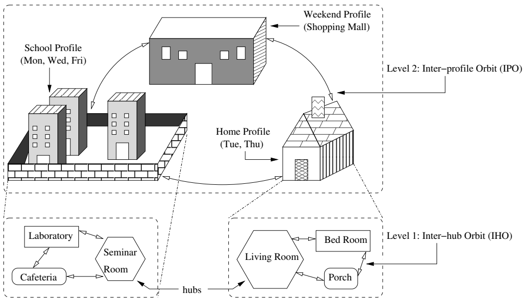


Figure 5.1: A hierarchical view of sociological orbits

Let us consider a graduate student who only has classes on Monday, Wednesday and Friday, when he is found on a school campus, spending most of his/her time in either his/her laboratory, a seminar room, or the cafeteria, each of which shall form a “hub” in this example (as shown in Figure 7.1). The actual list of hubs visited by the student on the same day is called a “hub list”. Even if such hub lists may vary from one day to another, that variation is only marginal (as shown later in Section 5.4). In most cases, a number of hub lists over a period of days may be clustered together and represented by a single “weighted hub list”, where the weight associated with each hub denotes the probability of the student visiting that hub within that period. In this work, we shall refer to such a weighted hub list to be a user’s “Mobility Profile”, and the movement in between the hubs within a profile as an “Inter-hub Orbit” (IHO). If one wishes to locate the student on a school day, knowing this *School Profile* within the campus shall be helpful, where one can most probably find him/her in either the laboratory, or the seminar room, or the cafeteria, without having to look all over the campus.

In real life, it is observed (and later verified from the analyzed data) that a user over long periods of time is usually associated with more than one mobility profile, mixed with certain probabilities. This is shown in Figure 7.1 as the *Weekend Profile* and the *Home Profile* to account for the student’s remainder of the week. Such a movement in between multiple profiles at a higher level is referred to as the “Inter-profile Orbit” (IPO). Over different periods of time, this mixture of profiles may change, causing what we call an “IPO Timeout”. The IPO and the IHO together constitute the hierarchical *sociological orbit* at two different levels. In this work, such orbital mobility information is shown to be most helpful in predicting the hub-level location of a user with much more accuracy than a general statistical method at a much lower overhead.

Table 5.1: Orbital Parameters

Category	Parameters
Hub-centric	Hub Form Hub Visits Hub Stay Time
User-centric	Mobility Profiles Hub List Size

To formalize the sociological orbit framework, we divide the orbital parameters into two categories: *Hub-centric*, and *User-centric*, as listed in Table 5.1. On the *hub-centric* side, the *Hub Form* depends on the actual definition of a hub in the network being modeled; *Hub Visits* denotes the number of users visiting a hub in a given period; and the *Hub Stay Time* is the amount of time a user spends at one stretch within a hub. On the *user-centric* side, the *Mobility Profile Parameters* include a list of hubs and their corresponding weights, and the *Hub List Size* refers to the number of unique hubs visited by a user on a day. Although, the study in [32] is similar to our study of the hub-centric parameters, to the best of our knowledge this work is the first to discuss and analyze the user-centric aspects related to sociological orbits.

5.2 Wireless Users’ Mobility Traces

In this section we briefly discuss the kinds of mobility traces currently available and the type of mobility information that favors our orbital analysis. Broadly, we may classify the method of obtaining mobility traces in the following 4 categories:

1. **AP-based traces collected by APs**: User/MAC ID, APs ID, times of events like association, disassociation, roaming etc. Examples include traces from Dartmouth [76] and ETH Zurich [127]. In order to translate AP-based traces to hub-level movement, we must aggregate the traces from multiple APs located in the same hub (e.g., a room or a building depending on the definition of the hub). This requires specific information regarding the location of each AP, but APs can be relocated or even decommissioned (in which case, its original location information is often lost, making its traces useless). An additional difficulty is that a user in a hub may be physically close to an AP in the hub but can be associated with another AP in a different hub, which sometimes also causes the so-called "ping-pong" effect. Furthermore, "missed poll" events and reliance on the users to turn on and off its device when entering and leaving a hub make it difficult to obtain a good estimate of the "hub staying time" or "inter-hub travel time", two measures of interest for building mobility profiles (and making intelligent routing decisions).

2. **AP-based traces collected by devices (PDAs, Laptops, etc.) carried by wireless users and uploaded to database servers**: Examples include traces from UCSD [98]. The specific data set in UCSD were from 257 undergraduate students carrying PDAs around for a period of 6 months. During this time, the PDAs not only recorded the APs they associated with, but also all the other APs that were visible at any time. Such user initiated trace collection is potentially more useful for studying users' hub-level mobility profiles. However, in this data set, a total of around 500 APs were visited by these PDAs, out of which only around 200 APs were identified to be within the UCSD network and could be located with exact geographical location coordinates. In addition, only 24 (out of 257) users

had traces that strictly involved the ~ 200 APs with known location coordinates. This makes the data set inadequate for the proposed study.

3. **User contact-based traces without location information**: Examples include experiments by Intel Research at Berkeley (HAGGLE) [65]. Such traces provide useful indications of available opportunities for ad hoc wireless networking. However, they are devoid of the information on the location where each volunteer were or where peers met with each other. As such, it is not suitable for any study that would require location information.

4. **User location-based traces collected by GPS or other devices carried by wireless users**. Ideally this type is most suitable since it allows us to observe and collect true sociological movement behavior of people who also use wireless networks from time to time. Unfortunately, such traces were not initially available, leading us to analyze AP-based traces instead as described below. In recent times, UMass Amherst has made such DTN traces available [27].

In the following sections we analyze wireless network users' mobility trace data collected on the ETH Zurich campus from 1st April, 2004 till 31st March, 2005. There were a total of $13,620$ users, 43 buildings, and 391 Access Points (AP). The data was obtained as system logs from the APs which recorded the *association, disassociation, missed polls*, and *roaming* events for users during the given period. Most users were seen to be present in the network for short durations in a sporadic manner. First, to study the observed distribution of the *hub-centric* parameters of the framework, we setup an Oracle database with these traces and employed standard SQL queries[1]. Second, to analyze the *user-centric* parameters we employ a clustering algorithm using a *Mixture of Bernoulli's*. We also develop efficient methods to model and analyze mobility profiles to validate the existence of sociological orbits. Finally, we use the mobility profiles to

[1] We thank Nirmal Thangaraj for developing the SQL queries

do hub-level location predictions more precisely than a statistical method, and at much lower overhead.

5.3 Hub-centric Parameter Distributions

In this section, we shall define and discuss the major parameters related to the concept of “hubs”.

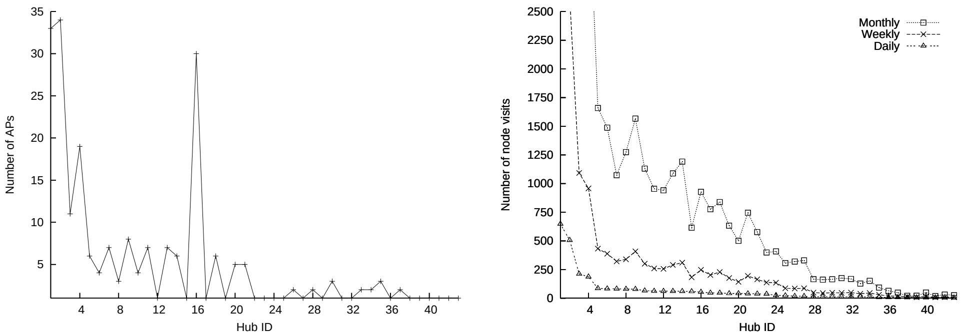
5.3.1 Hub Form

Hubs are generally defined as places of social importance. However, the hub form (e.g., size, shape) is mostly related to the type and scale of the network of interest. For instance, in a campus-wide wireless network (like the one we study in this work), one could consider each AP, or room, or floor, or the entire building to be a hub. Different choices are driven by the specific interest in the granularity of the movement of the users. For example, if one is only interested in identifying one building out of many (which say are located far apart on the same campus) to locate a user, and not the specific room or, the floor within the building (which say has a local network connecting all the rooms), then each building should be considered as a hub. As such, this decision also affects the time spent in each hub, and the total number of hubs in the model. A reasonable approach to decide upon a suitable hub form is to maximize the benefit of a high level orbital information, without having to monitor/update short-term or, short-range mobility information.

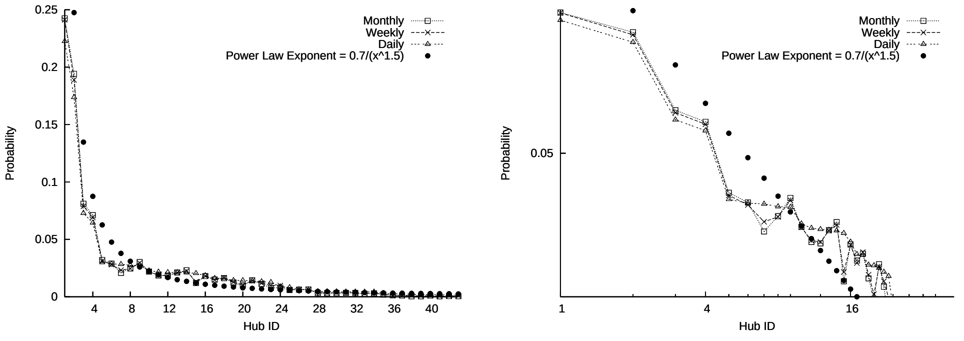
Note that in a broadly defined network of people, a hub is not required to have an AP (unlike in [32]), in which case a hub may still be identified in a variety of ways. The use of GPS service is the obvious first choice. Alternatively, in the broader contexts of *pervasive/ubiquitous computing* [2], and *Ambient Intelligence* (AmI) [112], localization in a cosmopolitan area will be even more readily available.

For the data we analyze here, each AP name includes the name of the associated building, which makes it possible to group the APs together and map the AP-based traces to hub-based movements where we consider each building within the entire ETH Zurich campus to be a hub. There are a total of 43 buildings/hubs that we analyze in the traces. We assign an unique "Hub ID" to each hub based on the decreasing order of daily hub visits (as to be discussed in the next Section 5.3.2). Some of these are academic buildings, libraries, cafeterias, etc. Each building has a number of floors, and each floor has multiple rooms, some of which have APs within them. Accordingly, a hub in our case may be covered by multiple APs.

5.3.2 Hub Visits



(a) AP Distribution within Hubs (b) Hub Visit Frequency



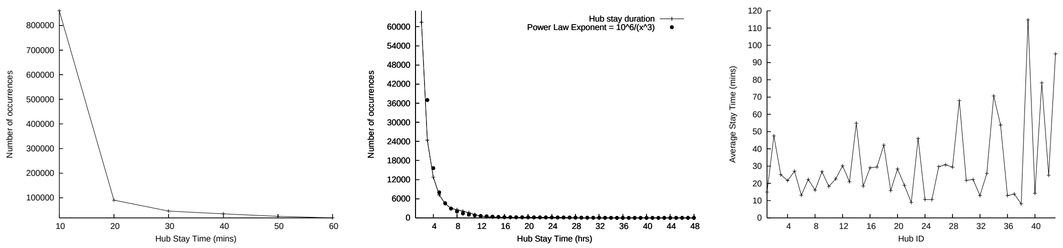
(c) Hub Visit Probability (d) Hub Visit Probability (log-log scale)

Figure 5.2: Hub visitations and AP distribution within hubs

In Figure 5.2(b), we plot the frequency with which network users visit each hub on a daily, weekly, and monthly basis to study the degree of social significance for each hub. As can be seen from Figure 5.2(b), only a few hubs (hub IDs 1 through 4) record large number of user visits, making them most "socially popular". We find the fraction of total node visits recorded by each hub to follow a power law distribution as shown in Figures 5.2(c) and 5.2(d). Also, the distributions for daily, weekly, and monthly visits are observed to be almost identical, indicating a temporal consistency of the social significance of each hub. For example, on either daily, weekly or, monthly basis, hub 2 is always seen to have a hub visitation probability of around 0.19. In Figure 5.2(a), we plot the number of APs in each hub. Interestingly, with the exception of hub 16, we observe the number of APs in each hub to be proportional to the number of its user visits. Larger number of APs could either indicate a larger size hub, or one may infer that there is a higher probability of recording AP-based system events (i.e., people turning on their network devices) in places with a stronger network (i.e., more number of APs). Whatever the cause and effect relation be, overall it is evident from Figure 5.2 that the number of popular hubs is *significantly smaller* than the total number of hubs, and such information may be useful for efficient information dissemination within a network by directing information to one of these popular hubs first.

5.3.3 Hub Stay Time

The Hub Stay Time parameter signifies the absolute time in between an *association* event, and either a *disassociation* or, *missed poll* or, a *roaming* event as recorded by an AP for a user. We measure the stay times in the granularity of 10 minutes and prune all values larger than 48 hours as they could indicate errors in the system logs. All other values are plotted in Figures 5.3(a) and 5.3(b). As seen in Figure 5.3(a), hub stay time of 10 minutes occurred most often, and there is a sharp drop and gradual decrease for 20 minutes or more. Figure 5.3(b) plots the hub stay time distribution at the hourly level, which is shown to follow a power law distribution. Most users are seen to stay in a hub for 4 hours or less, with the percentage of people staying in a hub for more than 12 hours being significantly lower. This fits in well with the diurnal cycle of mobility of casual network users (i.e., work/roam in day, and stay at home at night). Figure 5.3(c) displays the average hub stay time recorded by each hub. It is interesting to note that on an average, the hubs that recorded a lower number of user visits in Figure 5.2(c) seem to have the higher (*geq*60 minutes) average hub stay times. This may be due to the fact that while some hubs (e.g., cafeteria) record high number of shorter visits, others (e.g., library) may record lower number of longer visits. Such social influences on mobility patterns of users around hubs may be efficiently leveraged upon by applications such as routing.



(a) Hub Stay Time periods ≤ 1 Hour (b) Hub Stay Time periods ≤ 48 Hours (c) Average Hub Stay Time

Figure 5.3: Hub stay time distributions

5.4 Analysis of User-centric parameters

In this section, we shall analyze the *user-centric* parameters by examining individual network user's movement. To help select appropriate sample users, we first divide all the users in different user groups based on the number of days they are found to be “active” within the network (i.e., associated with at least one AP in the day). In Figure 5.4, we plot the fraction of total population vs. the number of their active days. The x-axis shows a range of values, i.e., 25 denotes up to 25 active days, 50 denotes anywhere between 26 and 50 active days, and so on. 80% of the total population is seen to be active for only 25 days or less in an year. Based on Figure 5.4, we categorize users in 6

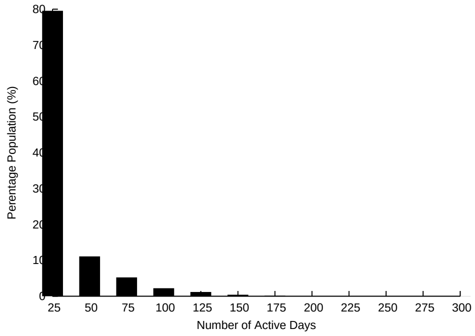


Figure 5.4: Number of active days for users

different groups (G_j) as follows: G_1 (Active for 0 to 25 days), G_2 (Active for 26 to 50 days), G_3 (Active for 51 to 75 days), G_4 (Active for 76 to 100 days), G_5 (Active for 101 to 125 days), G_6 (Active for 126 to 150 days).

We wish to choose one user to represent each group who is the “most active” within that group (i.e., we wish to maximize both the number of active days and the hub list size within each group). For a given group G_j, let $D^j_{\max}$ and $L^j_{\max}$ be the maximum number of active days and maximum average hub list size respectively, across all users in G_j. Let the pair D^j_i and L^j_i denote the number of active days and the average hub list size for a particular user i in G_j. Then, to represent group G_j we need to find a user who can *minimize*

$$\alpha\left(\frac{D^j_{\max} - D^j_i}{D^j_{\max}}\right) + \beta\left(\frac{L^j_{\max} - L^j_i}{L^j_{\max}}\right) \tag{5.1}$$

where, α and β are weights associated with each term. The results of using (5.1) with

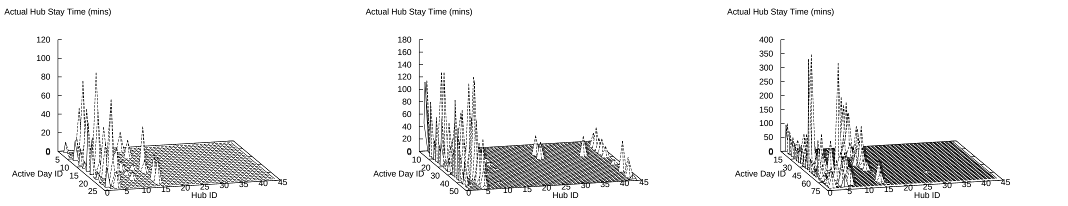
$\alpha = 0.5$ and $\beta = 0.5$ (we weigh both the number of active days and the hub list size equally) is summarized in Table 5.2. The basic intuition behind selecting the "most active" user from each group is the availability of more statistically significant mobility data for such an individual. At the same time, studying users from different groups help represent most of the total population as seen in Figure 5.4. Alternately, one may also select the sample users from each group to find users that are either "least active" or, "active on average".

Table 5.2: Sample users from all Groups

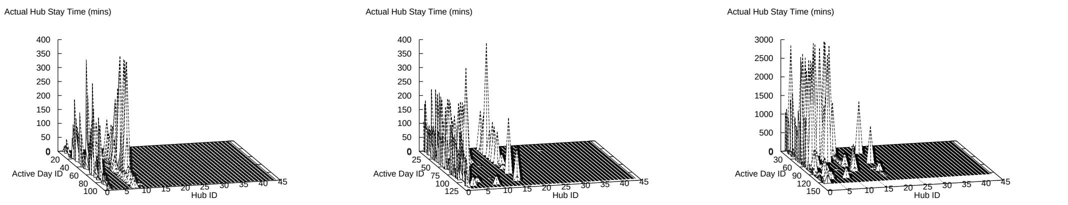
Group	MAC	D_i	L_i
G_1	0004.2396.92ab	24	2.29
G_2	0001.e30d.d737	49	3.27
G_3	0004.2398.82c0	71	4.08
G_4	0020.e089.9376	98	2.46
G_5	0004.2396.8ced	119	2.13
G_6	0005.4e41.cf1d	126	2.63

5.4.1 Model for Analysis of Mobility Profiles

We now present a study on the mobility of these 6 sample users. ***The key thing to note here is that none of these* 6 *users appear in the network for consecutive calendar days in their entire active period. So in the following, we only consider consecutive active day IDs, which may be far apart from each other with respect to actual calendar days. This is the inherent sporadic nature of our mobility data.*** We first plot their hub stay times in all the hubs during their active period, as shown in the 3-D Figure 5.5. To filter out noises (i.e., very brief hub stay durations), we run a horizontal plane parallel to the threshold value of 5 minutes across the plots shown in Figure 5.5 to obtain 2-D plots in Figure 5.6 showing only which hub(s) is(are) visited by a user (for more than 5 minutes) on a given active day.

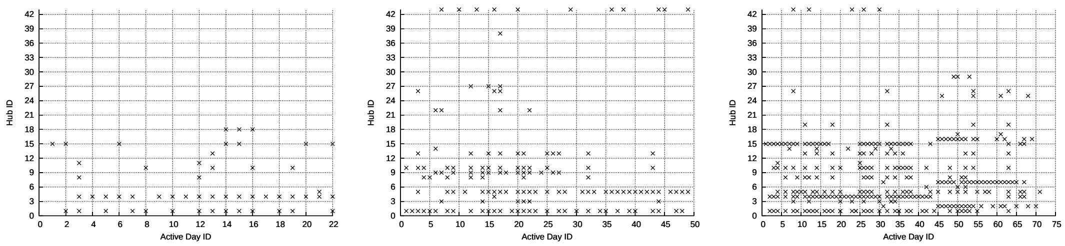


(a) Group 1 (0004.2396.92ab) (b) Group 2 (0001.e30d.d737) (c) Group 3 (0004.2398.82c0)

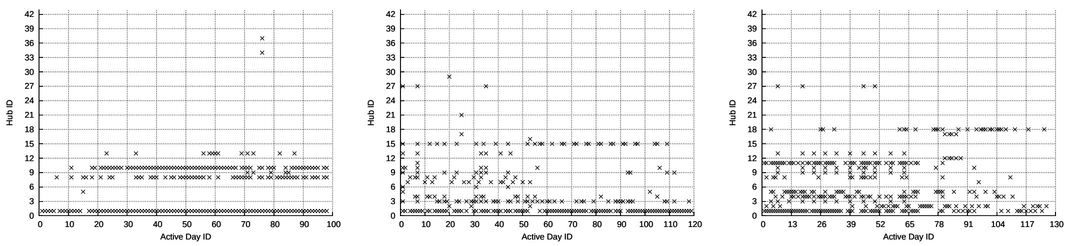


(d) Group 4 (0020.e089.9376) (e) Group 5 (0004.2396.8ced) (f) Group 6 (0005.4e41.cf1d)

Figure 5.5: Hub stay time distribution of all sample users



(a) Group 1 (0004.2396.92ab) (b) Group 2 (0001.e30d.d737) (c) Group 3 (0004.2398.82c0)



(d) Group 4 (0020.e089.9376) (e) Group 5 (0004.2396.8ced) (f) Group 6 (0005.4e41.cf1d)

Figure 5.6: Daily hub visitation patterns of all sample users

Next, we define an n dimensional space, where each dimension refers to a hub (i.e., $n = 43$ in our case). The hub list for a user in any given day (which is nothing but a binary vector of hub visits) may then be represented by a point in this space. For a particular user, similar hub lists on different days would generate several overlapping points whereas, two hub lists that differed only in terms of one or two hubs would generate points "close" to each other in this space. We use a clustering algorithm that helps define this concept of "closeness" by considering hub lists that say only differ in a maximum of 1 or, 2 hubs to be "close" and to belong to the same cluster. The mean of the cluster, which is a weighted hub list, then represents a mobility profile, as is described in more detail below.

5.4.2 Using a Mixture of Bernoulli's to Profile Mobility

A suitable choice to model the binary hub visitation vectors is a *Mixture of Bernoulli* distribution. In this mixture model there shall be more than one mixture component where, each component is considered an unique mobility profile represented by the component mean. Thus, a profile is nothing but a distribution over the hub visitation probabilities (i.e., a weighted hub list). We refrained from using the commonly used *Mixture of Gaussian* model because the domain of the Gaussian variable, being $(-\infty,\infty)$, is clearly not suitable for binary valued vectors. Assuming that the current mobility profile of a user is known, we model each hub visitation by a user as an independent event. On the other hand, if the current profile is not known, the general probability of a user visiting a hub is dependent on the probability associated with each mobility profile. The latter fact is crucial, since it allows for the knowledge of a user's hub visits to help infer the current mobility profile and therefore the probabilities of visits to other hubs on the same and future days, as shown later in Section 5.5.

More formally, we use h $(1 \leq h \leq H)$ to denote the unique hub id and i $(1 \leq i \leq n)$ to denote the day index, where H and n are the total number of hubs and days respectively. On each day i, we define a user's hub list to be a binary vector of hub associations

$\mathbf{y}^{(i)} = [\mathbf{y}_1^{(i)}, \ldots, \mathbf{y}_H^{(i)}]$ where each element $\mathbf{y}_h^{(i)} \in \{0,1\}$ such that $\mathbf{y}_h^{(i)}$ is equal to 1 if hub h was visited on day i, and zero otherwise. We denote the complete trace of hub visits across all n days with the symbol Y, which is the collection $Y = \{\mathbf{y}^{(1)}, \ldots, \mathbf{y}^{(n)}\}$. The total probability of Y is given by the product of a mixture of independent Bernoulli distributions as follows:

$$p(Y) = \prod_{i=1}^{n} p(\mathbf{y}^{(i)}) \ ,$$

where,

$$p(\mathbf{y}^{(i)}) = \sum_{j=1}^{k} p(j) \prod_{h=1}^{H} p(\mathbf{y}_h^{(i)} | \rho_{j,h}) \ .$$

Here, k is the number of mixture components (or, mobility profiles); $p(j)$ is the probability of following profile j; $\rho_{j,h}$ is the probability of visiting hub h on a day when following profile j. This framework is a generative Bayesian model in the sense that it defines a probability to every possible outcome, or pattern, that can be produced for Y.

This mixture model is trained using the Expectation-Maximization (EM) algorithm of Dempster, Laird and Rubin [40]. By employing consecutive Expectation (E)- and Maximization (M)- steps, the probability of the entire data set Y is guaranteed to monotonically increase (or, remain the same). The E-step consists of computing the posterior probability of membership of a datum (or, hub list) across the k mixture components (or, mobility profiles). Intuitively, at this E-step we look at each hub list and try to guess the mobility profile being followed on that particular day. Formally, this corresponds to computing the *responsibilities* of each component in the mixture, denoted by $r_j^{(i)}$, such that $\sum_{j=1}^{k} r_j^{(i)} = 1$, and are found using Bayes' theorem:

$$\textbf{E-step} \qquad r_j^{(i)} \equiv p(j|\mathbf{y}^{(i)}) = \frac{p(j)p(\mathbf{y}^{(i)}|j)}{p(\mathbf{y}^{(i)})} \tag{5.2}$$

$$\forall i = 1, \ldots, n \quad \text{and} \quad \forall j = 1, \ldots, k \ .$$

The M-step of the EM algorithm updates the parameters of each of the k components of the mixture model, in light of the responsibilities $r_j^{(i)}$ computed in the E-step. In other words, at this M-step we look at the probabilistic associations of the hub lists with each profile computed in the E-step, and update both the probabilities associated with each profile (i.e., mixing proportions), and the probabilities associated with each hub visitation within a profile. Thus, formally the parameters of the mixture model are: the mixing proportions, denoted by vector $\boldsymbol{\pi} = (\pi_1, \ldots, \pi_k)$ where $\pi_k = p(k)$ such that $\sum_{j=1}^{k} \pi_j = 1$; and for each mixture component j, there is a vector of dimension H of probabilities of each hub being used, denoted by $\boldsymbol{\rho}^{(j)} = (\rho_{j,1}, \ldots, \rho_{j,H})$. Thus each component in the mixture represents a mode of a user's interaction with a subset of the H hubs available (i.e., each profile is nothing but a weighted hub list). The updates to the parameters in the M-step are as follows:

$$\textbf{M-step}, \boldsymbol{\pi} \qquad \pi_j = \frac{1}{n} \sum_{i=1}^{n} r_j^{(i)} \qquad (5.3)$$

$$\forall j = 1, \ldots, k \; .$$

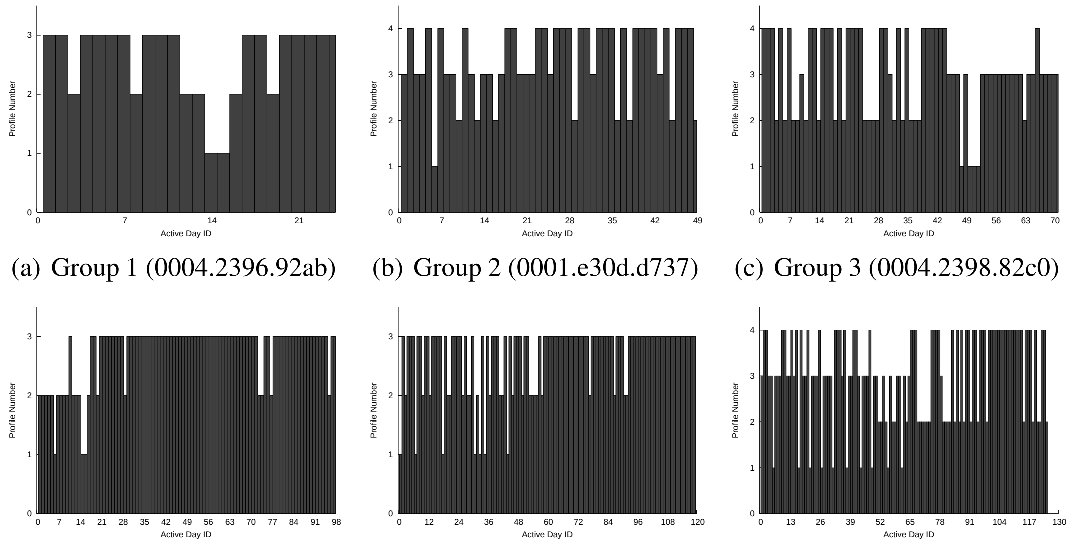
and

$$\textbf{M-step}, \boldsymbol{\rho} \qquad \rho_{j,h} = \frac{\sum_{i=1}^{n} r_j^{(i)} \mathbf{y}_h^{(i)}}{\sum_{i=1}^{n} r_j^{(i)}} \qquad (5.4)$$

$$\forall j = 1, \ldots, k \quad \text{and} \quad \forall h = 1, \ldots, H \; .$$

In this study, for each user we choose the number of components k (i.e, profiles) for each mixture model by visual inspection of the data distribution and initialized the *mixing proportions* and *component means* at random such that each profile has moderate associativity with hub lists. An alternate approach may include approximate Bayesian model selection techniques, e.g. via the Bayesian Information Criterion (BIC; [117])

or, other criteria. Figure 5.7 shows the pattern of mobility profiles over all the days. Table 5.3 lists both the probability that a user is in a given profile, and the probability that a hub is visited when following a particular profile. As an example, from Figure 5.7(a) we find that the sample user from group G_1 is following his/her mobility profile 1 on day 14. From Table 5.3 we see that given profile 1 for that user, the hub visitation probabilities indicate *definite* visits to hubs $1, 4, 15$ and 18 on day 14, which may then be verified from his/her actual hub list distribution shown in Figure 5.6(a).



(a) Group 1 (0004.2396.92ab) (b) Group 2 (0001.e30d.d737) (c) Group 3 (0004.2398.82c0)

(d) Group 4 (0020.e089.9376) (e) Group 5 (0004.2396.8ced) (f) Group 6 (0005.4e41.cf1d)

Figure 5.7: Daily distribution of mobility profiles

5.4.3 Overcoming Shortcomings of our Clustering Algorithm

Mixture of Bernoulli's distribution is able to cluster hub lists into mobility profiles, thereby enabling us to efficiently represent apparently chaotic AP system logs as probabilistic mixture of mobility profiles. In the next section, such profiles are even shown to be effective in making more accurate location predictions. However, this technique

Table 5.3: Mobility Profile Parameters

Group 1 Profiles $j(\boldsymbol{\pi}_j)$	Hub ID h (Hub Visitation Probability $\rho_{j,h}$)
1 (0.08)	1(1.0), 4(1.0), 15(1.0), 18(1.0)
2 (0.31)	1(1.0), 4(0.83), 8(0.27), 10(0.54), 11(0.27), 13(0.13), 18(0.13)
3 (0.61)	1(0.38), 4(0.81), 5(0.07), 15(0.34)
Group 2 Profiles $j(\boldsymbol{\pi}_j)$	**Hub ID h (Hub Visitation Probability $\rho_{j,h}$)**
1 (0.02)	1(1.0), 9(1.0), 14(1.0)
2 (0.14)	1(0.14), 4(0.14), 5(0.49), 43(1.0)
3 (0.31)	1(1.0), 3(0.27), 5(0.8), 8(0.4), 9(0.53), 10(1.0), 13(0.6), 26(0.13), 27(0.2), 38(0.07), 43(0.07)
4 (0.53)	1(0.54), 3(0.08), 5(0.68), 8(0.04), 9(0.19), 13(0.08), 43(0.11)
Group 3 Profiles $j(\boldsymbol{\pi}_j)$	**Hub ID h (Hub Visitation Probability $\rho_{j,h}$)**
1 (0.06)	1(1.0), 2(1.0), 5(0.75), 6(0.5), 7(0.75), 8(0.75), 10(0.5), 13(0.25), 16(1.0), 17(0.25), 29(0.25)
2 (0.25)	1(1.0), 3(0.22), 4(1.0), 5(1.0), 7(0.06), 8(0.66), 10(0.94), 11(0.06), 13(0.28), 15(0.89), 19(0.22), 26(0.11), 43(0.11)
3 (0.32)	1(0.09), 2(0.62), 4(0.05), 5(0.28), 7(0.75), 10(0.04), 15(0.05), 16(0.44), 17(0.04), 19(0.04), 25(0.18), 26(0.04), 29(0.09)
4 (0.37)	1(0.53), 3(0.12), 4(0.83), 5(0.18), 6(0.04), 8(0.04), 10(0.08), 14(0.19), 15(0.6), 43(0.11)
Group 4 Profiles $j(\boldsymbol{\pi}_j)$	**Hub ID h (Hub Visitation Probability $\rho_{j,h}$)**
1 (0.03)	5(0.33), 8(1.0)
2 (0.17)	1(1.0)
3 (0.80)	1(1.0), 8(0.65), 10(0.9)
Group 5 Profiles $j(\boldsymbol{\pi}_j)$	**Hub ID h (Hub Visitation Probability $\rho_{j,h}$)**
1 (0.06)	1(1.0), 3(1.0), 4(0.51), 5(0.14), 6(0.43), 8(0.29), 9(0.85), 10(0.58), 11(0.14), 13(0.58), 25(0.43)
2 (0.20)	1(0.14), 2(0.12), 3(0.22), 4(0.04), 7(0.54), 10(0.04), 15(0.26), 16(0.04), 30(0.04)
3 (0.74)	1(1.0), 3(0.36), 4(0.12), 5(0.01), 8(0.09), 9(0.06), 10(0.02), 15(0.23), 17(0.01), 21(0.01)
Group 6 Profiles $j(\boldsymbol{\pi}_j)$	**Hub ID h (Hub Visitation Probability $\rho_{j,h}$)**
1 (0.08)	1(1.0), 3(1.0), 4(0.3), 5(0.4), 8(0.2), 9(1.0), 10(1.0), 11(0.9), 13(1.0), 18(0.1), 27(0.4)
2 (0.21)	1(0.03), 2(0.92), 12(0.19), 17(0.15)
3 (0.26)	1(0.87), 2(0.22), 4(0.7), 5(0.13), 8(0.31), 10(0.03), 11(1.0), 18(0.11)
4 (0.45)	1(0.56), 2(0.03), 3(0.045), 4(0.17), 5(0.35), 8(0.02), 10(0.05), 11(0.11), 18(0.36)

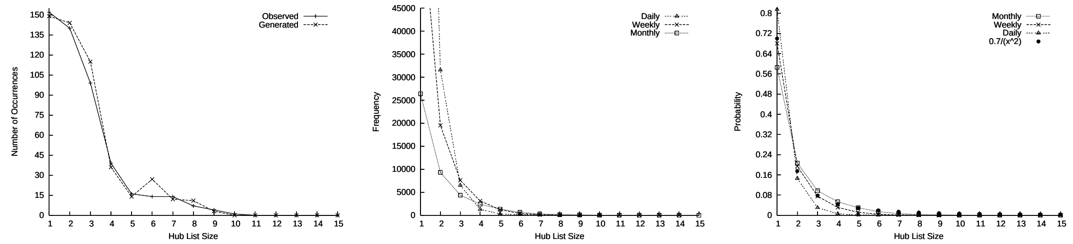
is still not sufficient for identifying a change in current profile given a sequence of hub visit observations. For example, if we are given the current profile being followed, and subsequently also informed about the hubs that a user visits day after day, this method will not be able to indicate if a change in the current profile has occurred. For this purpose, we may need to explore other time series analysis tools.

Even in its current form, our clustering algorithm treats a hub "visit" and a hub "non-visit" as equal. This may not be desirable in the practical sense. For example, let us consider 4 binary hub visitation vectors for a system with 6 hubs (i.e., each vector will have 6 bits with “1” for a visit and a “0” for a non-visit): $h_1 =$ “$110000''$, $h_2 =$ “$100000''$, $h_3 =$ “$111111''$, $h_4 =$ “$111110''$. If we were to compare the “closeness” of h_1 with h_2 and that of h_3 with h_4, our clustering algorithm will label both pairs to be 1 bit apart from each other. However, in reality the first pair $\{h_1, h_2\}$ have only 1 hub in common, whereas the second pair $\{h_3, h_4\}$ have 5 hubs in common. So practically, we are more interested in the closeness of the latter pair. To enable our clustering algorithm to differentiate between these two cases, we may need to introduce some bias towards a “1” and away from a “0”. We shall explore the possibility of constructing a Mixture of Bayesian Bernoulli’s distribution for this purpose.

5.4.4 Hub List Size Distribution

The results in Table 5.3 may seem to indicate that several users tend to visit many hubs in any given day as their mobility profiles include multiple hubs. Hence, to study the distribution of the hub list sizes of our sample users we generate daily hub lists for each of them over their individual activation period based on their mobility profiles. More specifically, for each day we first choose one of their possible profiles at random following the mixing proportions, and then generate visits to each hub individually following the hub visitation probabilities in that chosen profile. We then obtain the aggregated (i.e., across all sample users) Hub List Size distribution and compare it with the actual distribution observed in the trace data. As seen in Figure 5.8(a), both the observed and the generated hub list sizes are distributed almost identically, and shorter (≤ 3) hub list sizes occur most often.

For a more comprehensive study, we also present the hub list size distribution observed for all the nodes in the analyzed data. Figure 5.8(b) shows the overall number



(a) Observed vs. Generated Hub List Size (b) Overall Hub List Size Frequency (c) Overall Hub List Size Probabilities

Figure 5.8: Hub list size distributions

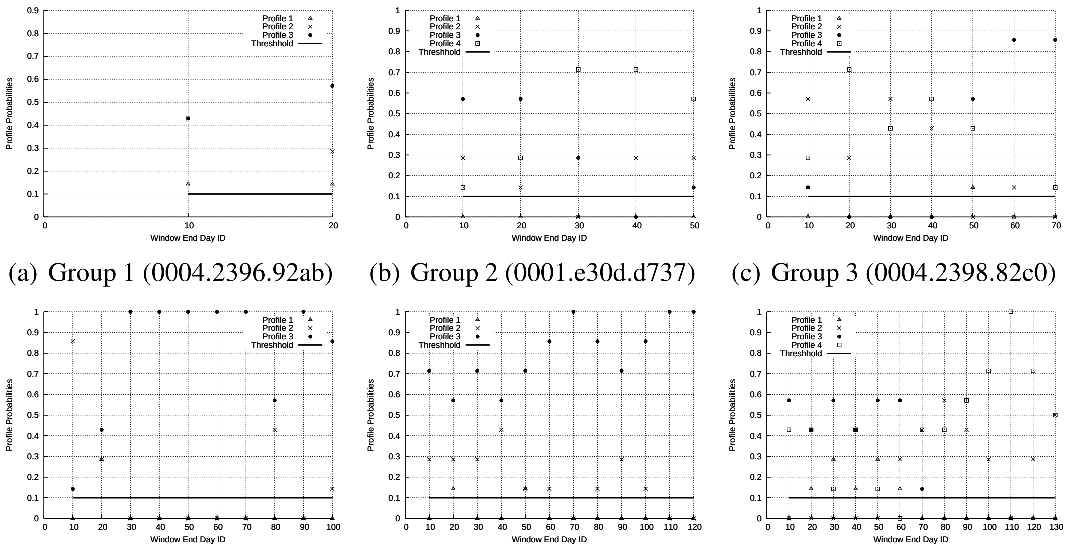
of occurrences of different hub list sizes on a daily, weekly, and monthly basis. In these results below, each hub list size denotes the *unique* number of hubs a user visits in a day, week, or month. As seen in the figure, for all the time scales (i.e., daily, weekly, monthly), shorter hub lists sizes (≤ 3) occur most often, following which there is a sharp decrease for hub list sizes of 4 or more. Figure 5.8(c) shows the fractional occurrences of the observed hub list sizes, which are almost identical across the daily, weekly, and monthly values, and may be approximated as shown with a power law distribution. Based on these results, we may infer that the length of each hub list is not only *significantly smaller* than the actual number of hubs present in the system, but is also not dependent on the period for which the user is active. Once more, such sociological movement behavior can prove beneficial to several mobile wireless systems and applications.

5.5 Mobility Profile based Location Predictions

In this section, we highlight another important contribution of our work by showing how the mobility profiles may be useful in making hub-level location predictions with more accuracy than a general statistical method and at a lower overhead. More specifically, we first show an efficient way to apply the clustering algorithm described in Section 5.4.2 and identify the right mixture of mobility profiles with lower overhead than compared to that in a statistical method. We then focus on two types of profile based predictions: *Unconditional Prediction*, where given the hub visit information over a window of n days, we wish to predict the hub visit patterns for the next window of n days; *Conditional Prediction*, where given that we can identify the current mobility profile of a user (based on available information about a hub a user either visited, or plans to visit), we wish to find the probability of that user visiting another hub in that same day.

5.5.1 Determining a Mixture of Mobility Profiles

From Figure 5.7, it becomes lucid that the seemingly random movements of a user as seen from Figure 5.6 can now be systematically described via a mixture of mobility profiles over a period of time. However, since this mixture will eventually change, we still need an efficient method to identify the right mixture of profiles describing the user's movement pattern over a given period. ***Since the active days of any user are not consecutive calendar days, we could choose any arbitrary window size of past active day IDs to predict future locations.*** However, if the window size is small, we would expect higher accuracy with higher frequency of profile updates. On the other hand, larger window would mean less overhead at the cost of possibly lower prediction accuracy. In our experiments we empirically chose the window size to be 10 active day IDs. Thus one may use the mobility traces of hub visits collected over 10 days (our window size) to determine the possible mobility profiles and their corresponding mixing proportions using the *Mixture of Bernoulli's* described in Section 5.4.2. It is then possible to identify the appropriate mixture to include all the profiles with a corresponding mixing proportion greater than some specified threshold (which is 10% in our case, as shown in Figure 5.9). One may then choose to only consider this specific mixture for the next 10 days, when the next mixture update is performed. Considering the sample user from group G_5 as an example, we find that a mixture update (due to a significant change in the mixture from one window to the next) is required on days $20, 50, 60, 70, 80$, and 110, which amounts to only 6 updates for a 120 days activation period. Later in this section, we show that even with such infrequent updates (i.e., low overhead) our mobility profiles are able to predict daily hub-level locations with more accuracy than a common statistical method, which in contrast would require a hub visitation information update *every day*.



(a) Group 1 (0004.2396.92ab) (b) Group 2 (0001.e30d.d737) (c) Group 3 (0004.2398.82c0)

(d) Group 4 (0020.e089.9376) (e) Group 5 (0004.2396.8ced) (f) Group 6 (0005.4e41.cf1d)

Figure 5.9: Windows of 10 days to find mixture of mobility profiles

5.5.2 Unconditional Prediction

In this part, we study the accuracy of the unconditional profile based hub-level location prediction, and compare it with that made from statistical observation alone. We again consider only the sample users.

Statistical based prediction: In this method, we assume no knowledge of mobility profiles and hence no clustering by the mixture of Bernoulli's distribution is performed. One simply collects the mobility traces of a sample user for n days, and then based on the hub visit frequencies determines the user's hub visit probabilities. This probability distribution can then be used for hub visit predictions and may be compared with the observed hub lists to compute the Statistical based Prediction Error (SPE) rate as

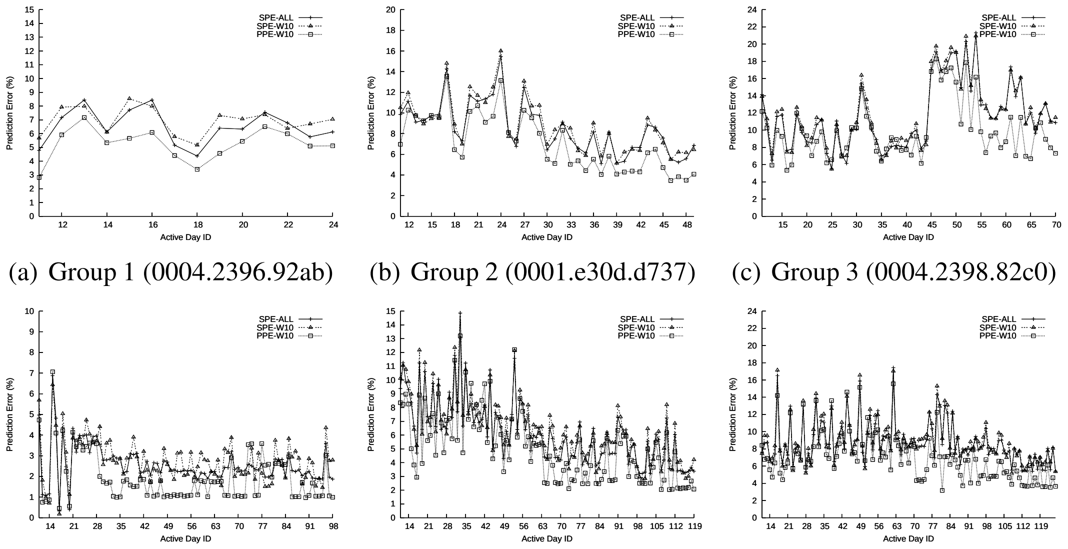
$$SPE = \frac{Incorrect\ number\ of\ hub\ predictions}{Total\ number\ of\ hubs} \tag{5.5}$$

We consider two variations of this statistical approach. In the first one, prediction for day $n+1$ is done based on the history of past n days. After day $n+1$, the sample user's hub visit probabilities are recomputed based on the past $n+1$ days and then used to predict the hub list for day $n+2$, and so on till the end of the activation period for the sample user is reached. The error rate for this approach will be referred to as ***SPE-ALL*** from now on.

In the second version, the past history of a window of size n days (e.g., days 1 till n) is considered to predict the hub list for not only day $n+1$ but for the entire next window of n days (e.g., days $n+1$ till $2*n$). After this the learning window shifts over the days $n+1$ till $2*n$ to predict hub lists for days $2*n+1$ till $3*n$, and so on till the end of the activation period for the sample user is reached. The error rate for this approach will be referred to as ***SPE-W10*** from now on as a window size of 10 is empirically chosen for our given data set.

Profile based prediction: This approach assumes knowledge of mobility profiles. One initially collects a sample user's mobility traces for a window of size n days (e.g., days 1 till n), and then applies the clustering algorithm described in Section 5.4.2 to find out a mixture of mobility profiles and their associated probabilities (as was shown in Figure 5.9). Based on this profile information, one not only predicts the hub list for the next day (i.e., day $n+1$) but also for the entire next window of n days (i.e., day $n+1$ till $2*n$, like the second version of the statistical approach). To be more precise, for each day, one first randomly chooses one mobility profile out of the mixture of profiles based on their mixing proportions and then predicts the day's hub list based on that chosen profile. For each day within the window, one compares the hub visit predictions with the observed hub visit values to compute the daily Profile based Prediction Error (PPE) rate similar to that shown for SPE in (5.5). Since an empirical value of 10 is chosen for the window size, this error rate is referred to as ***PPE-W10*** from now on.

After the next window of n days, one re-computes the mixture of mobility profiles based only on the hub visit information of the last n days and uses the new mixture information to predict the hub lists for days $2n+1$ till $3n$, and so on till the end of the user's activation period is reached. *Note that the number of re-computations required for each sample user in the profile based method is thus only* $1/n$ *times that in the first version of the statistical method, while it is equal to that in the second version.*



(a) Group 1 (0004.2396.92ab) (b) Group 2 (0001.e30d.d737) (c) Group 3 (0004.2398.82c0)

(d) Group 4 (0020.e089.9376) (e) Group 5 (0004.2396.8ced) (f) Group 6 (0005.4e41.cf1d)

Figure 5.10: Unconditional prediction comparisons

In our experiment, we choose a window size of $n = 10$ to be consistent with the results shown in Figure 5.9. In Figure 5.10, we plot the percentage values for SPE-ALL, SPE-W10, and PPE-W10. Each error value shown is an average over 1000 runs. As seen in Figure 5.10, PPE-W10 has lower values than both SPE-ALL and SPE-W10 in almost all cases. The few cases where PPE-W10 has higher values is mostly attributed to a substantial change in the mixture of mobility profiles, where the old and new set of profiles had very different hub associations. To quantify the improvement in location prediction achieved by our profile based method over that by the statistical methods, we define

$$Prediction\ Improvement\ Ratio(PIR) = \frac{SPE - PPE}{SPE}$$

where PIR-ALL indicates improvement of PPE-W10 over SPE-ALL, and PIR-W10 indicates the improvement of PPE-W10 over SPE-W10 and present its distribution parameters in Table 5.4. As seen, the mean values (considering the standard errors) are all positive, indicating a much better overall performance of our profile based hub-level location predictions as compared to the statistical approach with similar (when compared to SPE-W10) or much better (when compared to SPE-ALL) cost of location updates. This is one of the most critical contributions of our concept of profiling mobility based on sociological orbits.

Table 5.4: The Distribution of PIR (%)

Group	Mean ± Standard Error	
	PIR-ALL	PIR-W10
G_1	20.6 ± 2.3	24.3 ± 3.2
G_2	18.9 ± 2.0	21.4 ± 2.1
G_3	12.9 ± 2.0	14.5 ± 1.9
G_4	27.2 ± 3.0	27.9 ± 3.8
G_5	21.5 ± 1.4	24.6 ± 1.6
G_6	21.2 ± 1.5	22.6 ± 1.6

5.5.3 Conditional Prediction

In this section, we show how the current mobility profile information may improve the performance of certain hub-level predictions. The authors in [32, 35] have shown that a common statistical approach (similar to the one described in Section 5.5.2) is capable of keeping track of a user's visits to different locations (via the system logs on APs). Consequently, it is possible to provide a probabilistic view of finding the user in any location at any time based on the past history of that user's hub visits. *Let us assume that this mobility behavior for all our sample users repeats itself the next year, such that their future visits in the next activation period may be validated by the data present.* Taking the user from group G_2 as an example, we find that he/she visited hub 43 on 11 days in a 49 day activation period within the year the data was collected, as seen in Figure 5.6(b). If we were to consider that this mobility pattern over 49 days is going to repeat itself the next year following our assumption above, then the general probability of finding that user in hub 43 during his next activity period would be $\frac{11}{49} = 0.22$. From within our profile based framework, we not only are capable of providing similar general information but, given the current mobility profile, also can be much more specific. For instance, given the same example and assumption as above, the general statistical probability $P(h)$ of finding the user in hub h on any given day may be calculated equivalently through our approach as

$$P(h) = \sum_{j=1}^{k} \boldsymbol{\pi}_j * \rho_{j,h} \tag{5.6}$$

Using (5.6) and the data in Table 5.3, the general probability of finding the user from group G_2 in say "target hub" $H_t = 43$ on day $D = 16$ of his next period of activity would be given as: $(0.02) * (0.00) + (0.14) * (1.0) + (0.31) * (0.07) + (0.53) * (0.11) = 0.22$ (which is the same as that noted before). However, as soon as the user ventures into say "identifier hub" $H_i = 4$ on day 16 (see Figure 5.6(b)), our method shall identify the current profile (P_{now}) to be 2, as it is the only one with hub 4 in it. With this additional knowledge, our approach would then be able to re-compute the probability of finding the user in hub 43 on day 16 to be $\rho_{2,43} = 1$. From Figure 5.6(b), we see that under our assumption of repeated period of activation, the user would indeed visit hub 43 on day 16 of his next activation period (i.e. $\mathbf{y}_{43}^{(16)} = 1$), which makes our profile based prediction more precise. Several similar cases for each user type are listed in Table 5.5, where we find that the conditional probability ρ_{j,H_t} (obtained based on mobility profiles) is closer to the actual event $\mathbf{y}_{H_t}^{(D)}$ than the general probability $P(H_t)$ (obtained from the common statistical approach). In particular, as seen in the cases for the users from groups G_2, G_3, G_4 and G_6 our predictions would be completely accurate, whereas those from the statistical method are far from correct.

Table 5.5: Conditional Prediction Comparison

Group	H_t	D	$P(H_t)$	H_i	P_{now}	ρ_{j,H_t}	$\mathbf{y}_{H_t}^{(D)}$
G_1	11	13	**0.08**	8	2	**0.27**	**1**
G_2	43	16	**0.22**	4	2	**1**	**1**
G_3	7	7	**0.3**	14	4	**0**	**0**
G_4	1	15	**0.97**	5	1	**0**	**0**
G_5	7	53	**0.11**	2	2	**0.54**	**1**
G_6	3	63	**0.1**	9	1	**1**	**1**

Essentially, the mobility profiles help us group the hubs in separate (but, potentially overlapping) sets of hubs on the basis of visits occurring to them within the same period of time (i.e., following some mobility pattern), unlike in the statistical method where all the hubs are treated independently and identically. Note that in practice it may not always be possible to uniquely identify the current mobility profile based on the hubs visited so far (i.e., *identifier hubs*), as one hub could belong to 2 (out of say 4) profiles. However, as shown earlier in Section 5.5.2, as long as the *identifier hub* is able to suggest a proper subset (or, a mixture) of the user's mobility profiles for a given period, we are able to predict hub visits more precisely than a common statistical method and at a lower overhead.

5.5.4 Hub Transitional Probability based Predictions

So far we have only considered the binary hub visitation probabilities each day in our mobility profiles. For a more in depth study, we compute and use the probabilities for transition from one hub to another (i.e. outgoing transitional probability) for each of our sample users in predicting the sequence of hub visitations each day. We chose 3 levels of information for performance comparisons as described below.

Statistic based transitional probability: In this method, we assume no knowledge about mobility profiles. For each user a simple statistic of outgoing transitions is maintained for each hub in that chosen user's union of hub lists, and later used in calculating the transitional probabilities from each hub of that user. The entire activation period for each user is considered.

Profile based transitional probability: In this method, mobility profiles are considered and the daily profile distribution is assumed to be known. Within each profile of a user, a separate statistic is maintained for each hub ever visited by the user following that profile. This information is used to compute the outgoing transitional probabilities specific to each profile for a user for the entire activation period.

Profile based temporal transitional probability: This method is similar to the *Profile based transitional probability* except for that we not only assume knowledge about profile distributions, but also assume that the time information associated with transitions is available. Accordingly, we divide the entire day in zones of 3 hours [2] (e.g. 12:00 am – 2:59 am, 3:00 am – 5:59 am, and so on) to get 8 distinct time zones. Following this, for each profile of each user, we compute outgoing transitional probabilities

[2] This choice for time zone width was chosen empirically for our specific data set.

specific to each time zone for that profile of that user. Once more the entire activation period for each user is considered.

For example, consider the transitions in between 3 hubs A, B, and C. Lets say we are only interested in the transition probabilities from hub A to hubs B and C. Let us also consider that there are 2 profiles, $P1$ and $P2$, and also within each profile, we consider 2 time zones $T1$ and $T2$. Lets say Table 5.6 lists the example transition frequencies in all the possible cases.

Table 5.6: Example Hub Transition Frequencies

Hub A to Hub B			Hub A to Hub C		
Profile	Zone	Count	Profile	Zone	Count
$P1$	$T1$	25	$P1$	$T1$	150
$P1$	$T2$	5	$P1$	$T2$	75
$P2$	$T1$	150	$P2$	$T1$	5
$P2$	$T2$	75	$P2$	$T2$	25

Under these assumptions, lets say we are given that the user is in hub A on a given day, and is following profile $P1$ and is about to move out in time zone $T2$. Now we wish to predict the probability of finding the user going to either hub B or hub C following the different schemes. As expected, the statistical method is the most general as it does not have any concept of either profiles or time zones. It just computes the total number of transitions to both hubs B and C and computes the transitional probabilities, which come out to be equal in this case. The profile based method is more specific, as it only considers the transitions made under the specific profile $P1$ and fairs better than the statistical method. The temporal scheme having the most knowledge only looks at transitions made under profile $P1$ in time zone $T2$ and is expected to predict with the most accuracy. We list the transitional probabilities that will be computed by all the above 3 schemes for this specific example in Table 5.7.

In order to compare the performances of these methods, for each day of activation of each user we use each of these methods to predict the entire sequence of hubs to be

Table 5.7: Next Hub Prediction Example

Next Hub	Statistical	Profile	Temporal
B	50%	11.76%	6.25%
C	50%	88.23%	93.75%

visited by the user on that day. At first we assume that only the starting hub and its corresponding time zone is known each day. More specifically for the *Profile based temporal transitional probability* method, whenever we predict a *next hub*, we follow it with the selection of a Hub Stay Time duration from the power law distribution as shown in Figure 5.3(b) to compute the time zone for the next prediction. We then compare the predictions with the actual hub visitation sequence and compute the prediction accuracy as

$$Prediction\,Accuracy = 1 - \frac{incorrect\ predictions}{total\ predictions}$$

to get *Statistic based Prediction Accuracy (SPA), Profile based Prediction Accuracy (PPA),* and *Time based Prediction Accuracy (TPA)*. Each particular prediction result shown is an average over 1000 predictions. ***It is worth noting that the known hub does not need to be the starting hub of a day. This result can be generalized for any situation where the current hub (any time of day) is available and the hub visitation sequence for the rest of the day needs to be predicted.***

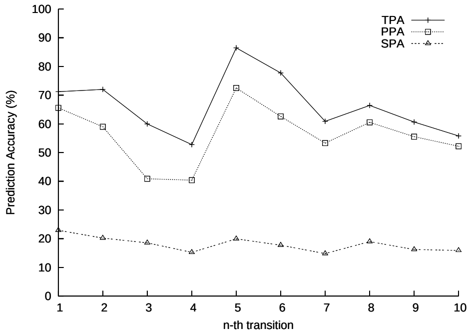
Additionally, we also compare the performances when we correct the predictions made at each hub in the sequence for each day to essentially limit the possibilities of a cascading error effect. As seen from the results in Table 5.8, prediction accuracy following the temporal transitional probability (TPA) performs the best having assumed the most level of mobility knowledge, followed closely by PPA, while SPA fairs the worst having no knowledge about mobility profiles.

Table 5.8: Prediction Accuracy for Hub Visitation Sequence

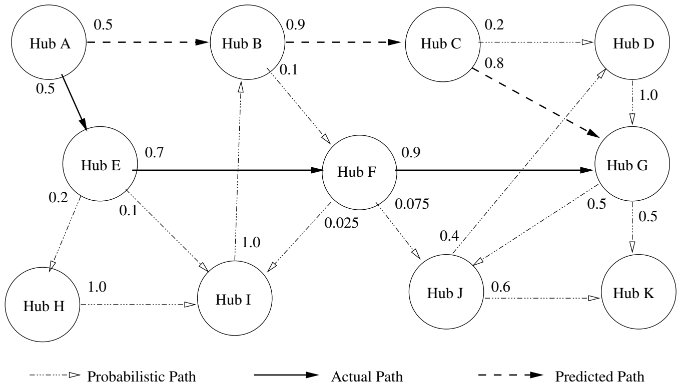
Group	Only starting hub known			Current hub always known		
	SPA	PPA	TPA	SPA	PPA	TPA
G_1	21.89%	45.93%	59.05%	25.59%	64.13%	74.48%
G_2	21.70%	35.62%	49.26%	24.90%	52.79%	65.89%
G_3	13.18%	25.83%	39.70%	24.50%	63.00%	71.05%
G_4	31.32%	53.82%	68.56%	31.62%	61.07%	73.97%
G_5	18.40%	58.84%	73.18%	19.95%	64.65%	75.88%
G_6	24.25%	46.11%	57.67%	28.33%	49.70%	61.97%

5.5.5 Prediction Accuracy vs. Transition Sequence

Given that only the starting hub (and the corresponding time zone) information is available, we wished to study the dependency of the prediction accuracy on the predictions farther away from the known hub, which on a intuitive note is expected to monotonically decrease. Interestingly however, we observed that due to the manner in which our transitional probabilities are obtained, the accuracy need not decrease steadily as shown in Figure 5.11(a) with the sample user from Group G_1 as an example. In order to explain this, let us consider Figure 5.11(b) as an example where a user travels along the path: Hub A $\Rightarrow$ Hub E $\Rightarrow$ Hub F $\Rightarrow$ Hub G. The outgoing transitional probabilities are shown along the edges. The starting Hub A is known. Let us assume that we make an incorrect prediction at start and predict the next hub to be Hub B instead of Hub E. In the next iteration, assuming Hub B to be the current hub let us say we once again incorrectly predict the next hub to be Hub C following the 90% transitional probability from Hub B to Hub C. Now although we made 2 successive incorrect predictions, assuming Hub C to be current, we have much higher probability (80%) of predicting the next hub to be Hub G than Hub D. In reality, the current hub at this point is Hub F and from there on too, there is a much higher probability (90%) of going to Hub G than any other, which say causes the next prediction to be accurate. Thus, no matter how many incorrect predictions we make in a sequence of hub visitations, we have the possibility to get back on track at any point, as our outgoing transitional probabilities from one hub is independent of those from another. Thus there is no fixed trend for a cascading error effect once an incorrect prediction is made somewhere in the hub visitation sequence.



(a) Group 1 (0004.2396.92ab) prediction accuracy



(b) Predictions for successive transitions

Figure 5.11: Example to show the independence of prediction accuracy with transitions

5.6 Other Applications of Mobility Profiles

In this section, we note that our sociological orbit aware mobility profiles can also be useful in other network aspects such as improving realistic mobility modeling, QoS routing, and resource allocation. The main focus of our ongoing work is to understand, define and develop potential applications that can benefit from the sociological orbital movement patterns. Example applications may be those that need to "push" a large amount of data (such as critical software updates, on-line movie rental, etc.) directly to a mobile user's laptop and/or PDA. In such cases, if the server is located far from the user and/or is busy when the user is connected, downloading may incur substantial delay. Knowing the orbital pattern, however, enables the server to send partial content while the user is connected (e.g., at work), and then predictively send the rest to a proxy server close to where the user will likely be (e.g., home). When the user goes online, the rest of the content can be downloaded quickly from the nearby proxy server seamlessly. Similarly, applications requiring a small amount of data (e.g., event alerts) to be sent or real-time audio/video or images from surveillance cameras to be streamed to concerned individuals (e.g., security personnel) will also benefit. Specifically, knowing the orbital pattern will enable an application to use simulcast to reduce the delivery time without flooding the message, that would otherwise cause severe congestion in the network and possibly even compromise security. Conceptually, this generalizes the simple techniques used to download files and stream multimedia to vehicles on a highway equipped with roadside base stations that are spaced a few miles apart from each other, whose mobility patterns may easily be modeled following the work in [36] for example.

Specifically for security monitoring, mobility profiles can be of great value. For example, in places like military campuses, or highly classifies research facilities, where there are a lot of authorized personnel with different levels of security clearances, it may be overwhelming to manually monitor the movement of each and every individual at all times. Alternatively, if everyone is wearing an access badge with an in-built sensor, it is possible to obtain and store their routine mobility profile, and automatically trigger alarms based on variations from that profile. Authors in [57] had profiled mobility sequences based on location coordinates for reducing false alarms in anomaly-based intrusion detection.

The advantage of using the knowledge of sociological orbits can also be extended to applications for health-care and environmental monitoring using mobile ad hoc networking techniques. An example application is to monitor air (or water) quality in a place without a networking infrastructure, and its impact on the health of the people who live or work there. A similar application is to detect and control the spread of a flu virus [132]. In these applications, people can wear tiny sensors with limited transmission range. During any given day, only a few people (called “carriers”) may travel to a site as a part of their social routine, where an access point is present, for uploading the sensor data and downloading control messages. A majority of others will have to send data to a person sharing the same “hub” as a part of its orbital pattern, and that person in turn will forward the data to another person until one of these “carriers” is reached. After the collected data is processed at a remote center, it is possible that certain symptoms are detected but definitive diagnosis is still not possible. In such cases, additionally, more intensive data collection may be needed only at selected locations and/or by selected persons. Knowing the orbital patterns of the persons will certainly help target the right subset of people and thereby reducing unnecessarily flooding of the request for data collection, and also cutting down the energy/bandwidth in collecting uninterested data.

Sociological orbits also has numerous applications for the study of extreme events either natural or man-made disasters. Knowledge of individuals’ patterns of mobility will prove useful in efficiently and effectively communicating accurate information about the event to those people who are most directly affected. For example, if there was a chemical spill on a highway, alternative routes may be suggested to those with a trajectory leading them towards the spill. This information can be communicated directly to their personal computers, PDAs, or intelligently enabled vehicles. This will allow those individuals to pro-actively alter their routes so as to avoid the event. Likewise, this information need not be directly communicated to those unaffected, thus avoiding panic or attracting unrelated individuals to observe the event.

5.7 Summary

In this chapter, we have analyzed the year-long mobility trace data of 13,620 WLAN users collected on the campus of ETH Zurich with 391 Access Points (APs) where most users do not stay present in the network consistently over a regular period of consecutive calendar days. We not only validated the so-called sociological orbits exhibited by mobile wireless users, but also profiled the user movements to help in location prediction even in the presence of the sporadic mobility data. Unlike previous work on analyzing similar mobility trace data which focus on AP-centric parameters, our focus was on hub-centric parameters (where a hub is a place of social interest and thus can be served by several APs) such as hub staying time, and in particular, user centric-parameters such as the number of hubs visited by a user in a day (or hub list size) and mobility profiles (i.e., a probabilistic list of hub visitations) of a user.

We have noted that it should not be surprising that sociological orbits do exist in users' mobility patterns. Nevertheless, one of the contributions of this work is that for the first time, it has been found from analyzing the traces that a user does exhibit sociological orbits at multiple levels on different time scales: a hub-level orbit consisting of a number of hubs during a day, and a mobility profile-level orbit consisting of a few profiles during a week or longer period. In addition, the results of our analysis will enable researchers to model users' mobility in more realistic way using the appropriate values and distributions of user-centric parameters we have identified via analysis.

Another, perhaps a more important contribution is that, although intuitively, it is beneficial to exploit the knowledge of the sociological orbits in users' movement, this work is the first that proposed an efficient method to determine the main mobility profiles of a user using a mixture of Bernoulli's as the clustering algorithm, and then make either unconditional or conditional predictions on which hubs a user might visit. More specifically, our results use only a short history (e.g., the past 7 days) of mobility trace data, but has been shown to predict around 10% to 30% more accurately than a general statistical approach that relies on daily collection and computation of the trace data. This illustrated the strength of our sociological orbit aware approach, and in particular, the usefulness of the mobility profiles of a user.

Note that although this work is based only on the mobility trace data from ETH Zurich, it is expected that the data analysis, mobility profiling and location prediction techniques we have developed, as well as the conclusions we have drawn in this work that validate the existence and usefulness of the sociological orbits are in general applicable to other university and corporate campuses, as well as other public/private environments (there certainly isn't a sufficient amount of mobility trace data available except from a couple of places). In addition, we expect that this work will inspire additional innovative work on social influence aware and user-centric designs and operations of not only wireless access networks, but also mobile ad hoc and peer-to-peer networks, as well as intermittently connected or, delay tolerant networks.

Chapter 6

SOLAR for Mobile Ad Hoc Networks (MANET)

In this chapter, we introduce a novel concept of "macro-mobility" information obtained from the sociological movement pattern of MANET users, and propose a routing protocol that can take advantage of the macro-mobility information. This macro-mobility information is extracted from our observation that the movement of a mobile user exhibits a partially repetitive "orbital" patten involving a set of "hubs". This partially deterministic movement pattern is not only practical, but also useful for locating nodes without the need for constant tracking and for routing packets to them without flooding.

More specifically, this chapter makes the following two contributions. First, it proposes an ORBIT mobility framework to achieve this macro-level abstraction of orbital movement. Second, to take advantage of this hub-based orbital pattern, it proposes a Sociological Orbit aware Location Approximation and Routing (SOLAR) protocol. Extensive performance analysis shows that SOLAR 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.

The rest of this chapter is outlined as follows. In Section 6.1 we describe our SOLAR protocol in detail and analyze the impact of different mobility models generated by our SOLAR framework on the protocol performance in Section 6.2. In Section 6.3, we compare the different forwarding schemes of SOLAR with ABSoLoM [46]. In Section 6.4, we present further results for the performance comparison of SOLAR with LAR and DSR and we conclude this chapter in Section 6.5.

6.1 Sociological Orbit aware Location Approximation and Routing (SOLAR)

In this section, we describe our Sociological Orbit aware Location Approximation and Routing (SOLAR) 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. In the following discussion, nodes in SOLAR are assumed to be equipped with either GPS, or other localization devices [1] to facilitate greedy geographic forwarding as to be explained in detail in the following sections.

6.1.1 Protocol Overview

Several motivations for *peer collaboration* (among acquaintances) were discussed by the authors in [10]. In one of our earlier work [46], we proposed an Acquaintance Based Soft Location Management (ABSoLoM) scheme, and also showed the advantages of using acquaintances to form a distributed location database. Although one of the basic concepts in SOLAR is also the use of acquaintances, it differs from ABSoLoM

[1] Note that this assumption is common to most location aided MANET routing protocols suggested in literature

in at least two 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 SOLAR 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 will 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).

Since the orbital mobility profile or, the hub list information stays valid for a much longer time when compared to the exact location coordinates, SOLAR can significantly reduce overhead in terms of *location updates* in the face of node mobility. More specifically, in SOLAR, 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 forming new acquaintances). Only the 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 Orbit Timeout). A more detailed description of SOLAR is as follows.

6.1.2 Profile Query and Response

In SOLAR, 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 is an acquaintance whose hub list is cached by the source, the *data* packet is sent towards the (center point of) hub(s) in that hub list (see Section 6.1.3 for more details).

This packet transmission is based on "greedy geographic forwarding" [89], where each intermediate node chooses its next hop from amongst its neighbors who is closest to the destination than itself (see Section 6.1.4 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 6.1.5. 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 consists 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 may forward 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 a *response* packet) with its own hub list, but also indicates its current hub. 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 Orbit Timeout.

6.1.3 Hub-level Transmissions

Once the source of any packet (*query, response*, or *data*, as well as *update* as to be described later) 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 the information is available, a single packet is geographically forwarded to (the center of) that current hub. However, if that information is unavailable, we suggest three different approaches:

Scheme 1: Sequential 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 than itself. When a local maxima occurs, the packet is redirected towards the next unvisited Hub in the destination's Hub list. 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.

Scheme 2: Simulcast The source geographically forwards a copy of the packet towards each Hub in the list. This way, although the overhead may be more, the latency is reduced. However, if a Hub is unreachable, then no further efforts are made to send a copy of the packet to it, unlike in the Sequential Scheme.

Scheme 3: Multicast This scheme is suggested to explore the middle ground of the above two schemes. The source forms a minimum spanning tree (MST) based on the Euclidean distances between the centers of the Hubs in the list, with its own Hub being the root. Then it multicasts copies of the packet to every branch from the root. The packet contains this MST in the header and following nodes keep multicasting down the tree.

For all the above schemes, location is updated only within nodes that have an active "data session" between themselves, as mentioned earlier. More specifically, 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 its 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 Orbit 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).

6.1.4 Packet Delivery via Geographic Forwarding

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 there is no neighbor closer to the Hub than the node itself(also called a local maxima or "geographic hole"), 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 [75].

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.

6.1.5 Optimizing Profile Queries

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

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

This is a minimum Set Cover problem, which is known to be NP-Complete [43]. To find an heuristic solution, we have adopted the Quine-McCluskey technique [97, 108] 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., $\nexists A_j, s.t. H_j \supseteqq H_i$). Formally, A_i (with H_i) is a *Prime* acquaintance *iff*:

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

Let us consider an example (6.1) where, $H_1 = \{1,2\}$, $H_2 = \{2,3,4\}$, $H_3 = \{1,4\}$, 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 $H_2 \supseteqq H_4$).

Essential Prime Acquaintance: This is a *Prime* acquaintance that covers at least one hub that is not covered by any other *Prime* acquaintance. Let $P = \{H_{p_1}, H_{p_2}, \ldots\}$

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*:

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

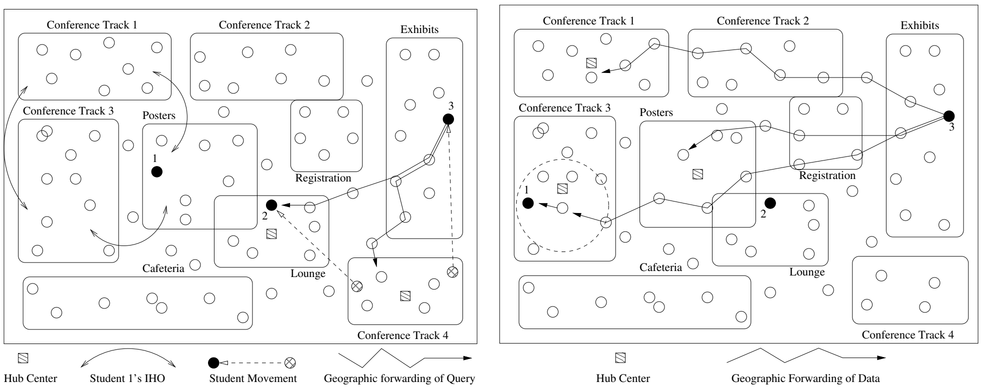
Continuing with the previous example 6.1, even though A_1 is a *Prime* acquaintance, it does not cover any hub that is not already covered by either A_2 or A_3. So A_1 is not an *Essential Prime* acquaintance. Following the same principle, A_3 is not an *Essential Prime* acquaintance either. However, 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.

Once the optimal subset of acquaintances is determined, each acquaintance in that subset is queried as explained in detail in Section 6.1.2. Referring to the previous example 6.1, first A_2 (being an *Essential Prime* acquaintance) will be chosen , following which any one of the other *Prime* acquaintances (A_1 or A_3) will be chosen to cover hub 1 (which 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 be sent to hub 1 for A_1 and to hubs 3 and 4 for A_2, but a single anycast packet destined for any of A_1 or A_2 will be sent to their common hub 2. In addition to reducing responses, this will also minimize the number of query packets generated, leading to a lower control overhead.

6.1.6 An Example Scenario

Continuing our previous example involving mobile users attending a large technical symposium, let us assume that three 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 6.1, where the Random Orbit model may be assumed for the sake of our discussion here. 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 constitute Student 1's Orbit), while Student 2 frequents the *Lounge* and the room for *Conference Track 4* (which constitute Student 2's Orbit), and Student 3 frequents the rooms for *Exhibits* and *Conference Track 4* (which constitutes Student 3's Orbit). 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 shared 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 6.1(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 6.1(b)).



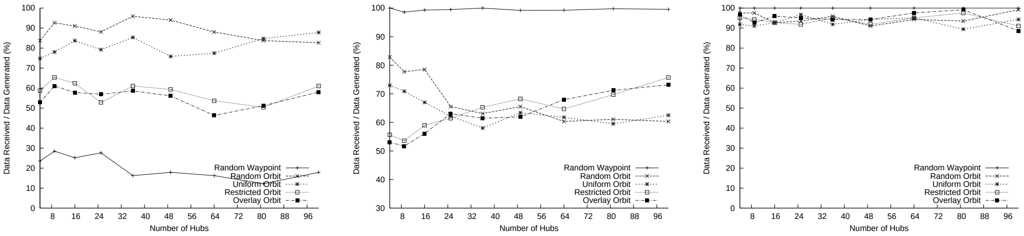
(a) Student 3 queries acquaintance (Student 2) for Student 1 (b) Student 3 sends data to Student 1's hub list

Figure 6.1: Use of Sociological Mobility Profiles of MANET Users for Location Approximation and Routing

6.2 Mobility Impact on Protocol Performance

To study the effect of mobility on the performance of SOLAR, DSR and LAR1 (LAR in Scheme 1) we simulated each of the protocols with respect to all the mobility models suggested as examples in Chapter 4, Section 4.3, and compared the *data throughput*. For simplicity, we just simulated the SOLAR Scheme 1 (Sequential) for comparing it with LAR1 and DSR. For both DSR and LAR1, we borrowed the implementation available in GloMoSim [137]. In our simulation of SOLAR, we fix the threshold for *logical hops* of *query* packets to 1. In this way, if the acquaintance of a source fails to provide the required information, it does not forward the *query* to its own acquaintances.

As seen in Figure 6.2(a), SOLAR performs best for Random Orbit and Uniform Orbit models as expected since the protocol was formed keeping the most general model in mind. The results for Restricted Orbit and Overlay Orbit are lower due to the restriction in movement imposed upon the mobile nodes that does not favor our acquaintance formation. In our modified Random Waypoint, all nodes have a single Hub as part of



(a) SOLAR Throughput (%) (b) DSR Throughput (%) (c) LAR Throughput (%)

Figure 6.2: Impact of mobility on protocol performance

their LAO. So all the packets make their way to the center of the terrain irrespective of the destinations location, resulting in the worst performance.

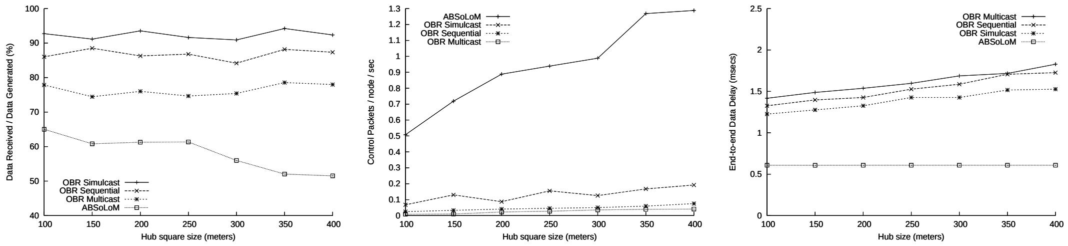
DSR on the other hand is seen in Figure 6.2(b) to perform the best with respect to the modified Random Waypoint where its flooding nature complements the fact that the nodes are evenly spread out all over the terrain. With respect to the other models it performs well with respect to the Random Orbit and Uniform Orbit for small number of Hubs when each Hub contains a fair share of the nodes, and does better for Restricted Orbit and Overlay Orbit when the number of Hubs increases, creating small Hubs with few nodes, thereby ensuring even node distribution. LAR1 (LAR in Scheme 1) as seen in Figure 6.2(c) is not much affected by the difference in mobility and performs consistently across all our example models.

6.3 Performance of SOLAR Forwarding Schemes and ABSoLoM

In this section, we proceeded to implement the other two forwarding schemes of SOLAR and compare them against each other and ABSoLom. In this section, we refer to the different SOLAR schemes simply as Orbit Based Routing (OBR) schemes. Following are the results observed.

6.3.1 Variation in Hub Size

The importance of this metric has already been mentioned before. 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 10s and 25s.



(a) Data Throughput (%) (b) Control Overhead (c) End-to-End Delay

Figure 6.3: Protocol Performance vs. Hub Size

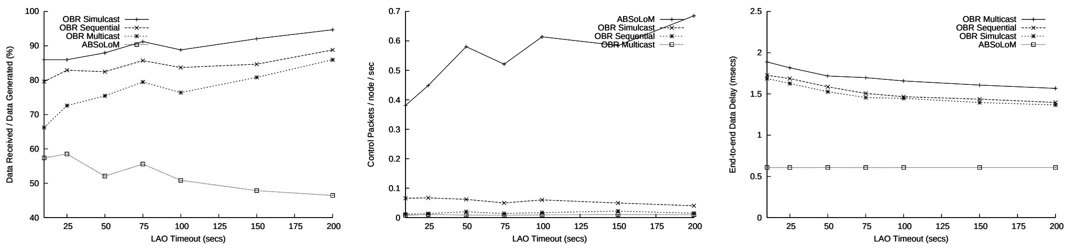
Data Throughput The data throughput is measured in terms of the fraction of the total number of data packets generated that were received successfully. In OBR, a source learns of a destination by first making acquaintances with nodes that are within its radio range, and then using the distributed location 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 acquaintanceship 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 6.3(a).

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 *acquaintanceship request* and *acquaintanceship 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 acquaintanceship breakage, leading to increased rate of acquaintanceship formation. Thus, the control overhead is seen to increase with increasing Hub Size in Figure 6.3(b). 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 uni-casting 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.

End-to-end Data 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 6.3(c), OBR shows marginally longer 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 uni-casting of multiple copies of the packet to all the Hubs.

6.3.2 Variation in Hub Stay

The importance of this metric too has been discussed before. The number of Hubs are fixed to 10 and Hub Size (min, max) is set to 200m and 300m.



(a) Data Throughput (%) (b) Control Overhead (c) End-to-End Delay

Figure 6.4: Protocol Performance vs. Hub Stay

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 6.4(a). Moreover, due to the connection maintenance phase, data packets are directed first to the last known Hub of the destination. Thus increased Hub Stay Time 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 Hub Stay Time, 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 Hub Stay Time.

Control Overhead The relative difference in control overhead shown in Figure 6.4(b) is similar to that seen in Figure 6.3(b) where OBR performs much better in general than

ABSoLoM. In ABSoLoM, the higher control overhead is primarily due to the increased number of acquaintanceship 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 Hub Stay Time. Among the OBR variations, Simulcast has the highest and Multicast has the lowest overhead for same reasons as described before.

End-to-end Data Delay As seen before in Figure 6.3(c), the delay in OBR is marginally higher overall, as shown in Figure 6.4(c). More specifically, an increase in the Hub Stay Time increases the probability of finding a destination in its last known current Hub, where a data packet is sent first, causing the delay to decrease gradually with an increase in the Hub Stay Time. Delay in ABSoLoM is not much affected by the change in Hub Stay Time 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.

6.4 Performance Analysis

In this section, to show that existing routing protocols under comparable assumptions (such as DSR and LAR) cannot be effectively used to deal with such practical orbital mobility pattern, we run extensive simulations to compare the performance of the SOLAR protocol with that of DSR and LAR scheme 1 (LAR1) using GloMoSim [137]. ***Note that it is neither possible, nor our intention to compare our protocol with every other existing MANET routing protocols. Rather, the purpose of this exercise is to show that new routing protocols such as SOLAR is needed to take advantage of the orbital mobility pattern for efficient routing within MANET.*** Accordingly, other routing protocols such as [75, 90] that require the use of a separate location service above the routing layer are not considered here.

We implement two versions of the SOLAR simulcast protocol, SOLAR-1 and SOLAR-

2. In SOLAR-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 SOLAR-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 one or two radio hops away. In both versions of SOLAR, we use a threshold value of two 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 [78, 107, 134]. We summarize the major simulation parameters in Table 7.1. We chose three metrics to evaluate the performance of each protocol as described below:

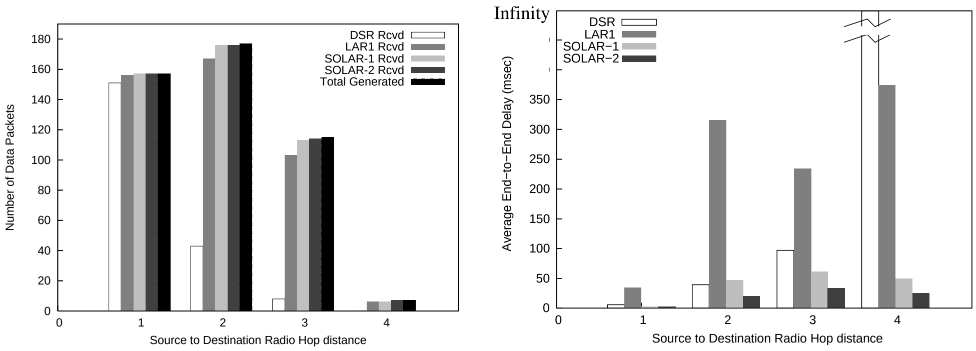
Data Throughput: This metric is defined as the ratio of the total number of data packets received by all intended destinations, to the total number of data packets generated by all sources.

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 SOLAR, the control packets are *Hello, Hub List Query, Hub List Response,* and *Location Update* packets. Although, in SOLAR, 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 6.1: Simulation Parameters for MANET

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 (RWP + P2P)
ORBIT PARAMETERS	
Total Hubs (*Rooms*)	*Vary*, (Default= 15)
Hub Size	*Vary*, (Default= 200m-300m)
Hub Stay Time	*Vary*, (Default= 50s-100s)
Orbit 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

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 includes the discovery delay, which is 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 SOLAR). 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 compare different protocols as fairly as possible, 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 minimum possible delay 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. We use the same minimum possible delay while calculating the approximation factor for each of the three protocols.



(a) Data Delivered (b) End-to-End Delay (msec)

Figure 6.5: Protocol Performance at various Source-Destination Radio Hop Distances

The main reason for dividing the observed delay of each packet by the minimum possible 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. To further illustrate this point, we have run a few simulations with the default parameters mentioned in Table 7.1 and classified the data packets delivered successfully, in terms of the radio hop distances between their respective sources and the destinations. Figure 6.5(a) shows the results of our simulation for a given set of packets generated with different source-destination radio hop distances (note that the number of packets generated with a distance of 4 hops is very small in this case). The number of data packets delivered by DSR drop significantly once the destination is more than 2 radio hops away from the source (whereas LAR1, SOLAR-1 and SOLAR-2 deliver a high percentage of packets consistently across all hops). On the other hand, the end-to-end delay in DSR increases with the hop distance (and may be considered to reach ∞ when the hop distance is 4 as zero packets are delivered in this case) as shown in Figure 6.5(b) [2]. Thus, a simple comparison of the average end-to-end delay, excluding these undeliv-

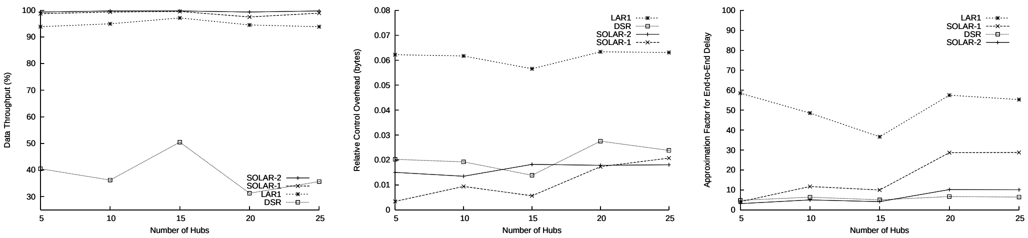
[2] whereas in LAR1 and SOLAR, this may not be the case due to the fact that the discovery delay (which is a part of the end-to-end delay) does not necessarily increase with the hop distance

ered packets in particular protocols, would not be fair. More specifically, DSR would enjoy a lower average delay when compared to LAR1 just because DSR is unable to send the 4-hop packets which would otherwise incur a higher delay over a longer path. To introduce a sense of fairness in the delay performance comparisons, we choose to compare the *approximation factor*, which measures the end-to-end delay relative to the "optimum" delay (after the notion of "approximation factor" in algorithms that measure the closeness of an heuristic solution to the optimum solution).

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.

6.4.1 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.



(a) Data Throughput (%) (b) Relative Control Overhead (c) Approximation Factor for E2E Delay

Figure 6.6: Protocol Performance vs. Total number of Hubs

Data Throughput: Figure 7.5(a) shows the data throughput of all the protocols with varying number of hubs. SOLAR-2 and SOLAR-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 SOLAR-2, SOLAR-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* [102] (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 SOLAR 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 6.6(b), we note that LAR1 has the highest overhead, followed by SOLAR-1, DSR and SOLAR-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 Orbit. 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.

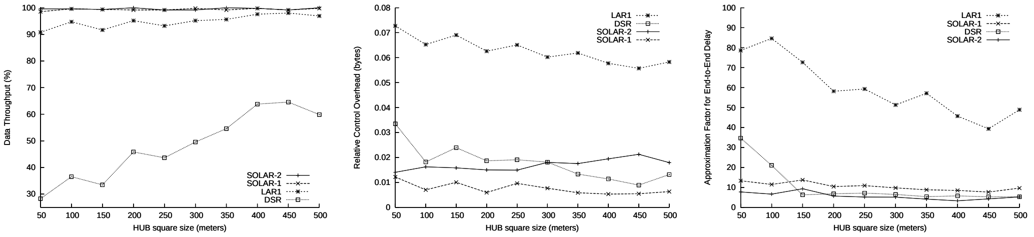
DSR adopts a less aggressive flooding scheme and is shown to have a lower overhead than LAR1. In the case of SOLAR protocols, the periodic HELLO beacon in SOLAR-2 contains more information than that of SOLAR-1. Thus, the overhead in SOLAR-2 is more than that of SOLAR-1. More specifically, in SOLAR, hub 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, SOLAR 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 6.6(c). LAR1 has the highest delay due to its iterative estimation of node location, and increased control overhead. In SOLAR-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 SOLAR-1 increases marginally with increasing number of hubs. SOLAR-2, with more information, does considerably better than SOLAR-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 7.1).

Note that the results on the relative performances of the three protocols shown in Figure 7.5 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.

6.4.2 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.



(a) Data Throughput (%) (b) Relative Control Overhead (c) Approximation Factor for E2E Delay

Figure 6.7: Protocol Performance vs. Hub Size

Data Throughput: Figure 7.7(a) shows that SOLAR-2, SOLAR-1, and LAR1 perform consistently well across all hub sizes, with the SOLAR 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 SOLAR and LAR.

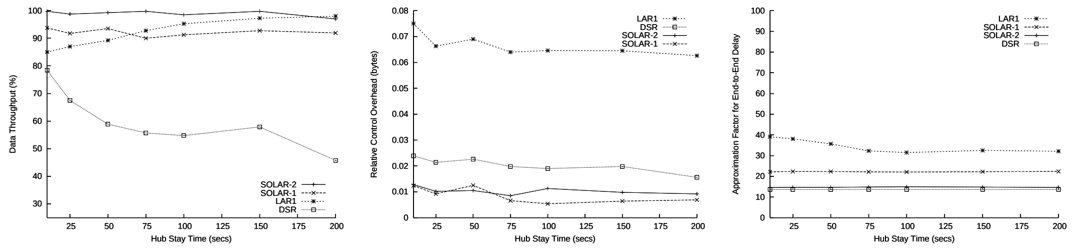
Relative Control Overhead: Once again, LAR has the highest control overhead, followed by DSR, SOLAR-2 and SOLAR-1, as seen in Figure 6.7(b). The reasons are similar to those given in Section 6.4.1 for Figure 6.6(b).

Approximation Factor for End-to-End Delay: Figure 6.7(c) shows LAR1 to have the highest approximation factor for the end-to-end delay, with DSR and SOLAR-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 SOLAR protocols enjoy a low hub list discovery latency as before, due to the use of the distributed location database within a network of acquaintances.

6.4.3 Variation in Hub Stay

This parameter has a direct impact on the average node velocity. Lower Hub Stay Time means shorter time spent by a node in a Hub, increasing the overall time spent in motion at higher speeds. On the other hand, higher Hub Stay Time signifies lesser nodes in transition between Hubs, thereby increasing the average node population in Hubs.



(a) Data Throughput (%) (b) Relative Control Overhead (c) Approximation Factor for E2E Delay

Figure 6.8: Protocol Performance vs. Hub Stay Time

Data Throughput: Since nodes in SOLAR 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, both SOLAR versions perform consistently well in terms of data throughput as seen in Figure 6.8(a). An increase in the

Hub Stay Time favors LAR1 by increasing its location estimation accuracy along with its data throughput. DSR suffers from congestion at the MAC layer as it tries to flood in a Hub with high node population. Thus, its data throughput steadily decreases with an increase in the Hub Stay Time. Due to the random CBR traffic used by our simulations, DSR cannot effectively use cached routes to overcome this problem.

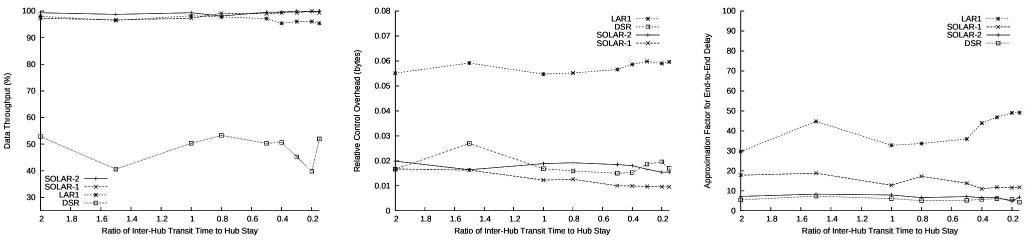
Relative Control Overhead: The relative difference in control overhead shown in Figure 6.8(b) is similar to that seen in Figure 6.7(b) where LAR has the highest control overhead, followed by DSR, SOLAR-2 and SOLAR-1. However, in the face of decreasing hub-level mobility (increasing Hub Stay Time), LAR1 is able to make better location estimates resulting in decreasing overhead. Similarly in SOLAR, lower mobility reduces the number of new neighbors a node interacts with leading to marginally lower control overhead.

Approximation Factor for End-to-End Delay: As seen before in Figure 6.7(c), the relative performances of the protocols for delay is seen to be the same in Figure 6.8(c) for similar reasons.

6.4.4 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 7.1, 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 6.9(a), shows that SOLAR-2, SOLAR-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



(a) Data Throughput (%) (b) Relative Control Overhead (c) Approximation Factor for E2E Delay

Figure 6.9: Protocol Performance vs. Ratio of Inter-Hub Transit Time to Hub Stay Time

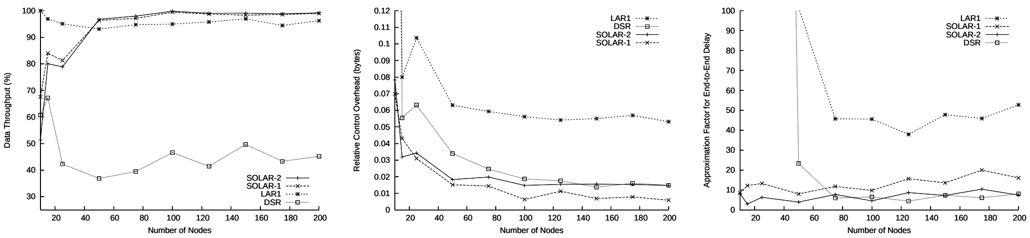
and higher delay as confirmed by our earlier observations. In the SOLAR protocols, high values of the ratio (i.e. nodes spending a large amount of time traveling in between hubs) does not have a significant impact on the throughput performance. This is because in SOLAR, intermediate nodes also respond 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 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, SOLAR-2, and SOLAR-1 in Figure 6.9(b) are similar to that observed in Figure 6.7(b), and for similar reasons.

Approximation Factor for End-to-End Delay: In terms of this metric, LAR1 has the highest values while both SOLAR-2 and DSR show comparable minimum results in Figure 6.9(c)(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. SOLAR-2 does better than SOLAR-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.

6.4.5 Variation in Number of Nodes (and Data Connections)

In this section, 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.



(a) Data Throughput (%) (b) Relative Control Overhead (c) Approximation Factor for E2E Delay

Figure 6.10: Protocol Performance vs. Number of Nodes (Data Connections = 2 x Number of Nodes)

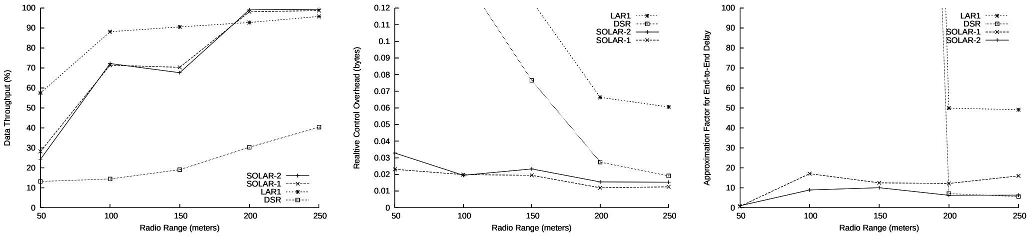
Data Throughput: With a small number of nodes (and connections), LAR1 performs the best as shown in Figure 7.6(a). In this case, DSR also benefits considerably and in fact, performs as well as SOLAR protocols. As for the SOLAR protocols, a very small number of nodes increases the chances of encountering a local maxima (or routing hole) while performing geographic forwarding. Additionally, with a fewer connections, SOLAR 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, SOLAR 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 6.10(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 6.10(c) shows that both LAR1 and DSR have a significantly higher delays with a small number of nodes. 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 SOLAR protocols, the orbital mobility information of the nodes is still effective enough to keep the node discovery latency to a consistently low value.

6.4.6 Variation in Radio Range (and Hub size)

The effect of a fixed radio range on varying hub sizes has already been discussed in Section 6.4.2. As a final study, 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.



(a) Data Throughput (%) (b) Relative Control Overhead (c) Approximation Factor for E2E Delay

Figure 6.11: Protocol Performance vs. Radio Range (Hub Size = Radio Range)

Data Throughput: As seen in Figure 6.11(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 SOLAR 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 and the relative performances of these three protocols are consistent with the findings so far.

Relative Control Overhead: Due to an increased average path length caused by a shorter radio range, flooding based protocols will incur higher overhead and delays. This is confirmed in Figure 6.11(b), which shows that both LAR1 and DSR have significant amount of control overhead for smaller radio range values. However, in the SOLAR protocols this effect is not as significant as in DSR or LAR1. More specifically, with shorter radio ranges, nodes in SOLAR have fewer neighbors that implies fewer acquaintances and a lower protocol maintenance overhead on average. On the other hand, for longer radio ranges, nodes get to know of many other nodes' hub lists, leading to reduced overhead in terms of request/response packets. Thus, the overhead in SOLAR is seen to be consistent across varying radio ranges.

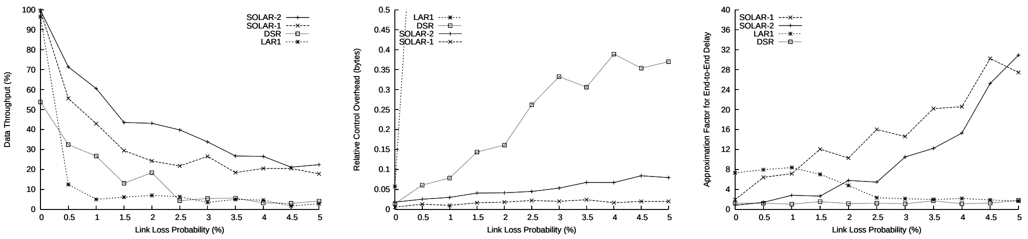
Approximation Factor for End-to-End Delay: Figure 6.11(c) shows that the delays for both LAR1 and DSR are significantly higher for small radio range values. This is because longer paths in source routing schemes tend to break more often, causing retransmissions, and in turn higher delays. In the SOLAR protocols, since nodes move along an orbit, they are able to continue to share hub lists with other nodes, and thus the delay remains consistently lower than that in DSR and LAR1.

An interesting point worth mentioning here is the relationship between the route/hub discovery latency in each protocol with the distance (in radio hops) from the source to the destination. In both LAR1 and DSR, the discovery latency for a previously unknown node is directly proportional to the distance. This is because, even with the caches at intermediate nodes, longer routes have a higher probability of link breaks, resulting in route errors and leading to higher discovery delay. However, in SOLAR protocols, this relation is not that intuitive. For example, in SOLAR-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 SOLAR 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.

6.4.7 Effect of Lossy Links and Gray Zones

Wireless networks are seldom attributed with 100% link reliability due to varying channel conditions. As such, it is common to find asymmetric links and “gray zones” [95] within the network. Unlike a geographic hole, a gray zone may occur due to the presence of a radio neighbor during neighbor discovery, but failure during the actual transmission of “unicast” data to that neighbor.

Although the presence of specific gray zones affect only a fraction of links in the network, we decided to study the effect of uniform link loss probability across the network. Thus, the effect of link loss is quite significant even at very low loss probabilities as any link is subject to the probabilistic failure at any time.



(a) Data Throughput (%) (b) Relative Control Overhead (c) Approximation Factor for E2E Delay

Figure 6.12: Protocol Performance vs. Link Loss Probability

Figure 6.12(a) shows a steady decline in data throughput of all the protocols, with SOLAR protocols doing much better than DSR and LAR1. LAR1 seems to be affected the most. It is possible that due to the orbital movement of nodes, the cached information of LAR1 expires quickly, and causes nodes to restart the query process. As all links have a probability of failure, queries may be lost in gray zones more often leading to reduced throughput. On the other hand, SOLAR enjoys longer lifetime of cached hub list information and require less amount of query packets to be forwarded. This is also reflected in Figure 6.12(b), where LAR1 is seen to have the highest relative overhead, and SOLAR the least.

The approximation factor for end-to-end delay in SOLAR is seen to increase in Figure 6.12(c), while that in DSR and LAR1 is seen to decrease. This may be explained by the fact that nodes in SOLAR are able to deliver more packets than any of the other protocols, at the cost of additional delay incurred when data is cached at different nodes.

Thus, even though SOLAR is affected by increasing link loss probabilities, it still outperforms DSR and LAR1 in terms of data throughput and relative control overhead, while the increase in the approximation factor for delay is justifiable. Note that the techniques suggested in [95] may be adopted within SOLAR to reduce the occurrence of gray zones.

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

6.5 Summary

In this chapter, we have exploited a partially deterministic and hub-based orbital mobility information by observing the social influence on the macro-mobility pattern of each MANET user. Specifically, we have proposed an ORBIT framework to capture the orbital movement pattern for MANET users based on a list of places or hubs that they frequently visit. ORBIT is useful in capturing realistic mobility by integrating micro-level mobility models with a macro-level inter-hub orbit. We have also used this simple yet practical mobility information to perform intelligent routing. In particular, we have proposed a Sociological Orbit aware Location Approximation and Routing

(SOLAR) protocol for MANET and established the advantages of SOLAR over conventional MANET routing protocols such as LAR and DSR in terms of higher data throughput, lower control overhead, and lower end-to-end delay.

Chapter 7

SOLAR for Intermittently Connected Mobile Ad Hoc Networks (ICMAN)

In this chapter, we present our novel routing solutions to the routing problems within a special class of Intermittently Connected Networks (ICN) called the ICMAN that is not only challenged by the uncertainty and time varying nature of network connectivity, but also shares other constraints of a MANET. First, we consider a practical mobility framework based on our findings in [44], where an analytical study of wireless users' mobility traces revealed that users usually move between a small set of socially significant places (called "hubs"), to form the so-called "sociological orbits" [51]. Second, to leverage on the sociological orbit based mobility profiling techniques [44] in routing within ICMAN, we propose one multi-path hub-level routing method, and two variations of user-level routing techniques based on "contact probabilities". We then compare these approaches with that of Epidemic Routing [128] to highlight the advantages of using our concept of mobility profiles based on the sociological orbital framework in performing efficient routing within ICMAN. But first, we describe the sociological mobility framework suitable for ICMAN.

Sociological Movement Pattern for ICMAN: In the real world, users routinely spend a considerable amount of time at a few specific place(s) that we refer to as hub(s). For example, a graduate student in school may visit and spend some significant amount of time in his/her laboratory, a seminar room, or the cafeteria. Although it is hard (or may be even against privacy policies) to keep track of an individual at all times, one can still take advantage of the fact that most users' movements are within and in between a list of hubs. In these situations, it is often possible to estimate/measure hub-visit probabilities and inter-hub movement patterns of an individual. This information then constitutes a part of the users' *mobility profiles*. For example, even if we do not know the exact location of the graduate student at any given time, given his/her mobility profile we can most probably find him/her in either the laboratory, or the seminar room, or the cafeteria, without having to look all over the building/campus. The more "periodic" the movement pattern is, the more we can take advantage of the mobility profile. This orbital concept is illustrated in Figure 7.1.

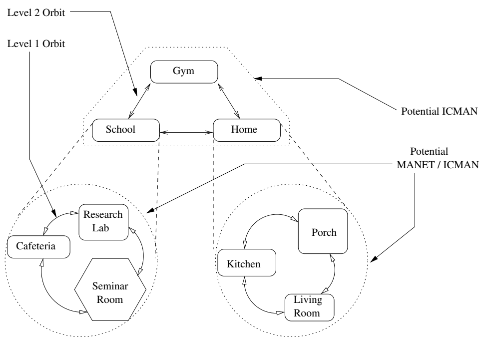


Figure 7.1: A hierarchical view of sociological orbits

In practice, hubs can be identified in a variety of ways. The use of GPS service is the obvious first choice. Alternatively, signal strengths of wireless Ethernet packets can be used for location sensing and real-time tracking [81]. In the broader contexts of *pervasive/ubiquitous computing* [2], and *Ambient Intelligence* (AmI) [112], localization in a cosmopolitan area will be even more readily available. As part of our ongoing work we are in the process of analyzing real time experimental data on user mobility and network usage patterns to validate the existence of hubs and mobility profiles.

An Example Probabilistic Orbit Model: To illustrate the concept of the sociological orbital movement, we first construct a simple yet practical orbital model called the Probabilistic Orbit, which is different from our earlier proposed Random Orbit model [51] in ways that are explained in the following description. The Probabilistic Orbit model allows for the creation of a certain number of hubs within the simulation terrain for all the nodes, as specified by the parameter *Number of Hubs*. These hubs are located at random places within the terrain. However, unlike in Random Orbit, in Probabilistic Orbit hubs are not allowed to overlap with each other (in accordance with the disconnected nature of an ICMAN). Each node can visit a subset of randomly chosen hubs, forming its hub list, following a specified transition probability matrix creating a *Probabilistic Orbit*. The list of hubs a node visits is bounded by *Hub List Size*, and the time it spends in each hub is an exponential random variable with mean specified by *Hub Stay Time* (unlike the uniformly distributed hub stay time in Random Orbit). Together, these two parameters define an Inter-Hub Orbit (IHO). Probabilistic Orbit further differs from Random Orbit in that Probabilistic Orbit assumes the hub list assigned to each node to stay constant for the entire duration of the simulation.

The mobility pattern of individual nodes shall comprise of two parts: movement inside a hub, and movement in between hubs. For convenience, the movement inside each hub, which shall also be referred to as the Intra-Hub Movement (IHM), was chosen to follow a modified Random Waypoint mobility model, whose speed range is denoted by *Intra-Hub Speed* (with a non-zero minimum as suggested by [136]), and whose pause time is denoted by *Intra-Hub Pause*. For movement in between hubs, we define a Point-to-Point Linear (P2P Linear) model. In this model, when a node wants to leave one hub for another, it first randomly selects a point within the destination hub. Second, it chooses a transition time that is exponentially distributed with mean *Inter-Hub Transi-*

tion Time. Third, it moves towards the destination point linearly from its current position with the velocity obtained by dividing the total travel distance by the total travel time. However, unlike in Random Orbit, nodes in Probabilistic Orbit are assumed *in most cases* to not communicate with any other node while it is traveling from one hub to another. ***Note that for each of the two parts, any known practical mobility models satisfying similar properties as above, may be chosen.***

Figure 7.2 illustrates the Probabilistic Orbit model. ***Note that, this example Probabilistic Orbit model does not simply integrate two common mobility models (Random Waypoint, and P2P Linear), but most importantly also introduces the practical orbital movement amongst hubs.*** Such a model is suitable for modeling wireless devices carried by users working in an office building, attending a convention, or around a campus, which may constitute the ICMAN. As users move around, devices either automatically, or with the user's permission/assistance may record the hubs visited, along with the frequency of visits to each of those hubs, and share the hub-based orbital mobility profile with trusted "acquaintances". Such mobility profile can then help improve routing as described next.

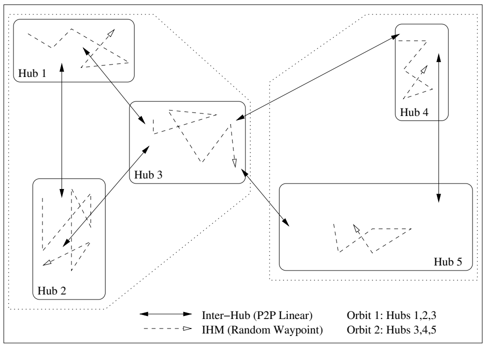


Figure 7.2: The Probabilistic Orbit Model

The rest of this chapter is outlined as follows. In Section 7.1, we present a detailed description of the suite of SOLAR protocols suitable for ICMAN scenario and analytically study a few of them in Section 7.2. We then compare the performance of our protocols via rigorous simulation study in Section 7.3, and finally conclude this chapter in Section 7.4.

7.1 The SOLAR Protocols

In a conventional MANET (as opposed to intermittently connected networks), Sociological Orbit aware Location Approximation and Routing (SOLAR) framework uses a concept of "acquaintance" similar to that in our work in Acquaintance Based Soft Location Management (ABSoLoM) protocol [46], as well as to some degree the concept of *peer collaboration* (among "acquaintances") in [10]. For a detailed description of SOLAR and its application towards MANET the readers are referred to our prior work in [50, 51]. In an ICMAN however, we apply the concepts of "contact probability" and "delivery probability" in proposing a series of multi-path protocols to address the inherent network differences between a MANET and an ICMAN.

7.1.1 Static SOLAR-KSP Algorithm

In this version of SOLAR, we assume that each node knows of every other node's hub list and its associated transition probability matrix. Also, as the simulation proceeds we assume nodes to be able to communicate with other nodes within radio range even while traveling from one hub to another. Under this assumption, each node distributively does the following: First, every node computes the *contact probability* with every other node, taking into consideration the communication of nodes within hubs only. Second, every node computes the *delivery probability* to all other nodes. As to be discussed in Chapter 8, Section 8.1, the method for obtaining the contact probability of two nodes within a specified time interval T is computationally expensive. Thus, for the sake of simplicity we estimated the contact probability in our simulation from the actual observed mobility patterns of every pairs of nodes for the duration of the entire simulation. However, it should be noted that in the real world the computation of this contact probability as suggested by our analytical model in Chapter 8, Section 8.1 would need to be done only once at start (and may change only rarely), and hence is practical. For computing the *delivery subgraph* associated with this SOLAR protocol we use the Dijkstra's Shortest Path algorithm [41], and then compute the delivery probability as described next.

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

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

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

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

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

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

1. *send-to-all*: send copies of the packet to all the next hops on the k-best paths that come within radio range

2. *send-to-best*: send a copy of the packet to the next hop node on the best path only, if it comes within radio range

3. *send-to-any*: send a copy of the packet to any of the next hop nodes on the k-best paths that come within radio range

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

7.1.2 Dynamic SOLAR-KSP Algorithm

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

7.1.3 SOLAR-HUB Algorithm

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

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

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

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

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

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

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

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

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

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

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

7.2 Brief Analytical Study of S-SOLAR-KSP

In this section, we shall briefly analyze the different schemes of S-SOLAR-KSP to gain a better understanding of their effects on the protocol performance.

7.2.1 Analysis of Forwarding Schemes

In the interest of analyzing the relative performance of the *send-to-best* forwarding scheme to that of the *send-to-all* forwarding scheme, we consider a simple example network (Figure 7.3(a)) where the height (h) of the shortest path between the source (s) and the destination (d) is 2. Let us assume that all nodes (except d) have an incoming degree of 1 and an outgoing degree (k) of 2. Let P be the *contact probability* for all pairs of nodes. Let us consider all paths from s to d that have a maximum length of $h+1$ (since paths longer than that have negligible delivery probabilities in our simulation environment). The ratio (R) of the delivery probabilities (P^d) of the two schemes can then be written as

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

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

To get a lower bound, we always ignore all higher degree terms in the denominator in each calculation that are less than 0. By doing this we have

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

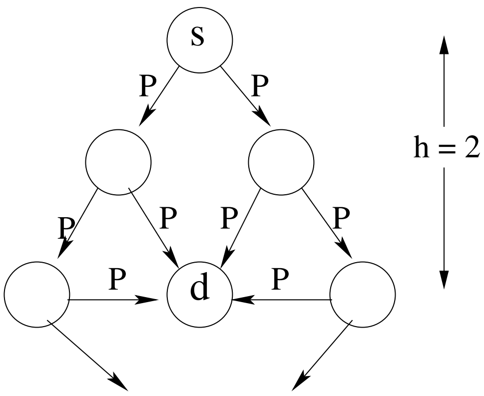
The denominator will be maximum for $P = 0.5$, giving the worst case ratio of P^d_{best} to P^d_{all} as 0.4. If we extend the graph to have $k = 2, h = 3$ (Figure 7.3(b)) and still consider all paths with maximum lengths of $h+1$, we can get the simplified ratio

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

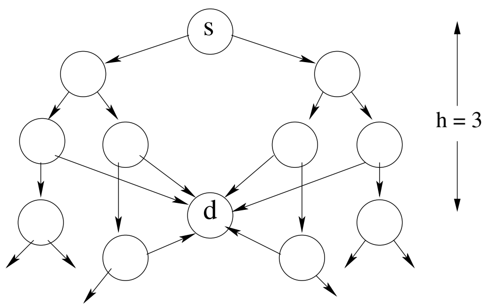
By neglecting the higher order terms in the denominator once again, we can get the general form of this ratio as

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

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



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



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

Figure 7.3: Examples for analysis of forwarding schemes in Static SOLAR-KSP

7.2.2 Performance Study

To evaluate the effect of different k values on SOLAR-KSP under each of the 3 schemes: *send-to-all, send-to-best*, and *send-to-any*, we simulated our protocol in the GloMoSim

[137] simulator using the default parameter values as shown in Table 7.1. We compare the metrics *data throughput, network byte overhead,* and *average end-to-end data delay*, all of which are defined in more detail in Section 7.3. For the sake of our following discussion, let us define N_{next}^{ksp} for each node i to be the set of nodes (maximum of k) that form the next hops on each of the KSPs emanating from node i to a given destination node.

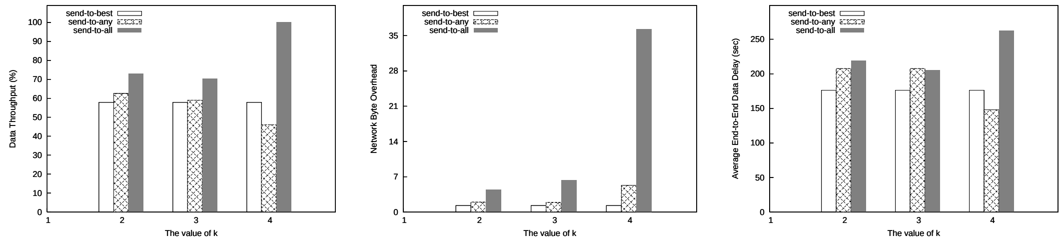
In the results shown in Figure 7.4 we see that *send-to-best* is independent of the values of k since it only sends data along the best path. Whereas, for the other two schemes: *send-to-any* and *send-to-all*, a value of $k = 1$ makes them similar to *send-to-best*. Hence, we only show protocol results for k from 2 to 4 and results for *send-to-best* represent the performance of all 3 schemes for $k = 1$.

In Figure 7.4(a) we see that *send-to-all* has the highest throughput, but it also has the highest overhead as seen in Figure 7.4(b). This is understandable as *send-to-all* uses all of the KSPs computed at start of simulation. More specifically, for larger values of k the overhead of *send-to-all* is seen to grow significantly due to the exponential increase in the number of packets forwarded at each intermediate node. Overall, we get the optimum results (high throughput, but low overhead) for *send-to-all* with $k = 2$. On the other hand, the throughput for *send-to-any* is seen to decrease with increasing values of k. This is because as k increases, the cardinality of N_{next}^{ksp} increases, and as *send-to-any* chooses the first available node in N_{next}^{ksp}, the difference in the delivery probability of that chosen node to that of the best choice within N_{next}^{ksp} may increase. Scheme *send-to-best* has the lowest overhead as it waits for the best choice in N_{next}^{ksp}, and consequently has the highest dropping rate (i.e., lowest throughput) when that best choice is not available within a specified time interval T.

In Figure 7.4(c), we see that for each of the 3 schemes, the end-to-end data delay is proportional to the data throughput as seen in Figure 7.4(a), with varying values of

k. This is because, data packets are sent from the source uniformly to nodes that are both near and far, with destination nodes that are near (shorter paths) having a higher probability of receiving a packet correctly. Hence, larger number of packets delivered also implies larger number of packets reaching destination nodes further apart (over longer paths), involving higher values of delay, which consequently increase the average end-to-end delay.

From the results above, it is evident that in our given scenario Static SOLAR-KSP performs the best with the *send-to-all* scheme with $k = 2$. As such, we shall use this variation of Static SOLAR-KSP, also referred to as ***S-SOLAR-KSP***, for our performance comparison in Section 7.3 with the other SOLAR protocols, to be described next.



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

Figure 7.4: Performance of Static SOLAR-KSP with varying k

7.3 Performance Analysis of SOLAR Protocols

In this section, we describe the extensive simulation study we carried out to compare the performance of the SOLAR protocols: ***S-SOLAR-KSP, D-SOLAR-KSP***, and ***SOLAR-HUB*** using the GloMoSim [137] simulator. We included the Epidemic Routing protocol [128] (referred to as ***EPIDEMIC*** in this work) in our comparisons because of its simple yet efficient performance in face of general intermittently connected networks. For the simulation scenario, we considered an ICMAN built within a corporate campus consisting of several buildings (hubs). Corporate employees spend most of their time within the hubs and intermittently move in between hubs. To model realistic speeds of mobile users within such a network, we considered the work in [78, 134] and fixed the ORBIT Inter-Hub and Intra-Hub time/speed parameters, along with the other simulation parameters as shown in Table 7.1. We chose three metrics to evaluate the performance of each protocol as described below:

Table 7.1: Simulation Parameters for ICMAN

GENERAL PARAMETERS	
Simulation Duration (each run)	3000s
Terrain Size	1000m x 1000m
Number of Nodes (*Users*)	*Vary*, (Default= 100)
Radio Range	125m
Cache Size	*Vary*, (Default= 200 Packets)
Cache Timeout	*Vary*, (Default= 400s)
MAC Protocol	IEEE 802.11
Mobility Model	Random Orbit (RW + P2P)
ORBIT PARAMETERS	
Total Hubs	*Vary*, (Default= 15)
Hub Size	50m x 50m
Hub Stay Time	Power Law (Mean= $10^6/x^3$)
Hub List Timeout	None
Hub List Size	Power Law (Mean= $0.7/x^2$, 1 to Total Hubs)
Inter-Hub Transition Time	Exponential (Mean= 40s)
Intra-Hub Pause	1s
Intra-Hub Speed	1m/s-10m/s
TRAFFIC PARAMETERS	
CBR connections	30 (120 packets each) Random
Data Payload	1460 bytes per packet

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

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

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

In what follows, we will examine how the total number of hubs, and the total number of nodes affect the protocol performance. To this end, we vary one of these two factors while fixing all other parameters to their default values. Each plot point in the results is averaged over 6 different simulation runs with varying random seeds.

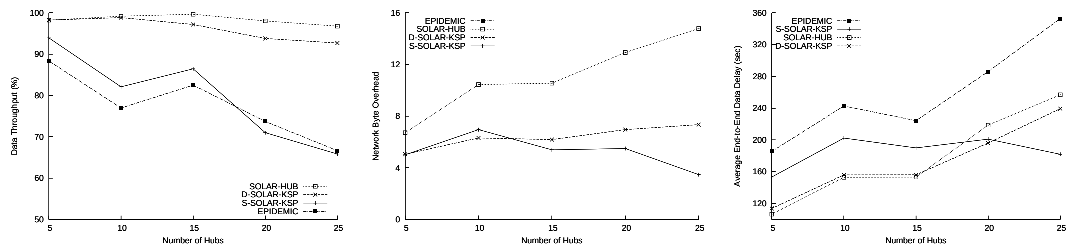
7.3.1 Variation in Number of Hubs

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

Data Throughput: As seen in Figure 7.5(a), the data throughput of SOLAR-HUB is the most consistent with a varying number of hubs. This is attributed to the assumption that each node knows of the next hub it is going to visit, which aids in the next hop selection process. In the case of the KSP based SOLAR protocols: D-SOLAR-KSP and S-SOLAR-KSP, increasing number of hubs decreases the node density within hubs and reduces the likelihood of finding suitable next hops with higher delivery probability to the destination. Amongst the two however, nodes in D-SOLAR-KSP have more freedom of choice in selecting the next hop, and can make best use of whichever nodes are available within radio range. Thus, the throughput of D-SOLAR-KSP is comparable to that of SOLAR-HUB. In S-SOLAR-KSP however, since nodes only forward to other nodes within their N_{next}^{ksp} set, decreasing node densities significantly decrease the probability of finding such nodes in the same hub within a specified time, thereby adversely affecting their throughput with increasing number of hubs. Similarly in EPIDEMIC, since the only way of data dissemination is via data exchange amongst neighbors, decreasing node densities within hubs causes the throughput to decrease. Moreover, since in our Probabilistic Orbit model, hubs do not overlap, nodes in EPIDEMIC cannot communicate with other nodes in other hubs, as they are far apart.

Network Byte Overhead: From Figure 7.5(b), we note that the network byte overhead incurred by SOLAR-HUB is the highest and it keeps increasing with increasing number of hubs. This is easy to understand since in SOLAR-HUB nodes try and forward a copy of a packet whenever they enter a new hub. Thus, as the number of hubs increases, the hub list size of each node grows and more packets get forwarded in the network. In D-SOLAR-KSP, the overhead is seen to be less than SOLAR-HUB primarily due to the reason that in D-SOLAR-KSP, it is possible for two nodes to forward different copies of the same packet for a specific destination to the same neighbor in a hub, with highest delivery probability to the destination. In this scenario, that neighbor will drop one of the packets and only forward one copy onward, reducing the total number of packet transmissions in the network. In SOLAR-HUB however, since packets are forwarded to the "most visited" and "second most visited" hubs of a destination, such dropping of duplicate packets at intermediate nodes are less frequent. S-SOLAR-KSP is more or less consistent with varying number of hubs since it only tries to forward to nodes within N_{next}^{ksp}, which is independent of the hub list size of nodes. However, with large number of hubs, the probability of meeting with favorable neighbors reduce considerably, reducing the the total data packets forwarded, which in turn lowers the overhead. The overhead in EPIDEMIC could not be included in this same graph as we found it to be an order of 100 more than any of the SOLAR protocols. In EPIDEMIC, a pair of nodes exchange data whenever they have space in their buffer and each has some data packet that the other does not. Thus, the number of data exchanges are enormous, leading to such high overhead.

Average End-to-End Data Delay: As seen in Figure 7.5(c), the end-to-end data delay in S-SOLAR-KSP is not much affected by the number of hubs. This is due to the fact that nodes in S-SOLAR-KSP only use the pre-computed KSPs that do not depend on the number of hubs. In D-SOLAR-KSP however, the dynamic and greedy next hop selection favors the end-to-end data delay when there are only a small number of hubs by making sure that the packet never waits for any particular neighbor to appear. However, when the number of hubs grow larger, such a greedy selection may cause the packet to reach the destination via a longer path (when compared to S-SOLAR-KSP), or may suffer delays at some intermediate node in the dynamically formed path, leading to higher end-to-end delay. In SOLAR-HUB, longer hub lists may cause the destination to visit the "most visited hub" less frequently, leading to higher delays. EPIDEMIC displays the highest end-to-end delay since it only relies on eventual dissemination of the data packet to the destination, which may take a long time when number of hubs are large and nodes meet in hubs less frequently.



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

Figure 7.5: Protocol Performance vs. Number of Hubs

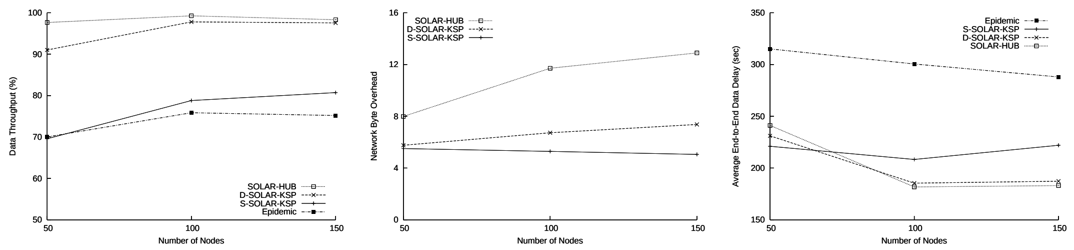
7.3.2 Variation in Number of Nodes

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

Data Throughput: From Figure 7.6(a), it is clear that as the number of nodes increases, all protocols benefit, for their own respective reasons. With a fixed number of hubs, an increase in the number of nodes also increases the node density within hubs, which as noted in the discussions in Section 7.3.1 positively affect all protocol performance. The relative performance amongst the protocols are similar to that seen in Figure 7.5(a) with SOLAR-HUB and D-SOLAR-KSP performing much better than S-SOLAR-KSP and EPIDEMIC. Since nodes in S-SOLAR-KSP only wait for k of its neighbors in N_{next}^{ksp} to come within radio range, it cannot fully leverage the increase in the number of nodes. On the contrary, higher node density provides the greedy approach in D-SOLAR-KSP with more choices of next hop, leading to better performance than S-SOLAR-KSP. In EPIDEMIC, larger number of nodes will cause a larger number of data exchanges. Given a fixed cache size this can lead to eventual dropping of packets, limiting the throughput to a low value with increasing number of nodes.

Network Byte Overhead: The relative performance of the overhead of all the protocols shown in Figure 7.6(b) is also similar to that seen in Figure 7.5(b), and much for the same reasons. With an increased node density within hubs, nodes in SOLAR-HUB almost always find k neighbors to forward to, leading to an increased overhead. D-SOLAR-KSP displays lower overhead than SOLAR-HUB for reasons discussed for *Network Byte Overhead* in Section 7.3.1. S-SOLAR-KSP has the lowest overhead since it only forwards to at most k pre-selected nodes, that may or may not appear within radio range in a specified timeout period. EPIDEMIC once again was omitted from this graph due to its much larger values of overhead.

Average End-to-End Data Delay: The end-to-end delay for all protocols decrease on an average with a larger number of nodes, as seen in Figure 7.6(c). Although, EPIDEMIC protocol eventually delivers data, due to its lack of direction (toward the destination node or hub) it suffers high end-to-end delay. S-SOLAR-KSP shows consistent performance by only selecting nodes from N_{next}^{ksp}, which is still bounded by k, and thus independent of the total number of nodes. On the other hand, increasing node density within hubs appear to cause significant improvements in the end-to-end delay for nodes in both SOLAR-HUB and D-SOLAR-KSP, as nodes find larger number of suitable nodes to efficiently forward their data packets.



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

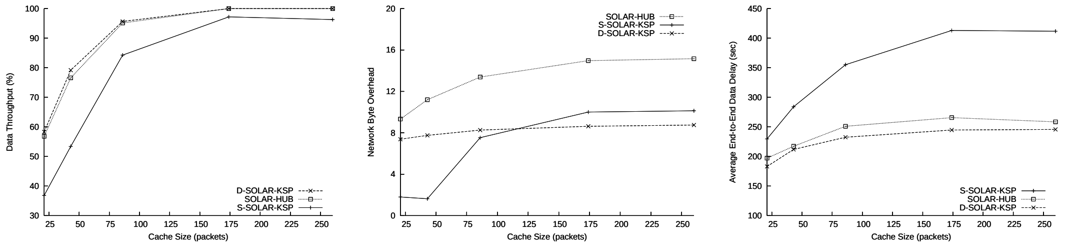
Figure 7.6: Protocol Performance vs. Number of Nodes

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

7.3.3 Effect of Cache Size

Here we study the effect of the maximum number of packets that a node can hold on to simultaneously, both for itself and for other nodes, referred to as the *Cache Size*. In this scenario, the time to live for packets in cache was set to the simulation time to effectively have no timeout, while the *Cache Size* was varied. As seen in Figure 7.7(a), the performance of all our SOLAR protocols improve with increasing cache size. With sufficient amount of storage space in all nodes, we see that both D-SOLAR-KSP and SOLAR-HUB are able to attain 100% throughput via their dynamic selection of next hop from amongst the available neighbors. Although, the throughput in S-SOLAR-KSP increases with larger cache sizes, it is still shy of 100% indicating some packet loss. This is easy to understand given the fact that in S-SOLAR-KSP, nodes wait and forward only to nodes in their N_{next}^{ksp} set. In most cases, the *contact probability* p of a node with those nodes in N_{next}^{ksp} within the simulation time period is less than 1. Thus, with probability $1-p$ packets will be dropped at a node when none of the expected nodes come within radio range, preventing the throughput from reaching 100%. It is obvious that given infinite time, S-SOLAR-KSP will be able to forward all packets eventually. This is the same reason why in Figure 7.7(c) we find the average end-to-end delay in the dynamic SOLAR protocols: D-SOLAR-KSP and SOLAR-HUB, to be far less than that in S-SOLAR-KSP, which waits to forward only to specific nodes. Also, as mentioned earlier in our performance analysis of *Static SOLAR-KSP* in Section 7.1.1, nodes uniformly send data to other nodes over both short and long paths, with the shorter paths having more probability of success. Hence, an increase in the throughput (with increasing cache size) also indicates more data delivered over longer paths, which in turn leads to an increase in the average end-to-end data delay for S-SOLAR-KSP, as seen in Figure 7.7(c).

As the cache size increases, each node can store more packets and can in turn forward more packets, leading to increased overhead as shown in Figure 7.7(b). Since nodes in S-SOLAR-KSP wait to forward to specific nodes, the number of transmissions is lower for small cache sizes, leading to lower overhead than the other two SOLAR protocols. However, as the cache size increases to hold more packets, S-SOLAR-KSP may have multiple transmissions of the same packet within the same hub, unlike in D-SOLAR-KSP and SOLAR-HUB, where nodes forward packets only when they move into a new hub. This explains the sudden rise of overhead in S-SOLAR-KSP with increasing cache size. However, this increase levels off beyond a certain point when the probability of finding a node within N_{next}^{ksp} provides an upper bound to the number of transmissions in a hub, making it independent of the cache size.



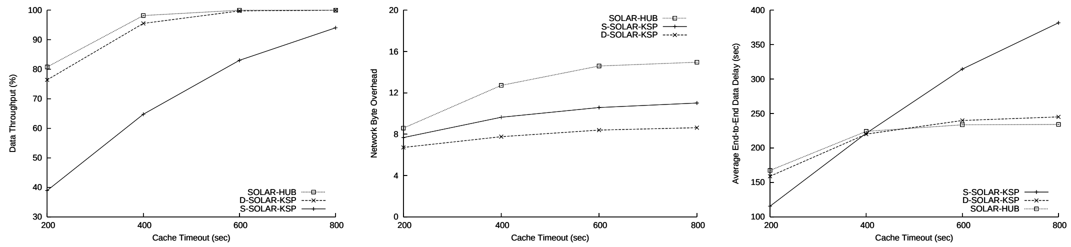
(a) Data Throughput (%) (b) Network Byte Overhead (c) End-to-End Data Delay (sec)

Figure 7.7: SOLAR Performance vs. Cache Size

7.3.4 Effect of Cache Timeout

Here we study the effect of the maximum time for which any packet is cached within a node, referred to as the *Cache Timeout*. In this scenario, the *Cache Size* was set to 650 packets, while the *Cache Timeout* was varied. As seen in Figure 7.8(a), even with a moderate cache timeout D-SOLAR-KSP and SOLAR-HUB perform well with around 80% throughput. This is because both these protocols keep selecting their next hops from the available set of neighbors, unlike S-SOLAR-KSP, which has to wait for specific nodes in N_{next}^{ksp}, and consequently has a much lower throughput with a low cache timeout value. However, with increasing cache timeout, the probability of nodes in S-SOLAR-KSP to find nodes in N_{next}^{ksp} within that timeout period increases, thereby increasing the throughput significantly. The corresponding average end-to-end data delay incurred by the protocols also show proportional results in Figure 7.8(c), with higher average delay for higher throughput. The reason is similar to that described before, where more packets getting delivered indicate greater number of packets being delivered over longer paths (and hence with longer delay), which increase the average end-to-end delay. More specifically, for nodes in S-SOLAR-KSP this effect of increase in cache timeout is more significant since each node waits for that time period to try and forward packets. Just as this improves the probability of finding expected nodes, thereby increasing the throughput as seen before, it also increases the average end-to-end delay by an appreciable amount.

The relative overhead performances of the 3 protocols in Figure 7.8(b) is seen to be similar to that shown in Figure 7.7(b), and for similar reasons. SOLAR-HUB has more number of transmissions at the intermediate nodes than D-SOLAR-KSP due to reasons discussed for *Network Byte Overhead* in Section 7.3.2. The overhead in S-SOLAR-KSP however, shows less drastic changes with varying cache timeout than that seen for varying cache size in Figure 7.7(b). This is because the lowest cache timeout we considered (200 seconds) was more than the average hub stay time (exponentially distributed with mean of 50 seconds), giving every node a fair opportunity to meet with nodes in their N_{next}^{ksp}. Overall, given more time every protocol has a higher probability of meeting suitable nodes for forwarding packets, thereby displaying a general increase in network overhead with increasing cache timeout.



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

Figure 7.8: SOLAR Performance vs. Cache Timeout

7.3.5 Note On Mobility Profile Lifetime

A priori knowledge of the mobility profile for all users presents some scalability issues: each user can only maintain mobility profile information regarding a finite number of users. A solution to this problem is to have each user keep a buffer containing mobility profiles for a set number of users, and to flush out older mobility profiles as new ones are required.

In all the variants of SOLAR introduced here, if the source does not have the mobility profile of its intended correspondent, it can request it from its neighbors by issuing a profile request. The neighbors can then forward the request to their own neighbors if they do not possess the requested profile, or provide the profile if they do. In [133] it is shown that the probability to have a positive answer to the request depends on the length of time each profile is kept by a user and on the maximum number of hops the request is forwarded. It can be ensured that the request is satisfied with high probability by setting the maximum of hops to the right value. The maximum number of hops also sets a maximal delay for the request to be satisfied.

7.4 Summary

In this chapter, we have addressed the issue of efficient location management and routing being studied under the constraints of network disconnections, common within Intermittently Connected Networks (ICN). The unavailability of contemporaneous end-to-end path from a source to a destination through intermediary peers renders most reactive and proactive routing protocols useless. Although, some work [67] has been suggested in literature that assumes deterministic mobility, the actual motion of users in real life are not as predictable. In the absence of any practical knowledge oracle to provide up-to-date network connectivity information, the routing problem remains a daunting one. To that end, researchers in [28, 58, 138] have also proposed methods to either exploit or influence the network mobility (e.g., by introduction of foreign mobile agents into the ad hoc network and controlling their motion) to improve network efficiency. However, such methods may not be too practical (e.g., foreign agents may not always be freely available and their ready deployment may not always be feasible).

In this chapter, we have applied our previously validated concepts of sociological orbits involving a set of socially significant places (or hubs), that is well suited for semi-deterministic node mobility within a special class of ICN formed of mobile ad hoc users called ICMAN. We have proposed a series of *Sociological Orbit aware Location Approximation and Routing (SOLAR)* algorithms, that leverage upon the underlying orbital mobility information to route data within an ICMAN more efficiently than other conventional routing approaches (e.g., Epidemic routing [128]) within an ICMAN, in terms of higher data throughput, lower network overhead, and lesser end-to-end data delay. We have already established the simplicity and efficiency of using our SOLAR framework in a Mobile Ad Hoc Network (MANET) [50, 51]. In this chapter, we have presented equally strong results to prove that our proposed SOLAR is as strong a candidate of choice when it comes to meeting the challenges of routing within an ICMAN.

Chapter 8

Theoretical Modeling and Analysis

So far in this thesis, we have proposed techniques for profiling users based on their mobility and analyzed several SOLAR routing strategies that make use of these mobility profiles and their pairwise contact probabilities. In this chapter, we aim to provide in detail a mathematical model for computing the contact probabilities and subsequently the delivery probabilities that were used by our SOLAR protocols, based on our mobility profile information and the hub transitional probabilities. To facilitate our discussion, we shall define a couple of terms:

1. *contact probability*: The probability of two nodes ever coming within each other's radio range (in contact) during the entire observed time window.

2. *delivery probability*: The probability of a source delivering a packet to a destination via all possible paths of intermediary nodes that come "in contact" with their predecessors and successors in their respective paths.

The rest of this chapter is outlined as follows. We start off by presenting a mathematical model for computing the pairwise user *contact probability* in Section 8.1. We then formulate our novel routing problem in Section 8.2 and discuss its complexity in Section 8.3. We present an approximation algorithm for computing the delivery probability in Section 8.4 and compare its performance with an optimal solution. In Section 8.5, we propose and analyze an edge-constrained routing protocol called EC-SOLAR-KSP, which makes use of contact probabilities, and compare its performance with other probabilistic and epidemic routing schemes proposed in literature for intermittently connected networks. Finally, we summarize this chapter in Section 8.6.

8.1 A Model for Computing Contact Probability

In this section, we present a model for computing contact probability of network nodes in the context of intermittently connected mobile ad hoc networks. The model is based on experimental data we gathered from real-world user movements' traces. Then, in the next section, the effectiveness of this model is illustrated by its usage in our routing algorithm.

Consider a node X whose set of hubs is $\mathcal{S}$. From our earlier work in [44] we have verified that X's staying time at a hub $h \in \mathcal{S}$ roughly follow a power law distribution with exponent λ_h^X. After staying at h, X moves to another hub $h' \in \mathcal{S}$ with hub transitional probability $\beta_{hh'}^X > 0$. Obviously,

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

In the data set we analyzed in [44], we were unable to gather enough information about the inter-hub transition time. Hence, in our model the time it takes for X to move from h to h' is assumed to be exponentially distributed with parameter $\lambda_{hh'}^X$. This assumption can be relaxed/changed when there is better experimental data on users' mobility. The model proposed in [77] can be used, for instance. The following analysis can easily be modify to adopt the new and supposedly more correct distribution of hub transition times.

The movement pattern can be modeled with a Semi-Markov Chain (SMC) [68] whose embedded Markov chain (EMC) has state space

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

Here, the states (h,h') represent X being on the move from hub h to hub h'. The sojourn times at the "hub states" h are power-law distributed and the sojourn times at the "hub transition states" (h,h') are exponentially distributed. The transitional probability p^X_{ij} of the corresponding EMC can then be computed as

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

where $i, j \in I_X$ and $h,h' \in \mathcal{S}, h \neq h'$. The states in this SMC when node X moves from hub h to hub h' are illustrated in Figure 8.1.

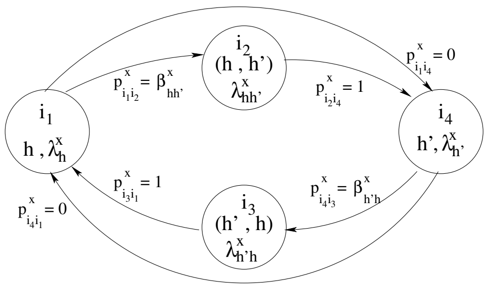


Figure 8.1: States of Markov Chain for movements between hubs h and h'

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

Let I_X and I_Y be the state spaces of the SMCs capturing the movements of X and Y, respectively. In order to track the relative positions of X and Y together, define a SMC $\{Z_t \mid t \geq 0\}$ which is the Cartesian product of the SMCs for X and Y. For any $x \in I_X, y \in I_Y$, the sojourn time of the state (x,y) of Z has the distribution of the random variable which is the minimum of the sojourn times of x and y. Since the sojourn times at x and y are either exponential or power-law with known parameters, it is easy to compute the distribution of the sojourn time at (x,y). We omit the detailed formulae, which come from relatively simple exercises.

The corresponding EMC of (Z_t) has state space $I = I_X \times I_Y$. To characterize this process, we need to compute the jumping probabilities from a state (i_1, i'_1) to another state (i_2, i'_2), where either $i_1 = i_2$ or $i'_1 = i'_2$. (With probability zero the two chains for X and Y jumps at the same time.) Consider any state (i,i'). If the sojourn time T_i at state i of X is smaller than the sojourn time $T_{i'}$ at state i' of Y, then with probability p^X_{ij} the chain moves to state (j,i'), for some $j \in I_X$. Conversely, if $T_i > T_{i'}$, then the chain moves to state (i,j'), for some $j' \in I_Y$. Consequently, we can compute the jumping probabilities of the EMC for (Z_t) by conditioning on the event $\{T_i > T_{i'}\}$ and its complement. Again, we omit the detail case-by-case formulae.

The states of X and Y can be assumed to be overlapping. (If the states are not overlapping, X and Y will never meet.) The EMC of the Cartesian-product SMC as defined above is ergodic as long as the EMC for X and Y are ergodic (under most normal circumstances, otherwise we can disturb the chain by adding a few transitions with infinitesimal probabilities). In this case, we can easily compute the occupancy probabilities at equilibrium of any state (i,j) of the product chain by solving for the stationary distributions of the EMCs of X and Y. We are interested in only the occupancy probabilities $\pi^{XY}(h,h)$ of the states (h,h) where $h \in \mathcal{R}$. This is precisely the probability that

X meets Y in h at equilibrium. Finally, the probability that X meets Y at equilibrium is the sum of $\pi^{XY}_{h,h}$ over all $h \in \mathcal{R}$.

Suppose X holds a packet it would like to transmit to a downstream neighbor towards a destination. It cannot hold the packet forever due to limited buffer size (and possibly delay requirements). Some routing strategy may require X to try its best to deliver the packet to (some of) the best neighbor(s) within a pre-defined time interval T. Consequently, given a time interval T and given that X is in some hub $h \in \mathcal{R}$, we are also interested in the probability that Y will be in h within T. Computing this probability is the same as computing the densities of the hitting times of the SMC corresponding to Y (probability that Y hits h given some initial distribution). There is no known general formula. Computationally however, there are methods to compute these densities using Laplace transforms [59] for larger chains or uniformization [20] for smaller chains.

8.2 A Routing Problem in Probabilistic Graphs

In this section, we will rigorously formulate this problem. A good solution to this problem can be used to devise provably (near) optimal solution to the probabilistic routing problem. Moreover, it can also be used as a benchmark to compare probabilistic routing protocols in the literature.

Define a directed graph $G = (V,E)$ whose vertices represent nodes in the network under consideration. For each pair of nodes u and v, let $p_{(u,v)}$ denote the probability of u being able to deliver a data packet to v given some practical constraint(s) and/or some mobility model for nodes' movements in the network. For simplicity, we assume that all these contact probabilities are independent. If not, the problem becomes too complex to be useful. This assumption is not too restrictive, as we will demonstrate with our routing protocol in a later section.

(Note that, in this section we are not yet concerned about the question of how to compute $p_{(u,v)}$. The problem of computing the contact probabilities is orthogonal to the routing problem on random graphs that we are formulating. In a later section, we will present a model for estimating these probabilities.)

The edges of G are precisely those pairs $e = (u,v)$ for which $p_e > 0$. Consider a source s and a destination t in G, and the problem of finding the best way to deliver a packet from s to t in the network. The obvious objective is to maximize the delivery probability of the packet. A broadcast (epidemic-like) routing algorithm seems to be best in terms of maximizing the delivery probability; however, broadcasting imposes a high cost in terms of data and processing overhead. It is thus natural to formulate a problem investigating the tradeoff between delivery probability and overhead.

Consider any routing algorithm/protocol A run by all nodes in the network. Let $G[A] = (V, E_A)$ denote the subgraph of G induced by A, i.e. (u,v) is an edge of $G[A]$ if there is a possibility that u delivers a packet to v under A. For instance, if A is a naive broadcast strategy where each node delivers a packet it receives to all nodes it meets within a time interval T, then (u,v) is an edge of $G[A]$ if the probability that u meets v within T is positive. We will refer to $G[A]$ as the *delivery subgraph* induced by A. Note that $G[A]$ along with the probabilities $p_e, e \in E_A$ define a probability space of random graphs (the Erdős-Rényi model $\mathcal{G}(n, \{p_{u,v}\})$ [21]). The probability that A successfully delivers a packet from s to t is the probability that there is a (directed) path from s to t in a random graph H chosen from this space. This probability is often denoted by $Conn_2(G[A])$ (or $Rel_2(G[A])$ for undirected graphs) in the network reliability literature [37]. The notation implicitly assumes the source s and the destination t to be known in advance. Also in this literature, $G[A]$ is called a *probabilistic graph*, so is G for that matter.

To this end, we have $Conn_2(G[A])$ as the objective function of our problem. We next define the constraints. As we have mentioned earlier, the key to the problem is the tradeoff between delivery probability and the data overhead. The maximum number of packets that A could produce is precisely the number of edges of $G[A]$. Thus, a very natural constraint to our optimization problem is to give a threshold k on the number of edges of $G[A]$.

To summarize, the centralized version of our problem can be formulated as follows. We are given a probabilistic graph G, i.e. a graph along with a probability function $p : E(G) \to [0,1]$, where p_e represents the probability that a packet can be delivered along edge e at a random point in time. The problem is to choose a delivery subgraph $G[A]$ of G with at most k edges, such that the probability $Conn_2(G[A])$ that there is a path from s to t in a graph H chosen randomly from the probability space defined by $G[A]$ is maximized. To the best of our knowledge, this problem and its complexity has not been addressed in the literature. Also, there is the corresponding distributed version of the problem where the delivery subgraph $G[A]$ is to be constructed distributively, yielding a routing protocol.

In this chapter, we will present a practical solution to this problem in the context of ICMAN, involving the following steps: (a) proving the hardness of our optimization problem and the hardness of computing the objective function $Conn_2(G[A])$ (the hardness of $Conn_2(G[A])$ is well-known, but that does not imply the hardness of the optimization problem); (b) giving an algorithm to approximate $Conn_2(G[A])$; (c) devising a mobility model to estimate the contact probabilities p_e; and (d) designing a routing protocol for the problem based on the contact probabilities computed in step (c) and insights obtained from step (a), and showing the protocol's effectiveness by comparing it with other protocols for probabilistic routing.

Remark: if we also consider using Erasure Codes [114] for dada transmission, we can add an additional constraint in terms of the code rate, keeping the objective function the same. Some previous work on routing in DTN has considered this dimension [91, 131].

8.3 Complexity of the Routing Problem

In this section, we will investigate the complexity of the problem defined in Section 8.2. The problem is to find a subgraph $G[A]$ of G so that $Rel_2(G[A])$ is maximized.

Unfortunately, computing the connectedness probability in a random graph is very hard (even for graphs with bounded degree like in our case). There is a vast literature on this problem. In the probabilistic sense, for example, Chapter 7 of [21] contains a partial set of references. It is unlikely that this probability is a simple function [80]. In the computational complexity sense, the problem is #P-Hard, as it is precisely the well-known reliability problem for two terminals [37]. A minor point: when G is directed, the problem is often referred to as the *s,t*-connectedness problem denoted by $Conn_2(G)$. In the classic paper [129] Valiant was the first to establish that both $Rel_2(G)$ and $Conn_2(G)$ are #P-complete [70].

The optimization problem, however, may not be hard even though computing the objective function is hard. This point is a little bit subtle. Given an integer a, computing the function $f(a) = 2^a!$ takes exponential time; yet, the optimization problem of finding which member a of a set A of integers has the largest $f(a)$ has the same complexity as sorting.

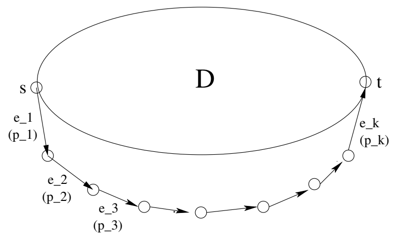


Figure 8.2: Construction of G for proving the #P-hardness of maximizing *s,t*-connectedness problem in random graphs

To this end, we consider the optimization problem of finding a delivery subgraph $D = G[A]$ with at most k edges which maximizes $Conn_2(D)$. We will show that[1]

Theorem 8.3.1 *The routing problem on random graph is #P-hard.*

Proof: We will reduce the 2-terminal connectivity problem ($Conn_2$) itself to our problem. Consider a generic instance of $Conn_2$ where we are given a directed graph $D = (V,E)$ along with a source s, a destination t, and all the edge probabilities p_{ij}. The problem is to compute the probability $Conn_2(D)$ that there is a path form s to t in a random graph chosen from the probability space defined by D. Note that all p_{ij} are rational numbers represented by a numerator and a denominator which are integers. Let c be the least common multiple of all the denominators, then the number of bits to represent c (i.e. $\log_2 c$) is certainly polynomial in the input size.

Consequently, if we have a procedure to decide if $Conn_2(D) \leq c'/c$ for any integer $c' \leq c$, then we can compute $Conn_2(D)$ by a simple binary search. We will prove that an algorithm solving our optimization problem can be used to decide if $Conn_2(D) \leq c'/c$.

Construct a graph G as shown in figure 8.2, where the upper part is D itself, and the lower part is a simple path from s to t consisting of exactly $k = |E(D)|$ edges. The edge probabilities of the upper part is the same as those of D. The edge probabilities of the edges $e_1,\ldots,e_k$ of the lower part is chosen so that $p_1 \cdot p_k = c'/c + \varepsilon$, where $\varepsilon < 1/c$. Our optimization problem is to compute a subgraph H of G with $|E(H)| \leq k$ so that $Conn_2(H)$ is maximized.

Let A be any algorithm solving our problem. It is easy to see that A will either return the upper part or the lower part of G. If A does return the lower part, then $Conn_2(D) > c'/c$; otherwise, $Conn_2(D) \leq c'/c + \varepsilon$. But, by the way c was chosen, $Conn_2(D)$ is exactly a multiple of $1/c$; hence, A returning the upper part implies that $Conn_2(D) \leq c'/c$. Consequently, A can be used to decide if $Conn_2(D) \leq c'/c$, implying

[1] We thank Prof. Charles Colbourn for fruitful discussions leading to this proof.

that our problem is at least as hard as $Conn_2$ by the analysis above. ■

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

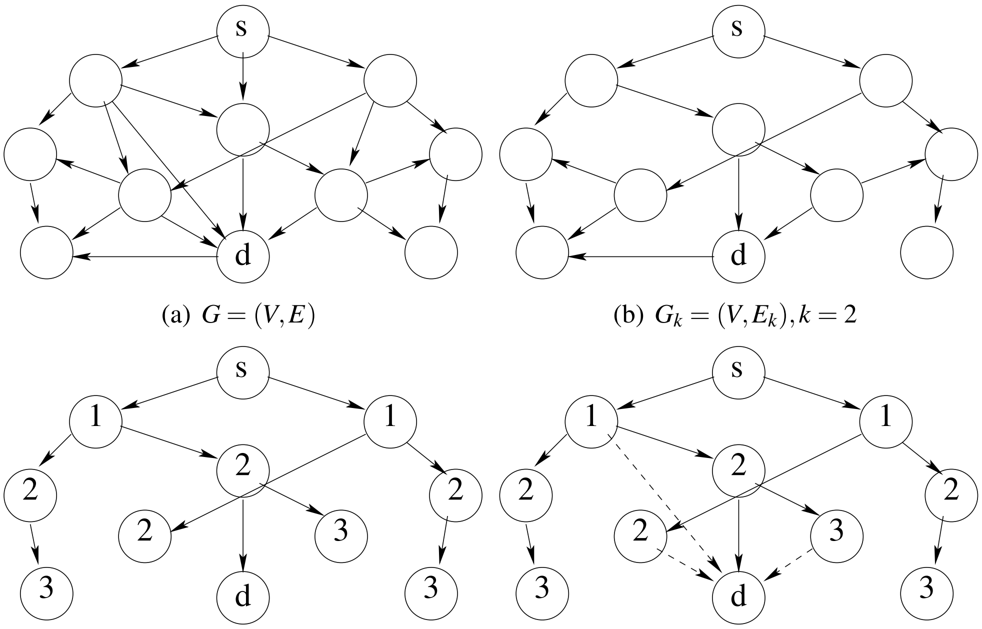
1. Find a polynomial-time computable function $p(G[A])$ to approximate $Conn_2(G[A])$ well. Then, devise an algorithm A that maximizes $p(G[A])$. Note that, $p(G[A])$ can also be used to compare the outputs of different routing algorithms; thus, it is useful in its own right whether or not we can devise an algorithm optimizing $p(G[A])$.

2. Find a routing strategy (heuristic) A for which $Conn_2(G[A])$ can be reasonably computed or estimated.

In the following sections, we present our results on both approaches.

8.4 Approximation Algorithm for Computation of Delivery Probability

In the light of the discussion in the previous section, we propose an approximation algorithm for computing the delivery probability from source s to destination d in a network that is modeled as mentioned before: a directed graph $G = (V,E)$, where edge e exists between two nodes u and v with probability $p_e(u,v)$ = *contact probability* of u and v, as shown in Figure 8.3(a). First, we construct another graph $G_k = (V,E_k)$ from the graph G by having each node (starting from s onwards) choose at most k edges to downstream neighbors, and deleting all other edges not chosen, as shown in Figure 8.3(b). Second, we modify the weight of each edge in G_k to be $w_e = -1 * \log(p_e(u,v))$ for all nodes u and v, and call this new graph as G'_k. Third, we construct a shortest path tree $G_{sp} = (V,E_{sp})$ from G'_k as shown in Figure 8.3(c), and assign a level



(a) $G = (V,E)$

(b) $G_k = (V,E_k), k = 2$

(c) $G_{sp} = (V,E_{sp})$

(d) $G' = (V,E')$ where dotted edges are *special edges*

Figure 8.3: Steps in preparing a network graph for the application of Approximation Algorithm 1

number to each node in a breadth first manner. Fourth, we replace the weight of each edge w_e in G_{sp} with $p_e(u,v)$, as in the original graph G. Finally, we add *special edges* (dotted edges in Figure 8.3(d)) between any node v and destination d in graph G_{sp} that were connected by an edge $e \in E$ in the original graph G, to get our *delivery subgraph* $G' = (V,E')$.

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

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

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

Algorithm 1 : Approximation of Delivery Probability

```
Input ← G = (V,E), s, d
P^d(d,d) ← 1
L ← maximum assigned level number
while L ≥ 1 do
    for all u ∈ V, u ≠ d with assigned level number L do
        i ← 1
        for all outgoing edge e ∈ E from u do
            v ← head of edge e
            p ← weight of edge e * P^d(v,d)
            P[i] ← p
            i ← i + 1
        end for
        p1 ← 1
        for j ← 1 to (i − 1) do
            p2 ← 1 − P[j]
            p1 ← p1 * p2
        end for
        P^d(u,d) ← 1 − p1
        if u = s then
            print P^d(s,d)
            exit
        end if
    end for
    L ← L − 1
end while
```

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

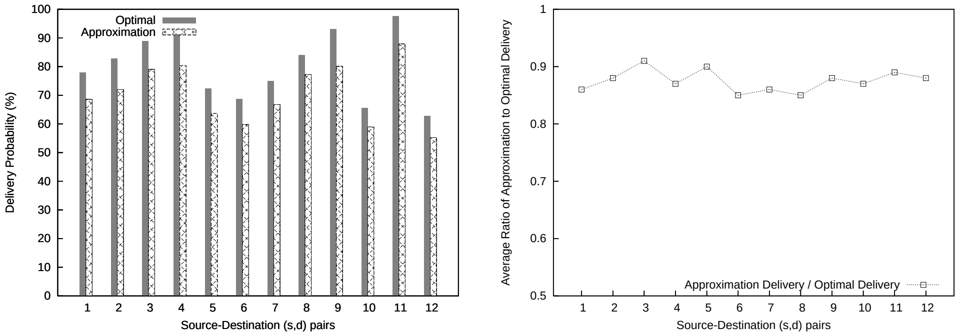
1. Calculate all possible paths from s to d

2. Apply Algorithm 2 to compute the delivery probability by rules of inclusion and exclusion

Algorithm 2 : Optimal computation of Delivery Probability

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

We simulated using Matlab [66] a small directed graph with 25 nodes with a given distribution of pair-wise contact probabilities $p_e(u,v)$ to evaluate the performance of our suggested approximation algorithm in comparison to the optimal algorithm. We chose 12 distinct source-destination $\{s,d\}$ pairs and only computed the delivery probabilities $P^d(s,d)$ through the two algorithms, without sending any actual traffic. Figure 8.4(a) shows the results of our simulation runs where our approximation algorithm is seen to perform within 88% of the optimal algorithm on an average. For each $\{s,d\}$ pair, we further simulated for 20 different pair-wise contact probability matrices. Figure 8.4(b) shows the relative performance of the approximation algorithm with respect to the optimal algorithm for each $\{s,d\}$ pair, averaged over all the runs with different contact probability matrices for that $\{s,d\}$ pair. Once again we find our approximation algorithm to perform within 85% to 90% of the optimal solution.



(a) Absolute performances with fixed contact probability matrix

(b) Relative performance averaged over different contact probabilities

Figure 8.4: Performance comparison of our approximation algorithm for delivery probability with the optimal solution

8.5 Edge Constrained SOLAR Heuristic

In this section, we present a Sociological Orbit aware Location Approximation and Routing (SOLAR) heuristic that makes use of mobility profile and hub transitional probability based computations for contact and delivery probability.

8.5.1 Edge-Constrained SOLAR-KSP

We chose our Static SOLAR KSP (S-SOLAR-KSP) algorithm proposed in [45] to form the base of this heuristic, with some additional modifications. In general, in this version of user-level routing protocol SOLAR-KSP, we assume that each user knows of every other user's mobility profiles and each user distributively does the following: First, every user computes the *contact probability* with every other user. In this work, we compute these probabilities based on the simulated mobility traces, as opposed to other various ways suggested in [65, 74, 88, 121] for example. Second, we represent the contact information between all users as a complete weighted graph $G = (V, E)$, where V

is the set of all the users, and E is the set of weighted edges between every pair of users that have at least one hub in common. Let $P(u,v)$ be the contact probability of users u and v. Then the weight of edge (u,v) is given by $w(u,v) = \log(1/P(u,v))$. Now whenever a source user s has a packet to forward to destination user d, it applies the Algorithm 3 on the weighted graph to find a delivery subgraph to d that has at most $L(\leq |E|)$ edges. In other words, s iteratively uses Dijsktra's Shortest Path algorithm [41] to find the shortest path from s to d. In a single iteration, if the number of new edges encountered on the shortest path (that are not already in the delivery subgraph formed so far) is less than the remaining number of edges allowed under the edge constraint, then those edges are added to the delivery subgraph and the edge constraint parameter is adjusted accordingly. Also, to ensure that a new path is formed in the next iteration, the lowest weighted edge in the shortest path is deleted from the working graph. This algorithm terminates either if the working graph is exhausted, or if no more path exists between s and d, or if the edge constraint is met. Once this delivery subgraph is obtained, the source inserts this additional information into the header of all the packets and waits for all the next hop neighbors on all paths to appear. The intermediary users keep forwarding in accordance with this delivery subgraph till the destination is reached. We shall refer to this edge constrained SOLAR version as **EC-SOLAR-KSP**.

8.5.2 Performance Comparison Results

We compare the performance of EC-SOLAR-KSP with that of a probabilistic routing (referred to here as **PROB-ROUTE**) based on [93] and Epidemic Routing (referred to as **EPIDEMIC**) [128]. In PROB-ROUTE, users each go through the "initialization" phase whenever they meet another user, where upon their contact probability is updated. When a pair of user do not meet for long, their corresponding contact probability is "aged". Also when a source has a packet to send to a destination it may calculate

Algorithm 3 : Edge constrained delivery subgraph

Input $\leftarrow$ Complete weighted graph $G = (V, E)$,
  Edge constraint L, source s, destination d
$DSG = (V', E') \leftarrow$ initial delivery subgraph
  with $V' \leftarrow V$ and $E' \leftarrow NULL$
while $|E|$ **do**
  apply Dijkstra's Shortest Path algorithm [41]
  to find shortest path from s to d i.e., $SP_{s,d}$
  if $SP_{s,d} ==$ NULL **then**
    return DSG
  end if
  if $|\{e \mid e \in SP_{s,d},\ e \notin E'\}| \leq L$ **then**
    $L \leftarrow L\ -\ |\{e \mid e \in SP_{s,d},\ e \notin E'\}|$
    $E' \leftarrow E' \bigcup \{e \mid e \in SP_{s,d},\ e \notin E'\}$
  end if
  if $L == 0$ **then**
    return DSG
  end if
  $E \leftarrow E -$ the edge in $SP_{s,d}$ with least weight
end while
return DSG

transitional probability through other users. The reader is referred to [93] for the detailed description and equations. In our implementation of PROB-ROUTE we used a value of 0.5 for all three parameters P_{init}, γ, and β, and allowed each user to forward a copy of a packet to at most 3 different neighbors with higher delivery predictability. We consider 3 variations of EC-SOLAR-KSP as well: **EC-SOLAR-KSP1** with $L = |E|$, **EC-SOLAR-KSP2** with $L = 0.8 * |E|$, and **EC-SOLAR-KSP3** with $L = 0.6 * |E|$.

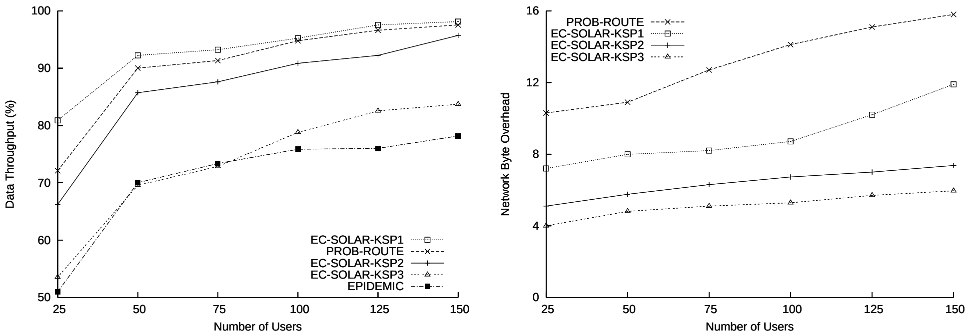
For simulation, we consider an ICMAN built within a campus consisting of several buildings (hubs) in accordance with the findings from our study [44] of an year long wireless users' mobility traces on ETH Zurich campus. In the Probabilistic Orbit model simulated, the users spend most of their time within a number of hubs, and intermittently move between hubs. To model realistic speeds of mobile users within such a network, we consider the work in [78, 134] and fix the Inter-Hub and Intra-Hub time/speed parameters, along with the other simulation parameters and their default values or range of values as shown in Table 7.1. In this work, we only present the protocol performances with a varying number of users as it is a significant factor in ICMAN settings, but we do study other variations as indicated in Table 8.1 which yield similar results.

Table 8.1: Simulation Parameters for EC-SOLAR-KSP Performance Comparison

GENERAL PARAMETERS	
Simulation Duration (each run)	3000s
Terrain Size	1000m x 1000m
Number of Nodes (*Users*)	*Vary*, (Default= 100)
Radio Range	125m
Cache Size	*Vary*, (Default= 200 Packets)
Cache Timeout	*Vary*, (Default= 400s)
MAC Protocol	IEEE 802.11
Mobility Model	Probabilistic Orbit
ORBIT PARAMETERS	
Total Hubs	*Vary*, (Default= 15)
Hub Size	50m x 50m
Hub Stay Time	Power Law (Mean= $10^6/x^3$)
Hub List Timeout	None
Hub List Size	Power Law (Mean= $0.7/x^2$, 1 to Total Hubs)
Inter-Hub Transition Time	Exponential (Mean= 40s)
Intra-Hub Pause	1s
Intra-Hub Speed	1m/s-10m/s
TRAFFIC PARAMETERS	
CBR connections	30 (120 packets each) Random
Data Payload	1460 bytes per packet

We chose 2 metrics to evaluate the performance of each protocol: *data throughput* = (data packets delivered)/(data packets generated); and *network byte overhead* = (total bytes transmitted)/(total data packets delivered);

As seen in Figure 8.5(a), EC-SOLAR-KSP1 performs the best closely followed by PROB-ROUTE. Since EC-SOLAR-KSP1 has the additional knowledge of mobility profile based network connectivity, it is able to compute the delivery probability via neighbors it has not met yet, unlike PROB-ROUTE where users have to meet at least once to update their initial contact probability. EC-SOLAR-KSP2 and EC-SOLAR-KSP3 have decreasing performances because of their increasing edge constraints. Epidemic performs the worst in the face of limited buffer. Also the fact that these results are observed within a limited time of 3000*s* takes its toll on EPIDEMIC, which is capable of "eventually" delivering all packets if time is not a constraint. Overall, all protocols do well with increasing number of users as it helps in finding a larger number of deliverable paths from source to destination. Figure 8.5(b) is not able to show the results for EPIDEMIC which has alarmingly large overhead. Amongst the rest, PROB-ROUTE incurs maximum overhead due to its arbitrary forwarding to neighbors just on the basis of larger delivery predictability. EC-SOLAR-KSP1 has lower overhead than PROB-ROUTE as it only forwards within the edge constrained delivery subgraph. The overhead in EC-SOLAR-KSP2 and EC-SOLAR-KSP3 is seen to decrease due to their increasing edge constraints. Overall, this "edge constraint" parameter gives us a good handle to a tradeoff between the desired throughput and the corresponding overhead.



(a) Data Throughput (%) (b) Network Byte Overhead

Figure 8.5: Protocol performance with varying number of users

8.6 Summary

Mobility of wireless users poses as one of the main challenges in effective routing of packets in any mobile wireless network. The added constraint of intermittent connectivity makes it even more difficult to employ traditional reactive protocols well suited for general Mobile Ad hoc Networks (MANET). Literature suggests several protocols based on the concepts outlined in [121, 128] which aims to infuse information within a network like an epidemic. Others [67, 84, 93] studied network characteristics and proposed probabilistic routing techniques. At the same time, profiling wireless users based on their mobility has also been proven beneficial to routing. The suggested approaches to profiling [85,120] have however varied, as well as the details of the mobility information required. In our previous work [44] we too established a mobility profiling framework based on the sociological influences on wireless users' movements and further demonstrated its use in probabilistic routing within Intermittently Connected Mobile Ad hoc Networks (ICMAN) [45,52].

In this chapter, we have analyzed our probabilistic routing framework mathematically and provided some insight into the computational complexity of both the contact probability and the delivery probability, that is used by our previously proposed routing protocols [45,52]. We have presented a novel formulation of the routing problem as one where given a random graph, we aim to find an optimal routing algorithm that will generate an optimal delivery subgraph so as to maximize the connectivity between a source and destination pair. We have studied and analyzed the hardness of computing such an optimal delivery subgraph given a graph and a routing algorithm. We then proposed an elegant algorithm to approximate the delivery probability of a delivery subgraph and presented its performance study in comparison with the optimal algorithm. We also presented a mathematical model for analyzing mobility and computing the pairwise user contact probabilities. Finally, we proposed an edge-constrained routing algorithm EC-

SOLAR-KSP which makes use of such contact probabilities and highlighted its novelty and superiority over other probabilistic and epidemic routing approaches proposed in literature to address the routing problem within intermittently connected wireless networks.

Chapter 9

Concluding Remarks and Future Direction

Mobility affects wireless network systems both adversely and favorably. The true understanding of the effects of mobility is the key to turning it into an advantage in improving the network performance. Initial work on routing protocols within Mobile Ad hoc NETworks (MANET) were not adequately aware of the impact of the underlying mobility and hence struggled to cope with its adverse effects such as loss of links, network partition, etc. The trend in the initial MANET routing protocols proposed in literature suggested routing optimizations either to counter the effects of mobility, or to adapt to it. We, on the other hand, study the effects of mobility itself in an effort to make it work to our benefit in terms of routing decisions. While recent work on Delay Tolerant Networks (DTN) focuses more on deterministic mobility (e.g., satellites in orbits, busses on fixed routes), we focus more on partially deterministic mobility under sociological influences. We conclude this dissertation by summarizing its major contributions and giving directions for future work.

9.1 Major Contributions

In this research, we have investigated novel mobility aware routing strategies for mobile wireless networks. Our major contributions lie in the design of soft distributed location management, mobility profiling techniques, and mathematical model and analysis of our proposed suite of routing protocols for various types of wireless networks.

Acquaintance based Soft Location Management:

In an initial attempt, we observe the basic property of mobility to encourage scattering of nodes all over the network, and make use of it in a novel concept to build "acquaintances" that not only provide greater network scouting ability to an individual node, but also help by maintaining a distributed database of location of acquaintances. We present results that show the usefulness of acquaintances in both querying for destinations and routing to them more efficiently and with much lower control overhead when compared to conventional routing strategies.

Hierarchical ORBIT Framework and ORBIT Based Routing:

Appropriate analysis and subsequent modeling of node mobility poses as the main challenge in evaluating routing protocol performance. To this end, literature has proposed several entity, group and scenario based mobility models. While some of them cater to short term networks (disasters, military, etc.), others model detailed scenarios. However, there is no model that captures the realistic orbital movement pattern found in our society. Thus, we proposed a novel framework called ORBIT based on intuitive observations on the sociological influences on wireless users' movement that is practical, general and useful. Moreover, it identifies with sociological orbits, and is also able to integrate different mobility models into a single framework. We generate several practical mobility models via the framework and present their analysis. We also design an example Orbit Based Routing (OBR) protocol that is among the first to effectively leverage mobility information for routing packets within a MANET. In particular, we study various versions of this protocol to exhaustively compare different packet transmission strategies (e.g., unicast, multicast, simulcast).

Profiling Mobility and Location Predictions:

To validate our ORBIT framework and its associated sociological assumptions, we analyze an year long mobility trace data obtained as access point (AP) system logs from the ETH Zurich campus. In our analysis of the data involving around $13,000$ users, 500 APs, and 42 buildings we present conclusive proof that each wireless user has a socially influenced list of places (or, hubs) that he/she visits in a regular fashion. Although current literature provides similar analytical work on mobility trace data, ours is the first to focus on the sociological influences on the user level movement at a higher layer of abstraction (i.e., hub level movement and prediction). We propose a novel mobility profiling technique using the Mixture of Bernoulli's distribution to parse sporadic mobility trace data and cluster them in a probabilistic mixture of mobility profiles. We present strong results to show the advantage of using such profiles to perform hub level location predictions (under different scenarios) with 10% to 30% more accuracy than a straightforward statistic based prediction strategy.

Using Mobility Profiles for Routing:

Based on our analytical study and our mobility profiling tool, we propose a novel suit of Sociological Orbit aware Location Approximation and Routing (SOLAR) protocols for MANET. These protocols extend upon the distributed location management based on acquaintances, but replace the user location information with the user mobility profile information. Simulation study shows that profiles can be beneficial to routing in improving performance within MANET without incurring higher protocol overhead. We also study routing issues within Intermittently Connected Mobile Ad hoc Networks (ICMAN) and propose a novel hub-based routing strategy that leverages upon the destination's mobility profile information and delivers packets to the hubs of destination instead of to the destination itself. We compare the performance of this strategy with two other user-based routing strategies based on pair-wise user contact probabilities that is also proposed. Although there has been related work in Delay Tolerant Networks (DTN) that uses the concept of contact and delivery probabilities to route messages, we differ in the process of computing these probabilities, where once more the mobility profiles are used most efficiently.

Formulating and Analyzing a Routing Problem for Probabilistic Graphs: We defined a novel routing problem formulation as follows. Given a probabilistic graph G and routing algorithm A, we wish to determine a delivery subgraph $G[A]$ of G with at most k edges, such that the probability $Conn_2(G[A])$ that there is a path from source s to destination t in a graph H chosen randomly from the probability space defined by $G[A]$ is maximized. To the best of our knowledge, this problem and its complexity has not been addressed in the literature. Also, there is the corresponding distributed version of the problem where the delivery subgraph $G[A]$ is to be constructed distributively, yielding a routing protocol.

In this work, our practical solution to this routing problem is multi-fold: First, we model mobility using a Semi-Markov Chain to estimate the pairwise user contact probabilities; Second, we prove the hardness of our optimization problem of finding a delivery subgraph that maximizes the delivery probability and discuss the hardness of computing the objective function $Conn_2(G[A])$ (which is not the hardness of $Conn_2(G[A])$ itself); Third, we present an algorithm to approximate $Conn_2(G[A])$ and compare it with an optimal algorithm; Fourth, we propose an edge-constrained routing protocol (EC-SOLAR-KSP) for intermittently connected networks based on the contact probabilities computed in the first step and insights obtained from the second step. We then highlight the protocol's novelty and effectiveness by comparing it with a probabilistic routing protocol, and an epidemic routing protocol proposed in literature for intermittently connected networks.

9.2 Directions of Future Work

Although a significant amount of research work has been accomplished in the area of mobility profiling and routing within mobile wireless networks, we have identified additional topics based on our research that need further investigation. We list a summary of these topics in this section.

Efficient Profiling Techniques:

Our proposed clustering algorithm based on a Mixture of Bernoulli's distribution has few shortcomings already discussed in Chapter 5, Section 5.4.3. On one hand, it cannot differentiate between a "hub visit" and a "hub non-visit". On the other hand, it has no way of dynamically indicating profile changes in real time, which may be useful in anomaly detection in network security.

It may be possible to explore other time series analysis tools to either extend upon our clustering algorithm, or to devise a newer method to profile users. Whatever the method, the two main profiling objectives are: represent apparently chaotic mobility trace data by a probabilistic mixture of weighted hub lists; and given current profile information and a run time sequence of hub visits, flag a change in the profile dynamically.

Profile Exchange and Management:

Our method of mobility profile information dissemination was based on our novel concept of acquaintances. However, there may be other more efficient ways to perform the same task. The distribution and maintenance of profiles in our network obviously may benefit from a large recent literature on distributed hash tables (DHT) or content-addressable networks (CAN), such as in [38, 109, 110, 113, 122]. In our case, node IDs may be keys and their profiles may be the corresponding values. However, there are several fundamental differences between maintaining a DHT in an Internet-based P2P network and in a DTN network that needs to be carefully studied.

Another related topic is the caching of profile information and its lifetime. User portable devices are most often constrained in buffer space. One can try and optimize the replication of profile information and time them out from caches in an intelligent way such that profile queries may be responded to within a lower bound on either the number of hops the query has to go, or the number of total query packets that need to be sent out.

Routing Problem Formulation:

In our proposed hub-level routing strategy (SOLAR-HUB) we try to route packets from a user to a list of locations that may contain the destination with varying probabilities. This can lay the groundwork for a more general and interesting routing problem formulation where a user needs to send a packet to the hub containing the destination with maximum reliability and minimum cost. Thus, this is not simple unicast or anycast, because if the destination is not in the chosen hub, the packet needs to be rerouted to another probable hub. It is not simple multicast where you can forward to all hubs in destination's hub list, or a broadcast to all hubs as it shall not be as cost efficient. Thus, this scenario needs a newer type of forwarding mechanism, where the packet needs to travel to hubs in some manner which can maximize the delivery probability while minimizing the number of packet transmissions required.

Our analytical results show the problem of finding a directed delivery subgraph that maximizes the s,t-connectedness in it to be #P-hard, and we propose approximation algorithm to compute the delivery probability. Now, we also need more research into developing an optimal routing algorithm that can generate a delivery subgraph which will maximize the delivery probability.

Bibliography

[1] Wireless lan medium access control (mac) and physical layer (phy) specifications. *ISO/IEC 802.11:1999(E)* (1999).

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

[3] AIDA, M., ISHIBASHI, K., TAKANO, C., MIWA, H., MURANAKA, K., AND MIURA, A. Cluster structres in topology of large-scale social networks revealed by traffic data. *Proceedings of IEEE Globecomm '05, St. Louis* (November 2005).

[4] AKYILDIZ, I. F., HO, J. S. M., AND LIN, Y.-B. Movement-based location update and selective paging for pcs networks. *IEEE/ACM Transactions on Networking (TON) 4*, 4 (August 1996), 629–638.

[5] AN, B., AND PAPAVASSILIOU, S. A mobility-based clustering approach to support mobility management and multicast routing in mobile ad-hoc wireless networks. *International Journal of Network Management 11* (2001), 387–395.

[6] AN, B., AND PAPAVASSILIOU, S. Geomulticast: architectures and protocols for mobile ad hoc wireless networks. *Journal of Parallel and Distributed Computing 63* (2003), 182–195.

[7] BAI, F., SADAGOPAN, N., AND HELMY, A. Important: a framework to systematically analyze the impact of mobility on performance of routing protocols for adhoc networks. *Proceedings of IEEE INFOCOM '03, San Francisco, CA 2* (March 2003), 825–835.

[8] BALACHANDRAN, A., VOELKER, G., BAHL, P., AND RANGAN, P. Characterizing user behavior and network performance in a public wireless lan. *Proceedings of ACM Sigmetrics, '02, Marina del Rey, CA* (March 2002).

[9] BALAZINSKA, M., AND CASTRO, P. Characterizing mobility and network usage in a corporate wireless local-area network. *Proceedings of ACM MobiSys '03, San Francisco, CA* (May 2003), 303–316.

[10] BANERJEE, S., AND CHRYSANTHIS, P. Peer support in a mobile world. *Proceedings of NSF Workshop on Context-Aware Mobile Database Management, Providence, RI* (January 2002).

[11] BARRETT, C., MARATHE, M., SMITH, J., AND RAVI, S. A mobility and traffic generation framework for modeling and simulating ad hoc communication networks. *Proceedings of 2002 ACM Symposium on Applied Computing (SAC), Madrid, Spain* (2002), 122–126.

[12] BASAGNI, S., CHLAMTAC, I., AND SYROTIUK, V. Dynamic source routing for ad hoc networks using the global positioning system. *Proceedings of IEEE Wireless Communications and Networking Conference, New Orleans, LA 1* (1999), 301–305.

[13] BASAGNI, S., CHLAMTAC, I., AND SYROTIUK, V. Geographic messaging in wireless ad hoc networks. *Proceedings of VTC 1999, Amsterdam, Netherlands 3* (Spring 1999), 1957–1961.

[14] BASAGNI, S., CHLAMTAC, I., AND SYROTIUK, V. Location aware one-to-many communication in mobile multi-hop wireless networks. *Proceedings of VTC 2000, Boston, MA 1* (Spring 2000), 288–292.

[15] BASAGNI, S., TURGUT, D., AND DAS, S. Mobility-adaptive protocols for managing large ad hoc networks. *Proceedings of IEEE International Conference on Communications (ICC), Helsinki, Finland* (June 2001), 1539–1543.

[16] BEAUFOUR, A., LEOPOLD, M., AND BONNET, P. Smart-tag based data dissemination. *First ACM International Workshop on Wireless Sensor Networks and Applications (WSNA02), Atlanta, GA* (June 2002).

[17] BETTSTETTER, C. Smooth is better than sharp: a random mobility model for simulation of wireless networks. *Proceedings of the 4th ACM MSWiM, Rome, Italy* (July 2001), 19–27.

[18] BLAZEVIC, L., BOUDEC, J.-Y. L., AND GIORDANO, S. A Location Based Routing Method for Irregular Mobile Ad Hoc Networks. *EPFL-IC Report Number IC/2003/30* (May 2003), CH–1015.

[19] BLAZEVIC, L., GIORDANO, S., AND BOUDEC, J.-Y. L. Anchored Path Discovery in Terminode Routing. *Proceedings of The Second IFIP-TC6 Networking Conference (Networking 2002), Singapore* (May 2002).

[20] BOLCH, G., GREINER, S., DE MEER, H., AND TRIVEDI, K. S. *Queueing networks and Markov chains*. A Wiley-Interscience Publication. John Wiley & Sons Inc., New York, 1998. Modeling and performance evaluation with computer science applications.

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

[22] BOSE, P., MORIN, P., STOJMENOVIC, I., AND URRUTIA, J. Routing with guaranteed delivery in ad hoc wireless networks. *Wireless Networks 7*, 6 (2001), 609–616.

[23] BOUKERCHE, A. Simulation based study of on-demand routing protocols for ad hoc wireless networks. *Proceedings of 34th Annual Simulation Symposium, Seattle, WA* (April 2001), 85–92.

[24] BOUKERCHE, A., AND ROGERS, S. GPS query optimization in mobile and wireless ad hoc networking. *Proceedings of Sixth IEEE Symposium on Computer and Communications, Hammamet, Tunisia* (July 2001).

[25] BOUKERCHE, A., SHEETAL, V., AND CHOE, M. A route discovery optimization scheme using GPS system. *SCS/ACM/IEEE Annual Simulation Symposium, San Diego, CA* (April 2002).

[26] BROCH, J., JOHNSON, D. B., AND MALTZ, D. A. Dynamic Source Routing in ad hoc wireless networks. *Mobile Computing, Kluwer Academic Publishers* (1996), 153–181.

[27] BURGESS, J., GALLAGHER, B., JENSEN, D., AND LEVINE, B. N. Maxprop: Routing for vehicle-based disruption-tolerant networking. *Proceedings of IEEE Infocom '06, Barcelona, Spain* (April 2006).

[28] BURNS, B., BROCK, O., AND LEVINE, B. N. Mv routing and capacity building in disruption tolerant networks. *Proceedings of IEEE INFOCOM '05, Miami, FL 1* (March 2005), 398–408.

[29] BUTENKO, S., CHENG, X., DU, D.-Z., AND PARDALOS, P. On the construction of virtual backbone for ad hoc wireless networks. *Cooperative Control: Models, Applications and Algorithms* (2003), 43–54.

[30] CAMP, T., BOLENG, J., AND DAVIES, V. 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 2*, 5 (2002), 483–502.

[31] CAMP, T., BOLENG, J., WILLIAMS, B., WILCOX, L., AND NAVIDI, W. Performance comparison of two location based routing protocols for Ad Hoc networks. *Proceedings of IEEE INFOCOM '02, New York, NY* (2002).

[32] CHEN, G., HUANG, H., AND KIM, M. Mining frequent and periodic association patterns. *Dartmouth College, Coomputer Science and Engineering, Tech Report: TR 2005-550* (July 2005).

[33] CHENG, C. T., LEMBERG, H. L., PHILIP, S. J., VAN DEN BERG, E., AND ZHANG, T. SLALoM: A Scalable Location Management Scheme for Large Mobile Ad-hoc Networks. *Proceedings of Wireless Communications and Networking Conference, Orlando, FL* (March 2002).

[34] CHIANG, C.-C., GERLA, M., AND ZHANG, L. Shared Tree Wireless Network Multicast. *Proceedings of IEEE 6th International Conference on Computer Communications and Networks (ICCCN '97), Las Vegas, NV* (September 1997).

[35] CHINCHILLA, F., LINDSEY, M., AND PAPADOPOULI, M. Analysis of wireless information locality and association patterns in a campus. *Proceedings of IEEE INFOCOM '04, Hong Kong 2* (March 2004), 906–917.

[36] CHOFFNES, D. R., AND BUSTAMANTE, F. E. An integrated mobility and traffic model for vehicular wireless networks. *Proceedings of the 2nd ACM international workshop on Vehicular ad hoc networks, Cologne, Germany* (September 2005), 69–78.

[37] COLBOURN, C. J., Ed. *The Combinatorics of Network Reliability*. Oxford University Press, Oxford, NY, 1987.

[38] CRAINICEANU, A., LINGA, P., GEHRKE, J., AND SHANMUGASUNDARAM, J. Querying peer-to-peer networks using p-trees. In *WebDB '04: Proceedings of the 7th International Workshop on the Web and Databases* (New York, NY, USA, 2004), ACM Press, pp. 25–30.

[39] DAS, S. R., PERKINS, C. E., AND ROYER, E. M. Performance Comparison of Two On-demand Routing Protocols for Ad Hoc Networks. *Proceedings of IEEE INFOCOM '00, Tel-Aviv, Israel* (March 2000).

[40] DEMPSTER, A., LAIRD, N., AND RUBIN, D. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society, Series B 39*, 1 (1977), 1–38.

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

[42] ENGE, P., AND MISRA, P. Special Issue on Global Positioning System. *Proceedings of the IEEE 87*, 1 (January 1999), 3–15.

[43] GAREY, M., AND JOHNSON, D. Computers and intractability: A guide to the theory of np-completeness. *W.H. Freeman and Company* (June 1979).

[44] GHOSH, J., BEAL, M. J., NGO, H. Q., AND QIAO, C. On profiling mobility and predicting locations of wireless users. *Proceedings of ACM/SIGMOBILE REALMAN 2006 workshop at ACM Mobihoc '06, Florence, Italy* (May 2006).

[45] GHOSH, J., NGO, H. Q., AND QIAO, C. Mobility profile based routing within intermittently connected mobile ad hoc networks (icman). *Proceedings of IW-*

CMC 2006 Delay Tolerant Mobile Networks workshop, Vancouver, Canada (July 2006).

[46] GHOSH, J., PHILIP, S., AND QIAO, C. Acquaintance Based Soft Location Management (ABSoLoM) in MANET. *Proceedings of IEEE Wireless Communications and Networking Conference (WCNC) '04, Atlanta, GA* (March 2004).

[47] GHOSH, J., PHILIP, S., AND QIAO, C. Orbit mobility framework and orbit based routing (obr) protocol for manet. *University at Buffalo, Technical Report, CSE Dept. TR-2004-08* (July 2004).

[48] GHOSH, J., PHILIP, S., AND QIAO, C. Performance analysis of mobility based routing protocols in manet. *University at Buffalo, Technical Report, CSE Dept. TR-2004-14* (September 2004).

[49] GHOSH, J., PHILIP, S. J., AND QIAO, C. Sociological orbit aware location approximation and routing (solar) in manet. *Accepted for publication in ELSEVIER Ad Hoc Networks Journal, Nov 2005.*

[50] GHOSH, J., PHILIP, S. J., AND QIAO, C. Poster abstract: Sociological orbit aware location approximation and routing (solar) in manet. *Poster in ACM Mobihoc '05, Urbana-Champaign, IL* (May 2005).

[51] GHOSH, J., PHILIP, S. J., AND QIAO, C. Sociological orbit aware location approximation and routing in manet. *Proceedings of IEEE Broadnets '05, Boston, MA* (October 2005), 688–697.

[52] GHOSH, J., WESTPHAL, C., NGO, H. Q., AND QIAO, C. Bridging intermittently connected mobile ad hoc networks (icman) with sociological orbits. *Poster in IEEE Infocom '06, Barcelona, Spain* (April 2006).

[53] GLANCE, N., SNOWDON, D., AND MEUNIER, J.-L. Pollen: using people as a communication medium. *First ACM International Workshop on Wireless Sensor Networks and Applications (WSNA02), Atlanta, GA 35*, 4 (March 2002), 429–442.

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

[55] GUARE, J. Six Degrees of Separation. *A Play* ((Vintage Books, New York) 1990).

[56] HAARTSEN, J. The bluetooth radio system. *IEEE Personal Communications Magazine* (February 2000), 28–36.

[57] HALL, J., BARBEAU, M., AND KRANAKIS, E. Using mobility profiles for anomaly-based intrusion detection in mobile networks. *Pre-conference wireless and mobile security workshop, ISOC NDSS '05, San Diego, CA* (2005).

[58] HARRAS, K. A., ALMEROTH, K. C., AND BELDING-ROYER, E. M. Delay tolerant mobile networks (dtmns): Controlled flooding in sparse mobile networks. *Proceedings of Networking '05, Waterloo, Ontario, Canada* (May 2005).

[59] HARRISON, P. G., AND KNOTTENBELT, W. J. Passage time distributions in large markov chains. *SIGMETRICS Perform. Eval. Rev. 30*, 1 (2002), 77–85.

[60] HENDERSON, T., KOTZ, D., AND ABYZOV, I. The changing usage of a mature campus-wide wireless network. *Proceedings of ACM/IEEE MobiCom '04, Philadelphia, PA* (September 2004), 187–201.

[61] HONG, X., GERLA, M., PEI, G., AND CHIANG, C.-C. A Group Mobility Model for Ad hoc Wireless Networks. *Proceedings of ACM/IEEE MSWIM '99, Seattle, WA* (August 1999).

[62] HONG, X., KWON, T., GERLA, M., GU, D., AND PEI, G. A mobility framework for ad hoc wireless networks. *In ACM 2nd International Conference on Mobile Data Management (MDM), Hong Kong* (January 2001).

[63] HORTON, M., CULLER, D., PISTER, K., HILL, J., SZEWCZYK, R., AND WOO, A. Mica: The commericialization of microsensor motes. *In Sensors Magazine* (April 2002), 40–48.

[64] HUGHES, B. D. *Random walks and random environments*. Oxford University Press, 1996.

[65] HUI, P., CHAINTREAU, A., SCOTT, J., GASS, R., CROWCROFT, J., AND DIOT, C. Pocket switched networks and the consequences of human mobility in conference environments. *Presented at the Workshop on Delay Tolerant Networking during SIGCOMM 2005, Philadelphia, PA*.

[66] INC, . The Math Works. *Matlab Reference Guide* (1992).

[67] JAIN, S., FALL, K., AND PATRA, R. Routing in a delay tolerant network. *Proceedings of ACM SIGCOMM'04, Portland, OR* (September 2004).

[68] JANSSEN, J., AND MANCA, R., Eds. *Applied Semi-Markov Processes*. Springer, 1st Edition, 2005.

[69] JARDOSH, A., BELDING-ROYER, E. M., ALMEROTH, K. C., AND SURI, S. Towards Realistic Mobility Models for Mobile Ad hoc Networks. *Proceedings of ACM/IEEE MobiCom '03, San Diego, CA* (September 2003).

[70] JERRUM, M. *Counting, sampling and integrating: algorithms and complexity*. Lectures in Mathematics ETH Zürich. Birkhäuser Verlag, Basel, 2003.

[71] JIANG, M., LI, J., AND TAY, Y. Cluster Based Routing Protocol (CBRP) Functional Specification.

[72] JOHANSSON, P., LARSSON, T., HEDMAN, N., MIELCZAREK, B., AND DEGERMARK, M. Scenario-based performance analysis of routing protocols for mobile ad-hoc networks. *Proceedings of ACM/IEEE Mobicom, '99, Seattle, WA* (1999), 195–206.

[73] JOHANSSON, P., LARSSON, T., HEDMAN, N., MIELCZAREK, B., AND DEGERMARK, M. Scenario-based performance analysis of routing protocols for mobile ad-hoc networks. *Proceedings of ACM/IEEE MOBICOM '99, Seattle, WA* (1999), 195–206.

[74] JUANG, P., OKI, H., WANG, Y., MARTONOSI, M., PEH, L. S., AND RUBENSTEIN, D. Energy-efficient computing for wildlife tracking: design tradeoffs and early experiences with zebranet. *Proceedings of the 10th International Conference on Architectural support for programming languages and operating systems (ASPLOS), San Jose, CA* (October 2002), 96–107.

[75] KARP, B., AND KUNG, H. T. GPSR : Greedy perimeter stateless routing for wireless networks. *Proceedings of ACM/IEEE MobiCom '00, Boston, MA* (August 2000), 243–254.

[76] KIM, M., AND KOTZ, D. Classifying the mobility of users and the popularity of access points. *Proceedings of the International Workshop on Location- and Context-Awareness (LoCA), Germany* (May 2005).

[77] KIM, M., KOTZ, D., AND KIM, S. Extracting a mobility model from real user traces. *Proceedings of IEEE INFOCOM'06, Barcelona, Spain* (April 2006).

[78] KNOBLAUCH, R. L., PIETRUCHA, M. T., AND NITZBURG, M. Field Studies of Pedestrian Walking Speed and Start-Up Time. *Transportation Research Board Records*, 1538 (1996).

[79] KO, Y. B., AND VAIDYA, N. H. Location-Aided Routing (LAR) in mobile Ad-Hoc networks. *Proceedings of IEEE INFOCOM '98, San Francisco, CA* (October 1998), 66–75.

[80] KRIVELEVICH, M., AND VU, V. H. Choosability in random hypergraphs. *J. Combin. Theory Ser. B 83*, 2 (2001), 241–257.

[81] LADD, A., BEKRIS, K., RUDYS, A., WALLACH, D., AND KAVRAKI, L. On the feasibility of using wireless ethernet for indoor localization. *IEEE Transactions on Robotics and Automation 20*, 3 (June 2004), 555–559.

[82] LAI, K., ROUSSOPOULOS, M., TANG, D., ZHAO, X., AND BAKER, M. Experiences with a mobile testbed. *Proceedings of The Second International Conference on Worldwide Computing and its Applications (WWCA98), Tsukuba, Japan* (March 1998).

[83] LAM, D., COX, D., AND WIDOM, J. Teletraffic Modeling for Personal Communications Services. *IEEE Communications Magazine* (February 1997), 79–87.

[84] LEBRUN, J., CHUAH, C. N., GHOSAL, D., AND ZHANG, H. M. Knowledge-based opportunistic forwarding in vehicular wireless ad hoc networks. *Proceedings of IEEE Vehicular Technology Conference, Dallas, TX* (Spring 2005).

[85] LEE, J.-K., AND HOU, J. C. Modeling steady-state and transient behaviors of user mobility:: formulation, analysis, and application. *Proceedings of AMC Mobihoc '06, Florence, Italy* (May 2006), 85–96.

[86] LEE, K., AND KIM, S. Modeling variable user mobility with stochastic correlation concept. *Computer Networks: The International Journal of Computer and Telecommunications Networking 38*, 5 (April 2002), 603–612.

[87] LEE, S. J., TOH, C. K., AND GERLA, M. Performance evaluation of table-driven and on-demand ad hoc routing protocols. *Proceedings of IEEE PIMRC '99, Osaka, Japan* (Sept 1999), 297–301.

[88] LEGUAY, J., FRIEDMAN, T., AND CONAN, V. Dtn routing in a mobility pattern space. *Proceeding of the 2005 ACM SIGCOMM workshop on Delay-tolerant networking, Philadelphia, PA* (August 2005), 276–283.

[89] LI, J., JANOTTI, J., COUTO, D. S. J. D., KARGER, D. R., AND MORRIS, R. A Scalable Location Service for Geographic Ad Hoc Routing. *Proceedings of ACM/IEEE MobiCom '00, Boston, MA* (August 2000), 120–130.

[90] LI, J., AND MOHAPATRA, P. LAKER: Location Aided Knowledge Extraction Routing for Mobile Ad Hoc Networks. *Proceedings of Wireless Communications and Networking Conference, New Orleans, LA 2* (March 2003), 1180–1184.

[91] LIAO, Y., TAN, K., ZHANG, Z., AND GAO, L. Estimation based erasure-coding routing in delay tolerant networks. In *Technical Report, MSR-TR-2006-05* (New York, NY, USA, 2005), Microsoft.

[92] LIN, G., NOUBIR, G., AND RAJARAMAN, R. Mobility Models for Ad hoc Network Simulation. *Proceedings of IEEE INFOCOM '04, Hong Kong* (March 2004).

[93] LINDGREN, A., DORIA, A., AND SCHELEN, O. Poster: Probabilistic routing in intermittently connected networks. *Proceedings of The Fourth ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc 2003), Annapolis, MD* (June 2003).

[94] LIU, T., BAHL, P., AND CHLAMTAC, I. Mobility modeling, location tracking, and trajectory prediction in wireless atm networks. *IEEE Journal on Selected Areas in Communications 16*, 6 (August 1998), 922–936.

[95] LUNDGREN, H., NORDSTROM, E., AND TSCHUDIN, C. The gray zone problem in IEEE 802.11b based ad hoc networks. *ACM SIGMOBILE Mobile Computing and Communications Review (MC2R) 6*, 3 (July 2002), 104–105.

[96] MARKOULIDAKIS, J., LYBEROPOULOS, G., TSIRKAS, D., AND SYKAS, E. Mobility Modeling in Third-Generation Mobile Telecommunications Systems. *IEEE Personal Communications 4*, 4 (August 1997), 41–56.

[97] MCCLUSKEY, E. J. Minimization of boolean functions. *Bell Systems Technical Journal 35*, 5 (1956), 1417–1444.

[98] MCNETT, M., AND VOELKER, G. M. Access and mobility of wireless pda users. *In Mobile Computing Communications Review 9*, 2 (April 2005), 40–55.

[99] MILGRAM, S. The small world problem. *Psychology Today 1* (1967), 61.

[100] MUSOLESI, M., HAILES, S., AND MASCOLO, C. An ad hoc mobility model founded on social network theory. *Proceedings of ACM/IEEE MSWIM '04, Venezia, Italy* (October 2004).

[101] MUSOLESI, M., HAILES, S., AND MASCOLO, C. Adaptive routing for intermittently connected mobile ad hoc networks. *Proceedings of IEEE 6th International*

Symposium on a World of Wireless, Mobile and Multimedia Networks (WOW-MOM'05), Taormina, Italy (June 2005).

[102] NI, S.-Y., TSENG, Y.-C., CHEN, Y.-S., AND SHEU, J.-P. The broadcast storm problem in a mobile ad hoc network. *Proceedings of ACM/IEEE MobiCom '99, Seattle, WA* (August 1999), 151–162.

[103] PEARLMAN, M., AND HAAS, Z. Determining the Optimal Configuration for the Zone Routing Protocol. *IEEE Journal on Selected Areas in Communications 17*, 8 (August 1999), 1395–1414.

[104] PERKINS, C. E., AND ROYER, E. M. Ad hoc On-Demand Distance Vector Routing. *Proceedings of the 2nd IEEE Workshop on Mobile Computing Systems and Applications, New Orleans, LA* (February 1999), 90–100.

[105] PHILIP, S., AND QIAO, C. Hierarchical grid location management for large wireless ad hoc networks. *ACM SIGMOBILE Mobile Computing and Communications Review (MC2R)* (To appear).

[106] PHILIP, S. J., GHOSH, J., AND QIAO, C. Performance Evaluation of Multilevel Hierarchical Location Management for Ad Hoc Networks. *ELSEVIER Computer Communications Journal, Special Issue on Performance Issues of Wireless LANs, PANs, and Ad Hoc Networks 28*, 10 (June 2005), 1110–1122.

[107] QUEEN, R. M., LIN, H.-Y., AND GROSS, M. T. 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*.

[108] QUINE, W. V. A way to simplify truth functions. *American Mathematical Monthly 62*, 9 (1955), 627–631.

[109] RAMABHADRAN, S., RATNASAMY, S., HELLERSTEIN, J. M., AND SHENKER, S. Brief announcement: prefix hash tree. In *PODC '04: Proceedings of the twenty-third annual ACM symposium on Principles of distributed computing* (New York, NY, USA, 2004), ACM Press, pp. 368–368.

[110] RATNASAMY, S., FRANCIS, P., HANDLEY, M., KARP, R., AND SCHENKER, S. A scalable content-addressable network. In *SIGCOMM '01: Proceedings of the 2001 conference on Applications, technologies, architectures, and protocols for computer communications* (New York, NY, USA, 2001), ACM Press, pp. 161–172.

[111] RAVIKIRAN, G., AND SINGH, S. Influence of mobility models on the performance of routing protocols in ad-hoc wireless networks. *Proceedings of IEEE Vehicular Technology Conference '04, Los Angeles, CA* (May 2004).

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

[113] RHEA, S., GODFREY, B., KARP, B., KUBIATOWICZ, J., RATNASAMY, S., SHENKER, S., STOICA, I., AND YU, H. Opendht: a public dht service and its uses. In *SIGCOMM '05: Proceedings of the 2005 conference on Applications, technologies, architectures, and protocols for computer communications* (New York, NY, USA, 2005), ACM Press, pp. 73–84.

[114] RIZZO, L. Effective erasure codes for reliable computer communication protocols. *ACM Computer Communication Review 27*, 2 (April 1997), 24–36.

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

[116] SCHWAB, D., AND BUNT, R. Characterizing the use of a campus wireless network. *Proceedings of IEEE INFOCOM '04, Hong Kong 2* (March 2004), 862–870.

[117] SCHWARZ, G. Estimating the dimension of a model. *The Annals of Statistics 6* (1978), 461–464.

[118] SHAH, R., ROY, S., JAIN, S., AND BRUNETTE, W. Data mules: Modeling a three-tier architecture for sparse sensor networks. *IEEE SNPA Workshop, Seattle, WA* (May 2003).

[119] SHAH, S., HERNANDEZ, E., AND HELAL, A. 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, San Diego, CA 11*, 1 (February 2001), 3–30.

[120] SONG, L., DESHPANDE, U., KOZAT, U. C., KOTZ, D., AND JAIN, R. Predictability of wlan mobility and its effects on bandwidth provisioning. *Proceedings of IEEE Infocom '06, Barcelona, Spain* (April 2006).

[121] SPYROPOULOS, T., PSOUNIS, K., AND RAGHAVENDRA, C. S. Spray and wait: an efficient routing scheme for intermittently connected mobile networks. *Proceeding of the 2005 ACM SIGCOMM workshop on Delay-tolerant networking, Philadelphia, PA* (August 2005), 252–259.

[122] STOICA, I., MORRIS, R., KARGER, D., KAASHOEK, M. F., AND BALAKRISHNAN, H. Chord: A scalable peer-to-peer lookup service for internet applications. In *SIGCOMM '01: Proceedings of the 2001 conference on Applications, technologies, architectures, and protocols for computer communications* (New York, NY, USA, 2001), ACM Press, pp. 149–160.

[123] SU, W., LEE, S.-J., AND GERLA, M. Mobility prediction and routing in ad hoc wireless networks. *International Journal of Network Management 11*, 1 (February 2001), 3–30.

[124] TAN, D., ZHOU, S., HO, J., MEHTA, J., AND TANABE, H. 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).

[125] TANG, D., AND BAKER, M. Analysis of a metropolitan-area wireless network. *Proceedings of ACM/IEEE MobiCom '99, Seattle, WA* (August 1999), 13–23.

[126] TANG, D., AND BAKER, M. Analysis of a local-area wireless network. *Proceedings of ACM/IEEE MobiCom '00, Boston, MA* (August 2000), 1–10.

[127] TUDUCE, C., AND GROSS, T. A mobility model based on wlan traces and its validation. *Proceedings of IEEE INFOCOM '05, Miami, FL 1* (March 2005), 664–674.

[128] VAHDAT, A., AND BECKER, D. Epidemic routing for partially connected ad hoc networks. *Technical Report CS-200006, Duke University* (April 2000).

[129] VALIANT, L. The complexity of enumeration and reliability problems. *SIAM J. Computing 8* (1979), 410–421.

[130] WANG, W., AND AKYILDIZ, I. F. On the estimation of user mobility pattern for location tracking in wireless networks. *Proceedings of IEEE Globecom '02, Taipei, Taiwan* (November 2002), 619–623.

[131] WANG, Y., JAIN, S., MARTONOSI, M., AND FALL, K. Erasure-coding based routing for opportunistic networks. In *WDTN '05: Proceeding of the 2005*

ACM SIGCOMM workshop on Delay-tolerant networking (New York, NY, USA, 2005), ACM Press, pp. 229–236.

[132] WANG, Y., AND WU, H. Dft-msn: The delay/fault-tolerant mobile sensor network for pervasive information gathering. *Proceedings of IEEE INFOCOM'06, Barcelona, Spain* (April 2006).

[133] WESTPHAL, C. On maximizing the lifetime of distributed information in ad hoc networks with individual constraints. *Proc. of ACM MOBIHOC '05, Urbana-Champaign, IL* (May 2005), 26–33.

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

[135] WOO, S.-C., AND SINGH, S. Scalable Routing Protocol for Ad Hoc Networks. *Wireless Networks 7* (January 2001), 513–529.

[136] YOON, J., LIU, M., AND NOBLE, B. Random Waypoint Considered Harmful. *Proceedings of IEEE INFOCOM '03, San Francisco, CA 2* (March 2003), 1312–1321.

[137] ZENG, X., BAGRODIA, R., AND GERLA, M. Glomosim: a library for parallel simulation of large-scale wireless networks. *Proceedings of the 12th Workshop on Parallel and Distributed Simulations (PADS) '98, Alberta, Canada* (May 1998), 154–161.

[138] ZHAO, W., AMMAR, M., AND ZEGURA, E. Controlling the mobility of multiple data transport ferries in a delay-tolerant network. *Proceedings of IEEE INFOCOM '05, Miami, FL* (March 2005).