BEHAVIORAL ANALYSIS, USER MODELING, AND PROTOCOL DESIGN BASED ON LARGE-SCALE WIRELESS NETWORK TRACES

By

WEI-JEN HSU

A DISSERTATION PRESENTED TO THE GRADUATE SCHOOL
OF THE UNIVERSITY OF FLORIDA IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

UNIVERSITY OF FLORIDA

2008

To my families

ACKNOWLEDGMENTS

I would like to thank first and foremost my adviser, Dr. Ahmed Helmy, for his guidance and enlightenment through the course of my work towards the Ph.D. degree. Dr. Helmy is not only a great academic adviser, but also a great mentor, who helped me significantly in keeping the faith and pursuing my dream, and a great role model from whom I learned so many different things in life. Looking back at the years I worked with him, they were not only intellectually satisfying, but also a tremendously joyful journey in my life.

I would like also express deep gratitude to my supervisory committee members, including Dr. Sartaj Sahni, Dr. Alin Dobra, Dr. Dapeng Wu, and Dr. Ye Xia, who have helped me significantly in the process of forming my dissertation. Their inputs bring new perspectives and understanding to the problem and make the dissertation more complete. In addition, I would like to explicitly thank them for their inputs during the exams, which have collectively made the whole process intellectually challenging and rewarding.

I would like to also extend my thanks to many professors and colleagues I had the privilege to work with. They have helped me in so many different ways. For Dr. Debojyoti Dutta, I appreciate his patience to spend time with me forming the research problems and identifying the proper tools, improving my writing, and in general many helpful discussions. For Professor Konstantinos Psounis and Dr. Thrasyvoulos Spyropoulos, I thank them for their help during the time-variant community mobility model project, which took me to the appreciation of rigorous theoretical work. For Dr. Jabed Faruque, I would like to thank first his many supportive challenges to my research problems, which make me think more carefully, and second I am also in debt to him for the time he spared from research to maintain a nice computing environment for the group, without which many of the results in this dissertation would be at least delayed. For Dr. Fan Bai, Shao-Cheng Wang, and Dr. Sapon Tanachaiwiwat, I would like to thank them for many helpful suggestions and review comments on my research work.

I would also like to thank Chih-Ping Li, Professor Bhaskar Krishnamachari, Haw-Wei Shu, Kashyap Merchant, and Chih-Hsin Hsu, for the very best experience to work together on various projects. I learned significantly from each of you. Finally, I want to thank all the fellow A-groupers, Udayan Kumar and Sungwook Moon, who helped me in preparing visually appealing demo of my work, and Dr. Karim Seada, Jeeyoung Kim, Ganesha Bhaskara, Yibin Wang, Shirin Ebrahimi, Dr. Yongjin Kim, Shamin Begum, Vishal Shankhla, for many helpful conversations and a very stimulating and enjoyable working environment we collectively created.

Finally, I would like to express my deepest gratitude towards my families, who have always been my best support and encouragement. This dissertation is made possible by the many years they prepared me for the great challenges in the world, and the endless support even if I could not be with them most of the times during the Ph.D. study. Without you, I would not achieve what I have today.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Abstract of Dissertation Presented to the Graduate School
of the University of Florida in Partial Fulfillment of the
Requirements for the Degree of Doctor of Philosophy

BEHAVIORAL ANALYSIS, USER MODELING, AND PROTOCOL DESIGN BASED ON LARGE-SCALE WIRELESS NETWORK TRACES

By

Wei-Jen Hsu

August 2008

Chair: Ahmed Helmy
Major: Computer Engineering

In this dissertation we describe the *TRACE* framework, which is a five-step procedure for the environment-aware approach towards wireless mobile computer networks. As mobile computer networks become ubiquitous and deeply integrated with the daily lives, it is crucial to understand the network and design its protocols and services with the environment-aware approach: We first collect extensive network *T*races that reflect truthfully the detailed behaviors of its users, and *R*epresent the rich data sets in concise representations. Then we *A*nalyze these constructed representations to *C*haracterize the users. While many observed characteristics are interesting in themselves and reveal important differences between the realistic environment and commonly made assumptions in the literature, we further add values to these findings by *E*mploying them in various important tasks, including modeling the network users and designing routing protocols.

The dissertation is centered around three major case studies, ranging from the microscopic, individual user behavior in the wireless networks to the macroscopic, global user encounter patterns. Specifically, in the case studies, we (1) observe the *individual user mobility* from the collected traces, identify *skewed preferences* and *periodical re-appearance at the same location* as prominent mobility characteristics, and propose the *time-variant community (TVC) mobility model* to capture such behaviors. The TVC model is *flexible* to match with many empirical traces while being *mathematically tractable*. (2) We construct an efficient way for mobile users to summarize their mobility preferences based on singular

value decomposition (SVD) and calculate the distance metric between users. Based on this distance metric, we identify *user groups in the population* based on their mutual similarities, and design a profile-cast service to deliver messages to these behavioral groups without knowing their identities. (3) We further analyze the *global encounter patterns* between nodes, observe a fast-emerging Small World encounter pattern, and leverage such a network property to design an efficient message dissemination protocol named *CSI*, in which *C*ommunication relies on the *S*table yet *I*mplicit structures in mobile networks.

CHAPTER 1
INTRODUCTION

1.1 Emergence of Mobile Networks

In the past decades, wireless access technologies at the last-hop of communication networks (e.g., cellular phone systems, wireless local area networks (WLANs)) took off and its wide-spread deployments brought great convenience to the end users. Encouraged by such an untethered environment, in recent years, we have witnessed the emergence of an array of portable computing and communication devices (e.g., laptops, PDAs, smart phones). Advances in wireless communication technologies and standards have made ubiquitous communication an emerging reality. With the ever expanding adoption of these wireless-capable devices, there is an increasing interest in new communication paradigms and applications that do not necessarily rely on the infrastructure.

Such a keen interest in infrastructure-independent communication has led to the establishment of the research area generally known as the *Mobile ad hoc networks* (MANETs) [3]. By definition, MANETs are self-organized, infrastructure-less networks, and considered as stand-alone networks in which the participants exchange information among themselves. Typically, MANETs consist of autonomous devices, and each device plays both roles of an end-host and a router at the same time. While the communication range of individual nodes is limited to its close vicinity due to the nature of the wireless medium, the end-to-end connectivity in the network is provided by the cooperation of its participants, through multi-hop forwarding, sometimes involving temporary storage of the messages in the non-volatile memory of intermediate nodes (in a sub-case of MANETs generally known as the Delay Tolerant Networks, or DTNs[4]). MANETs provide an attractive alternative way of communication where the setup of an infrastructure is infeasible or too costly, or when the disseminated information is meant for only local participants so there is no need to reach the Internet. Potential applications of MANETs include vehicular networks (VANET)[7, 30], disaster relief[5], wild-life tracking[32, 33],

and providing network connectivity to the rural area[6], to name a few. The emergence of personalized portable wireless communication/computing devices (e.g., PDAs, smart phones) also opens the door for creating a mobile virtual social network between people. We envision such a MANET would facilitate socializing applications (e.g., matching people with similar interests, information sharing among small groups, etc.) in the future.

1.2 Behavior-aware Network Approach

Traditionally, network research is done without specific assumptions or detailed understanding of the environments in which the proposed protocols or services are used. This approach leads to *generic, behavior-oblivious* protocols. The goal is to construct generic, robust protocols that work regardless of the actual environment. While this approach favors protocols that rely on a minimal set of basic primitives supported by the underlying environments and contributes greatly to the wide-range acceptance of the basic Internet protocols (e.g., IP, TCP, HTTP, etc.), the resulting protocols may not be the most efficient in a particular environment. Furthermore, without specific considerations at the design phase, it is usually not easy to fine-tune the resulting protocols and adapt them to various environments efficiently.

Contrary to the fore-mentioned design philosophy, we argue that one major requirement for future network protocols and services is its adaptability to specific environments or user behaviors. With the proliferation of the hand-held devices and the enhancement in their capabilities, we envision that future usage of mobile devices will be highly personalized. The network-capable terminals will have tight one-to-one correspondence to its owners. Users will incorporate these new technologies into their daily lives, and the way they use new devices and services will reflect their personality and lifestyle. This opportunity opens up the door for novel paradigms such as *behavior-aware* protocols and services. Such services look into user behavior and leverage the underlying patterns in user activity to adapt their operations and have the potential to work more efficiently and suit potentially very different needs among the users. One classical example

is the data mining efforts from various online stores (e.g., Amazon.com) that provide personalized shopping offers based on browsing history. However, little attention has been directed towards leveraging user behavioral patterns for services or protocol design in the mobile computing/communication paradigm.

Therefore we propose in this dissertation the *behavior-aware* approach to computer network research. This approach starts from detailed analysis of the underlying environments in which one expects the network will be deployed, and then *explicitly* incorporates the findings into the design phase of the network protocols and services. We propose this approach based on the following reasons:

1. By analyzing multiple data sets collected in realistic environment, the understandings we gain shed lights on fundamental user behaviors, such as preferences and periodicity. The commonalities and differences found from different environments also help us to distinguish common user behaviors from the specifics of a given environment.

2. The findings from the analysis provide a set of more suitable assumptions to be used later in the evaluation of the proposed network protocols and services. In particular, it helps to avoid making unrealistic assumptions in the evaluation stage, so that the results can be more meaningful.

3. The insight gained from the analysis of the environment usually provides a good basis to build *behavior-aware* protocols and services. For example, a good understanding of network usage pattern may lead to a good trend prediction and abnormality detection service.

4. With a thorough analysis of user behavior from *multiple* environments, one can identify the important commonalities and build the protocols with these facts as major considerations.

In the following section we introduce the generic framework, abbreviated as *TRACE*, for the behavior-aware approach towards computer networks in this dissertation. The dissertation features several case studies of the *TRACE* framework, with different focuses on various sub-problems of computer network design. They will be introduced in the subsequent section.

1.3 The *TRACE* Framework

There are five major steps in our behavior-aware approach to computer network research. The shortened name of the framework is the *TRACE* framework, as illustrated in Fig. 1-1. The individual steps are introduced below.

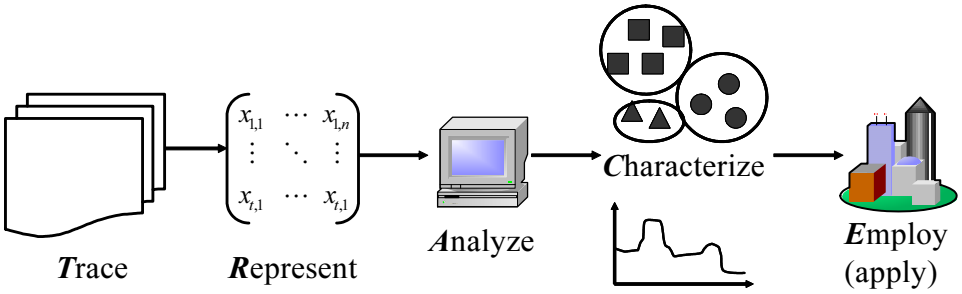


Figure 1-1. Illustration of the *TRACE* framework.

1. *T*race: The research starts with extensive collection of data sets about the underlying environment. The methodology for trace collection should be set up to maximize the relevant information captured in the traces. In this dissertation, we use both the data sets we collected ourselves and the publicly available data sets through archives in the research community (e.g., [1, 2]). Part of the trace collection effort is still ongoing at the University of Florida. We will explain the details about the traces in chapter 3.

2. *R*epresent: After the trace collection phase, post-processing should be applied to the large-scale traces so that the raw data is transformed into a proper representation to facilitate the later analysis step. Usually, the raw trace is presented as a sequence of events occurred in the given environment. The representation of the trace, in our context, is a quantitative measure of these events, presented as a scaler quantity, a vector, a matrix, or a graph. Various representations we choose are based on the specific points we wish to understand about the environment. When the *TRACE* framework is applied to compare multiple traces, this step also involves the normalization of data sets collected with different techniques, so they become comparable.

3. *A*nalyze: The analysis step involves the application of various mathematical tools or algorithms to obtain distilled information from the representations of the data sets. Examples of such are the distributions for scaler quantities, the singular vectors of the matrices, or the major component of a graph. We try various techniques in the dissertation, including basic statistics and probability analysis (leading to distributions of scaler quantities), singular value decomposition (SVD) that reveals the major trends in the matrices distilled from the data sets, unsupervised learning

(hierarchical clustering), and the application of Small World theory, to enrich the repository of our analysis tools.

4. *C*haracterize: We refer to the findings after the analysis and the interpretation of the results as the *characteristics* of the traces. These are the important lessons we learn from looking at the data sets. Consistent characteristics from multiple data sets reveal the major underlying trends in the environments, and these are the points we should pay attention to for network analysis and design.

5. *E*mploy: The characteristics we discover could be employed in various tasks, such as (1) building models for the underlying environment, (2) profiling and classifying individual users, (3) suggesting heuristics of designing protocols for message forwarding. In mobility modeling, the findings of spatial and temporal correlation in mobility patterns lead to the proposal of a new mobility model, the *time-variant community model*, that captures varying mobility characteristics depending on space and time. In the profiling effort, we discover that the location visiting preferences can be utilized to divide the population into distinct groups. We design salient metrics for the *distances* between users in terms of the similarity in location visiting preferences, and leverage unsupervised learning techniques to identify the important groups. Finally, we leverage the similarity metrics to design a *behavior-aware* message dissemination protocol. The protocol helps to reduce the transmission overhead under similar success rate when compared with *behavior-oblivious* protocols (e.g., flooding or random walk), when the goal is to transmit copies of messages to a group defined by its behavior. We will further explain these applications in later chapters of the dissertation.

It is perhaps interesting to note that the *TRACE* framework can be either *T*race driven or *E*mployment driven. In the *T*race driven scenario, one starts with rich data sets and seek to understand the data set. This approach sometimes leads to interesting, unexpected, or puzzling findings, which warrant further investigation and at times lead to important usage. In the *E*mployment driven scenario, one starts with a particular system design goal in mind, and then constructs trace-collection facilities in relevant environments, and consult the collected data sets for guidelines to achieve the goal. We take mainly the first approach in this dissertation.

1.4 Study Components

In this dissertation we use three case studies to display the usefulness of the *TRACE* framework. These study components are illustrated in Fig. 1-2. As shown in the figure, each case study has two separate parts, the *observation* and the *application*. The flow

of research in the case studies fall into the *trace driven* scenario mentioned above: we start from the extensive data sets, look at a specific aspect of the trace using a proper representation, and highlight the important characteristics through detailed analyses. Following the *observations*, for each study case we also devise an *application* to support the importance and usefulness of the *observations*. The three study cases focus on different aspects of the information obtained from the traces, from *microscopic* individual user behavior, to *macroscopic* structure of the whole user population. The study we present here is structured by the three case studies, as we introduce in details below.

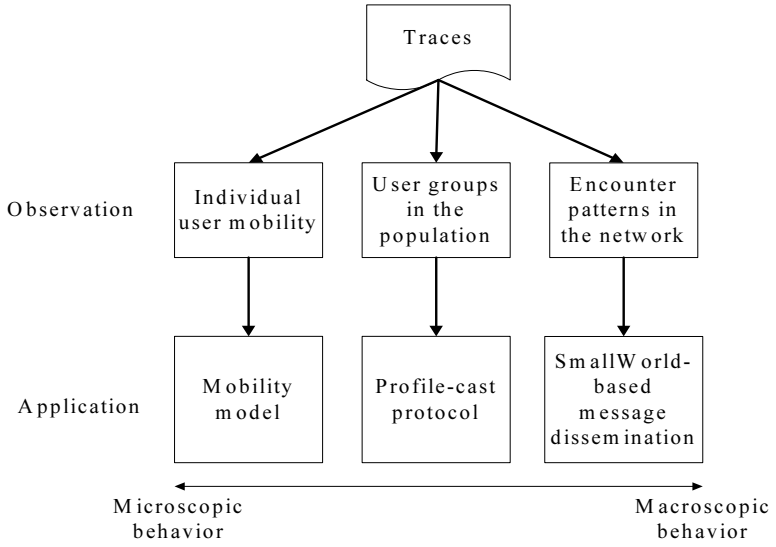


Figure 1-2. Components of the study.

We first discuss the related work in chapter 2 to position our work in the literature. In chapter 3, we introduce the data sets we use through the dissertation, address the strengths and the shortcomings of the currently available data sets in the community, and discuss about the ongoing data collection effort at University of Florida. We then move on to show our three case studies in chapter 4, 5, and 6/7, respectively. Finally, we conclude the dissertation in chapter 8.

Chapter 4 focuses on the observed individual user mobility (i.e., the microscopic user behavior) from the traces. We start by displaying several interesting common observations from the traces, including the *on-off user behavior*, the *skewed location*

visiting preferences, and the *periodic re-appearance of users at the same locations*. We further show, although these user behaviors are prevalent in realistic scenarios, *none* of the existing mobility models capture all of them successfully. Inspired by this, we propose a *time-variant community mobility model* as a *generic* and *realistic* mobility model to capture these behaviors. In addition, the *time-variant community mobility model* is *amenable to mathematical analysis*. This mobility model is a powerful tool to enable realistic performance analysis for various protocols and services in MANETs.

We move up one level in terms of the scope of focus in chapter 5. The central question we address in this chapter is whether it is possible to identify *groups* of users following similar trends from the collected traces, without any assumptions of the existence of groups in the environment. We apply *unsupervised learning* techniques (e.g., clustering) in this chapter to the data sets, and display that within the large population (in the order of thousands), with careful selection of the features and the distance metric between users we can classify users based on the *preferences* in their mobility patterns. Such a group identification technique has a wide-range of applications, from network management to intelligent advertising to *behavior-aware protocol design*. We choose *behavior-aware protocol design* as the application for this case study and propose a *profile-cast* service which targets a group of users defined implicitly by their behavior patterns as the destination nodes. The salient feature of the *profile-cast* service is that the sender does not have to know the receiver's network identities when sending the messages.

We further move up one level to understand the macroscopic *structure* of user interaction in chapter 6. In this chapter, we seek to understand the *encounter patterns* (i.e., the pattern of mobile nodes moving into the communication range of each other) realistically by representing the information obtained from the traces as graphs. We observed the emergence of a special graphic structure, known as the SmallWorld [8], from multiple traces. This suggests a potential correspondence between the existing social network structure and the communication opportunities between the mobile users in the

future MANETs. Inspired by this finding, we devise message dissemination protocols that relies on the SmallWorld encounter patterns. This part is a generalization of the *profile-cast* service shown in chapter 5, and we name this protocol as *CSI*, since its a new *C*ommunication protocol based on the *S*table yet *I*mplicit structure in human mobile networks. We present its detailed design in chapter 7.

1.5 Contributions

In this study we have made the following contributions:

1. Over the course of several years, we have collected user traces from WLANs in the University of Southern California and the University of Florida. With the help from the corresponding network administrators in the schools, we are able to obtain extensive, campus-wide measurements of the activities of the campus network users. We have gained the understanding of what information to collect to maximize the usage of the data. In the future, the collected traces will be shared with the research community through our project website [1].

2. We have built a rich set of different representations and analysis tools to investigate various aspects of the traces. As mentioned earlier, these tools reveal various behaviors of users in the trace, ranging from *microscopic* individual mobility to *macroscopic* network-wide encounter patterns.

3. We build the *time-variant community mobility model* based on the insight gained by studying the traces. This model provides a flexible and scalable platform on which researchers can set up a wide range of scenarios for MANET protocol and service evaluations. The code for the mobility trace generator is available at [9].

4. We propose a matrix representation based on *long-run user mobility preference* and a summarization technique to extract important features from the matrix. Then we construct a *distance* or *similarity* measure between users based on these features. The distance metric can be used to classify users into distinct groups, and the identification of such groups provides useful information for the network administrators.

5. We propose the *profile-casting* service for message delivery in mobile networks, a new communication paradigm in which the properties of individual users, instead of the network identities, are used to identify the desired destination nodes. We believe the *profile-cast* approach is more suitable than the traditional identity-centric approach, especially when the network is highly dynamic. The *profile-cast* service incorporates the understanding gained from detailed studies of the environment, such as the similarity metric mentioned above and the SmallWorld encounter patterns, into the protocol design phase.

CHAPTER 2
RELATED WORK

In this chapter we discuss related work in the literature. Since the analysis in the dissertation uses the collected traces, we first discuss about recent efforts of the research community to gain access to user behavioral measurements from large-scale in-provision networks or small-scale testbeds in section 2.1. We further discuss existing analyses done on the collected data sets in section 2.2 to put our research in context. We also introduce the work related to our two major applications, mobility modeling and message forwarding protocol design in the *delay tolerant network (DTN)* framework, in sections 2.3 and 2.4 respectively.

2.1 Trace Collections

Collecting traces has long been considered an empirical way to understand the dynamics in large systems realistically. Large-scale data analysis has proven helpful to unearth hidden trends and understand deeply the dominant dynamics in the system. Several classical examples of findings from trace analysis that further lead to high impact research are the discovery of the self-similar traffic from packet trace analysis [18] and the identification of power-law distributions in the node degree from network topology traces [19]. In both cases, the work was made possible by extensive collection of relevant data sets.

For the research in MANETs, nodal mobility is one of the major components to understand as the mobility changes the network connectivity and hence impacts the system-wide performance on many fronts. Therefore, there has been extensive efforts to collect user mobility traces through various methods. One straight-forward way to obtain mobility information is through close observations [22] of the moving users or surveys [21]. These approaches, although beneficial, have severe limitations in terms of its scalability – if human effort is involved, it is difficult to repeat the trace collection process to include a large population.

To enable large-scale data collections, researchers leverage the existing last-hop wireless network infrastructures to collect location information of the users. One prominent example of such networks is the wireless LANs (WLANs). User trace collection in WLANs started by the seminal work of Tang and Baker, who collected the WLAN traces from both an academic environment (i.e., a research building on a university campus) [23] and typical daily life (i.e., city-wide access points) [24]. Their effort has been followed by many researchers, including Balazinska and Castro [10], McNett and Voelker [11], Kotz and Henderson et al. [12, 13], and Papadopouli et al. [14], each collecting WLAN traces from infrastructures with different sizes and user populations. Among these efforts, Balazinska and Castro focus on WLAN users in three corporate buildings [10], McNett and Voelker collect usage traces specifically for hand-held devices (i.e., PDAs) [11], and the other trace collections [12–14] are obtained from generic users on university campuses. We have also collected WLAN traces from University of Southern California [15] and University of Florida campuses. Most of these traces are collected passively, i.e., there is no need for the WLAN users to actively participate in reporting any data. The access points (APs) and sometimes other logging servers in the network passively monitor the association and usage of individual users. The only exception is [11] where the researchers install reporting software on the PDAs to keep track of all APs in its communication range. The passive trace collection approach is usually more scalable as it does not require software installation or proactive participation from the users. With this method, traces with more than thousands of users are not uncommon. These efforts lead to rich data sets to understand user mobility in WLANs, especially on university campuses from where most of the traces are obtained. In addition to the location information revealed by the association with APs, the WLAN usage of each user (i.e., the amount of traffic sent/received) is usually also logged, widening the potential usage of the traces.

In addition to WLANs, other possibilities are also leveraged to collect traces. In the Reality Mining project [16], Eagle and Pentland program the cellphones of the

participating users to log the cellphone base stations they associate with, the mutual encounters between the cellphones via bluetooth probing, and activities on the cellphones (e.g., call history). While cellphones are probably the most popular wireless devices, large-scale traces are not yet released from the cellular phone operators due to privacy concerns. The traces from cellphones in the research community are mostly actively collected through additional programs on the participating devices (as in [16]), hence its scale is usually not comparable to the passively collected WLAN traces. However, it is note-worthy that large-scale user location traces collected from the cellular phone networks do exist, and as of the dissertation writing, studies of cellular user behavior also emerge in the literature (e.g., [25]). The data sets, at this moment, are only available to the cellular system operators. Along a different line, there are also efforts in collecting vehicle movement traces, in most cases through GPS positioning system. One example is the Cab Spotting project [17] which logs the location information of participating taxis in the greater San Francisco area. Projects of this nature also require active reporting from the monitored vehicles, hence the participants are usually in the order of hundreds.

More recently there are several testbeds deployed to collect encounter events (i.e., when devices move into the communication range of each other) between moving objects. The objective of these projects is to understand the emergence of communication opportunities between the devices carried by moving human beings. The Haggle project [26] focuses on the scenarios named as the pocket-switch networks, i.e., the users carry miniature devices equipped with short-range radio in their pockets, and these devices log their mutual encounters as potential communication opportunities. They have carried out experiments in conference settings at INFOCOM 2005 and 2006 [27] and in research labs. Experiments with a similar objective are also performed by Su et al. [29] in a university campus setting and by Leguay et al. in a college town [28] setting. These data sets can be leveraged, for example, to evaluate routing protocol performances in DTNs in empirical environments. While human mobility has received relatively more

attention, other experiments also focus on the encounter patterns between vehicles (e.g., the DieselNet project [30]) and wild animals (e.g., TurtleNet [31], ZebraNet [32], and whale tracking [33]).

Due to the fast emergence of traces, the research community tries to organize websites for archiving or maintaining pointers to the relevant traces. These websites help to provide better accessibility for researchers to locate the resources in the community. Two prominent examples of such websites are [1] and [2]. Most of the traces we use in this dissertation can be found on either website.

In addition to utilizing existing data sets in this dissertation, we have also conducted efforts in collecting data sets ourselves. We have been collecting WLAN traces from the University of Southern California since summer 2005, and part of the traces has been made available through the MobiLib website [1] established by Dr. Helmy. This data set consists the location information of wireless users on USC campus, and the netflow information (their usage of the network)[1] . We have also started a similar effort at the University of Florida with more detailed information[2] . The current trace collection includes not only the association with the access points (i.e., the location information) of WLAN users, but also the log-in and log-out timestamps from the authentication servers (this is a required step for UF wireless network users to access the Internet) and the amount of traffic sent/received in 30-minute intervals. This additional information of user login potentially allows us to distinguish between the scenarios when users actively uses the network versus when users turn on their computers but do not intend to access the network (e.g., when the users merely process the documents on the local host, they do not have to login to use the WLAN). We plan to make the trace available once we clean it

[1] Special thanks to Mr. Brian Yamaguchi and Carl Hayter at USC Information Technology Services for helping us collect the WLAN traces on the USC campus in the past three years.

[2] Special thanks to Mr. Marcus Morgan at UF Information Technology & Services for helping us collect the WLAN traces on the UF campus in the past year.

up and add sufficient anonymization to protect user privacy (this is an on-going effort by fellow researchers in the NOMADS research group [102]). Furthermore, through the course of the research work, we have obtained a good collection of pointers to existing data sets. We put these pointers online for the ease of references for us and the research community in the future. Please refer to the MobiLib website [1] for more details.

We have also tried out other techniques for the collection of mobility related information. One example is giving out surveys [21, 77] and asking people about their movement patterns. This approach is helpful in building the insights about mobility, but does not scale very well. Some of my early research work built on top of the surveys is summarized in the Appendix.

2.2 Trace Analysis

In this section we discuss the analysis based on the traces in the literature. We split the section into several subsections, depending on the methods used in the analysis. The focuses of the subsections are (1) General statistics, (2) Data mining techniques, and (3) Graph analysis.

2.2.1 General Statistics

In the incipient stage of WLAN trace based analysis, the researchers focus on understanding how users use the WLAN. In many early works in this area, a lot of basic statistics of users, such as average number of online users, average session duration, bytes sent and received, and protocols used, are included [10–13, 23, 24]. In addition to understanding individual users, the researchers also consider the WLAN from which the trace is collected as a system, observe how users utilize the system as a whole, and display the relevant statistics, such as average number of users per access point (AP), the distribution of AP popularity, and user handoff frequency between access points. Such an approach is natural as most of the current trace collection efforts obtain the traces from a single administrative entity (in most cases, from university campus networks, e.g., [11–13, 23]). These statistics help the researchers to gain understanding of user behaviors.

However, most of these works do not try to compare the findings between different traces. Furthermore, as the trace collection methods and the statistics presented in each study are not standardized, sometimes comparing the results is difficult. It is hence unclear whether a finding based on a particular trace is a general phenomenon among wireless network deployments. To improve upon this shortcoming, in this dissertation we consider *multiple* traces and analyze multiple traces with the same methods (see chapter 3 for details) in order to generalize the findings beyond a specific environment, enabling us to discuss about generic user behaviors. In order to achieve that, we have to apply appropriate normalizations to make the traces collected with different techniques comparable, and identify relevant metrics of user behaviors from different contexts.

With these traces available, later research works focus on characterizing user behaviors in wireless LANs. One particular important aspect, as mentioned earlier, is the mobility of users. Balazinska and Castro provide an analysis with special focus on user mobility, defining the notion of *home location* and two quantitative measures, *persistence* and *prevalence*, to gauge user mobility [10]. This provides a broad classification of users based on the degree of their mobility, but does not completely describe their detailed behavior (e.g., how the users split their online time to various locations). Along the line of modeling user association to access points (APs), in [34] the authors propose to cluster APs based on the time of peak user arrivals. In [35] the focus is on the arrival patterns of users at APs and the authors propose to use time-varying Poisson processes to model the arrival patterns, and further identify clusters of APs based on the parameters in its arrival process. These modeling efforts focus more specifically on capturing the changes of numbers of users associated with the APs by modeling the arrival and departure processes, hence the resulting models capture the dynamics of the users of an access point in aggregation (i.e., the variation of the total number of users associated with a particular AP) rather than the dynamics of the *individual user*. In contrast, we take a holistic view at modeling associations of *individual users*, and observe several aspects

of user association. In chapter 4, we propose a general framework which is applied to capture fundamental aspects of user association behaviors in the WLAN traces (e.g., User activeness, preferences in association, handoff, and repetitive periodic patterns in association) that can be used to build models for WLAN users. The major findings we observe from multiple WLAN traces at hands include the *on-off behavior*, the *skewed location visiting preferences*, the *hand-off*, and the *periodical re-appearance* of nodes. We use these metrics to provide a complete description for users' mobility-related behaviors in wireless networks. By studying multiple traces from different environments collected at different times, we are able to establish that most traces display similar trends, but the details differ due to differences in user population, environment, time, and methodologies of trace collection.

We believe that the design and evaluation of the next generation wireless networks should go hand-in-hand with deep, insightful understanding of the realistic environments in which they will be deployed and used. However, the WLAN traces studied in this dissertation do not provide directly nodal mobility models, as they represent the combined effects of coarse-grained (i.e., per-AP granularity) nodal mobility, plus the on-off usage patterns of the device owners and the influences of wireless signal propagation in the environments. In that sense, one may envision that all-encompassing models may be built by studying the traces. Understanding of such realistic scenarios sheds lights on sometimes falsely taken assumptions in over-simplified *random mobility models* (such as nodes holding the same probability to visit all locations or behaving similarly through the whole simulation period), and quantifies the detailed behaviors of users so that future models can incorporate them. We will further discuss this point regarding mobility models in section 2.3 below.

As a side note, there also exists analysis of other aspects than the user mobility based on the collected traces. For example, in [37] the authors propose models to describe traffic flows generated by WLAN users. This points out the wide applicability of the traces for

empirical studies. In this dissertation, we choose to focus on the user associations (in particular, on user mobility modeling and differentiating users based on mobility) obtained from the traces and its applications, and leave other aspects (such as traffic) out for future work.

2.2.2 Data Mining Techniques

Although user association pattern has been one major focus in studies about WLANs, for most works mentioned previously, the focus is either on aggregated statistics or on association models for individual user. There are hardly any studies on understanding the relationships between users in the literature. In chapter 5 we attempt a data mining approach to understand the relationship between users in the large-scale data sets. More specifically, we define similarity metrics between individual users based on their preferences of association, and leverage *unsupervised learning* techniques (i.e., clustering) to identify groups of coherent behavior from the diverse user population.

Along this line of research, there are only a few previous works that use data mining techniques to classify users. The earliest example is by Tang and Baker[24], where they classify Metricom users into groups with a two step procedure. The first step classifies the users based on mobility-related statistics, such as number of locations visited and distance moved. Each group identified in the first step is further classified in the second step based the activeness of the user (i.e., quantified by the events generated by the user) during the day. Another example we are aware of is [38], where Kim et al. classify users based on the periodicity and the movement range. Specifically, in their paper, users are classified based on the dominant periods in their movement (i.e., classified into groups that display strong daily or weekly movement patterns) and their longest movement ranges. They classify users based on different *representations*, hence the results have different interpretations to ours. In particular, their classification of users is based on high-level behavioral statistics, while our classification of users is based the fine-grained location preferences hence more

detailed behavioral groups can be identified. This provides a different and important perspective to understand user association patterns.

There are several other papers in the literature of trace analysis that also use clustering techniques for different objectives. For examples, in [39] the authors apply clustering techniques (K-means and Gaussian mixture model) to the trace of location coordinates of the same user at many different time instants to discover significant places for the user, but they have not focused on classifying users. In [78], the authors use the mutual encounter frequencies between nodes to identify the underlying communities, where the notion of a community refers to a group of nodes who remain in contact for long periods of time. The clustering is done based on communication opportunities available between the devices, not the similarity between user behaviors (However, we also note that high similarity in nodal mobility does lead to better communication opportunity between the similar nodes, hence there is a correlation between the two approaches).

In this work we represent the association history of each user in a matrix form, and utilize singular value decomposition[41] to obtain the association features from users. Singular value decomposition (SVD) is widely-applied to discover linear trends in large data sets. It is closely related to principal component analysis [40]. In [42], the authors utilize PCA to decompose the traffic flow matrices for ISP networks and understand the major trends in the network traffic flow matrices. Our application of SVD to individual user association matrices is similar in spirit to their work. Note that it is typical for people to follow dominant routines in lives, hence we expect the SVD approach to be applicable to various human behavioral data sets. In [16], the authors also use PCA to discover trends in a cellphone user group, which is similar to our analysis on individual users. In this dissertation, in addition to analyzing much larger data sets than the data set used in [16], we further quantify user similarity by defining distance metrics to classify wireless network users into groups with robust validation. Note that in order to make the eigen-behavior vectors obtained from all users comparable, we need to keep the origin

fixed among all association matrices. Hence we adopt a variant, called *uncentered PCA* [40] where the mean of each dimension is not subtracted. It has been used to study the diversity of species at various sites[43] in biology literature.

2.2.3 Graph Analysis

In chapter 6 we look into the encounter patterns (i.e., the patterns of wireless devices moving into communication range of each other) of the users in the traces. We seek to understand the global structure of the *relationships* between users in the traces. Specifically, we provide new perspectives to study the WLAN traces by looking into encounter distributions and utilizing the Small World theory to describe the encounter relationship between users as a graph. The Small World graph model is proposed in [8] and widely utilized to describe various networks in many areas, such as social networks, Internet topology, and electrical power networks [44]. In [45] the author applied the concept of Small World to devise a contact-based resource discovery scheme in wireless networks. Two prominent features of Small World graphs are high clustering coefficients comparable to the *regular graphs* and low average path lengths comparable to the *random graphs*. The emergence of the Small World properties indicates there is a correspondence between the encounters of devices in this traces and the fundamental social relationship between their owners, as the Small World network property is an important characteristic of the social networks of human beings.

In [46] Bai and Helmy find that, under mobility models with homogeneous behaviors (i.e., Each node follows exactly the same model with some randomness), eventually each node encounters with all other nodes in the network (i.e. achieving 100% encounter ratio). However, the empirical observations from large WLAN traces show very different behaviors, with most nodes encountering only a very small portion of the whole population, during a time frame as long as a month. This observation indicates that the user populations in larger environments, such as university campuses, are actually

not homogeneous. This also leads to the need of a flexible mobility model to capture the non-*i.i.d.* mobility of the population, which we discuss further in section 2.3 below.

Similar graph analysis of potential communication opportunities has also been done based on student class schedules in a university in [47]. The authors create a graph in which each student is represented as a node, and assume that students registered in the same class form links between the nodes corresponding to these students. They show that this graph also displays Small World properties. We must note, however, that the class registration information is an indirect indicator of the physical locations of the students, and hence does not directly translate into a graph of communication opportunities. In addition, the class schedule does not capture the mobility patterns outside of the classes. Using the traces for *actual* location information of the devices, by our discretion, seems to be a better information source to construct the communication opportunities graph (albeit it is coarse-grained location information).

2.3 Mobility Modeling

Mobility has been long recognized as one of the fundamental components that impact the operations of MANETs. On one hand, mobility presents itself as a challenge for network designers to overcome, as nodal mobility changes the topology of networks, breaks established wireless links, and overall makes reliable communication difficult. On the other hand, mobility also provides new opportunities. It is shown that mobility improves the scaling law of system-wide capacity to $O(1)$ as network density increases[48]. More recently, mobility has been utilized as the enabling factor for message delivery in delay tolerant networks (DTNs[4]), where a complete path from the source node to the destination node does not exist at any time instant, broadening the scenarios in which communication networks can be established. It is also an important system variable to consider for protocol performance analysis, as it is shown that different underlying mobility models change the performance ordering of various MANET routing

protocols[20]. Therefore designing good mobility models has become a topic that attracts significant attention from computer network researchers.

A wide variety of mobility models are available in the research community. See [49, 50] for a good survey. Among all mobility models, the popularity of *random mobility models* (e.g., random walk, random direction, and random waypoint) roots in its simplicity. They are not only easy to generate, tune and scale, but also amenable to mathematical analysis that reveals important fundamental properties in mobility, such as the stationary nodal distribution[51], the hitting time, the meeting time[52], and the meeting duration[53]. These quantities in turn enable routing protocol analysis to produce performance bounds[55–57]. However, *random mobility models* are based on over-simplified movement rules, and as we will show in chapter 4, the resulting mobility characteristics are very different from real-life scenarios observed from the real traces. Hence it is debatable whether the findings under these models will directly translate into performances in real-world implementations of MANETs. More recently, an array of *synthetic mobility models* are proposed to improve the realism of the simple *random mobility models*. More complex rules are introduced to make the nodes follow a popularity distribution when selecting the next destination[21], stay on designated paths for movements[59], or move as a group[58]. More variants of mobility rules can be found in various models[49, 50]. These rules enrich the scenarios covered by the *synthetic mobility models*, but at the same time make theoretical treatment of these models difficult. In addition, most *synthetic mobility models* are still limited to *i.i.d.* models (in which every node behaves statistically the same), and the mobility decisions are also independent of the current location of nodes and time of simulation.

A different approach to mobility modeling is by *empirical mobility trace collection*. Along this line, researchers have exploited existing wireless network infrastructure, such as wireless LANs (e.g., [10, 13]) or cellular phone networks (e.g., [16]), to track user mobility by monitoring their locations. Such traces can be replayed as input mobility

patterns for simulations of network protocols, as in [60, 61]. More recently, DTN-specific testbeds[27, 29, 30, 32, 33] aim at collecting encounter events between mobile nodes instead of the actual mobility patterns. However, most of these works (except [27]) do not include detailed mathematical analysis for the mobility characteristics. Also, due to the experimental nature of these studies, the size of the traces and the environments in which the experiments are performed can not be adjusted at will by the researchers. To improve the flexibility of the traces, the approach of *trace-based mobility models* have also been proposed[62–65]. Based on the collected traces, these models discover the underlying movement rules that lead to the observed properties (such as nodal distribution, duration of stay at locations, arrival patterns, etc.) in the traces. Statistical analysis is then used to determine proper parameters of the model to match it with the trace.

Ideally, a good mobility model should achieve a number of goals: (i) it should first capture *realistic* mobility patterns of scenarios in which one wants to eventually operate the network; (ii) at the same time it is desirable that the model is *mathematically tractable*; this is very important to allow researchers to derive performance bounds and understand the limitations of various protocols under the given scenario, as in [27, 48, 56, 57]; (iii) finally, it should be *flexible* enough to provide qualitatively and quantitatively different mobility characteristics by changing some parameters of the model, yet in a repeatable and scalable manner; designing a new mobility model for each existing or new scenario is undesirable.

Most existing mobility models excel in one or, less often, two aspects of the above requirements, but none satisfies all of them at the same time. The most widely used mobility models are *random mobility models* such as random walk, brownian motion, random direction, and random waypoint [49, 50]. Their strength is the theoretical tractability but their weakness is the lack of realism. More complicated *synthetic mobility models* (e.g., [21, 58, 59]) improve the realism, but most of the time at the expense of theoretical tractability. More recently, a large number of *empirical mobility traces* from

real mobile users have been collected [10, 11, 13, 16, 27]. Although one can use such traces directly in an evaluation with excellent realism, these traces are usually rather inflexible and provide only a single snapshot of the underlying mobility process. To address these two issues, *trace-based mobility models* [62–65] have been proposed (i.e. larger, more flexible synthetic traces created from the smaller empirically collected ones). Yet, most of these models do not possess the necessary flexibility to match mobility characteristics of traces other than the ones on which they are based.

As an application of the observations we make on the *individual user mobility characteristics* in chapter 4, we combine the strengths of various approaches to mobility modeling mentioned above and propose a *realistic, flexible, and mathematically tractable synthetic mobility model*. Large-scale deployments of WLANs in university[11, 13] and corporate[10] campuses provide excellent platforms in which huge amount of user data can be collected and analyzed. We leverage these traces to understand empirical user mobility, and propose a time-variant community mobility model based on the prominent mobility characteristics observed. We differentiate our work and other trace-based models ([62–65]) in several aspects. First, while the previous works emphasize the capability to truthfully recreate the mobility characteristics observed from the traces, we go beyond that and emphasize, in addition to the realism, the mathematical tractability of the model. This additional feature facilitates the application of our model to performance prediction of various communication protocols. Second, we abstract the observed mobility characteristics from WLAN traces, and propose a mobility model that has wider applicability – in addition to WLANs, it can be tuned to match with other types of traces, such as a vehicle mobility trace[17], and even with other characteristics in other traces of human mobility (e.g., the encounter duration and the inter-encounter time in [27]).

Our time-variant community mobility model (in short, the *TVC model*) is built upon our previous work[66] presented in section 4.1, in which we identify several prominent properties that are common in multiple WLAN traces. The *TVC model* extends the

concept of community model proposed in [52] by introducing time-dependent mobility and hence inducing periodical behavior of the nodes. Although capturing time-dependent behavior is suggested in [65], it has not been incorporated in their model. Among all efforts of providing realistic mobility models, to our best knowledge, this is the first work to explicitly capture time-variant mobility characteristics. The TVC model presented in this dissertation is a generalization of the previous conference version[67].

The concept of community is also mentioned in [68] in a different context. The authors assume the attraction of a community (i.e., a geographical area) to a mobile node is derived from the number of friends of this node currently residing in the community. In our paper we assume that the nodes follow location-based preference to make movement decisions, and each node moves independently of the others. Mobility models with inter-node dependency require a solid understanding of the social network structure, which is an important area under development. We choose to leave this as future work.

2.4 Message Forwarding Protocol Design in DTNs

In recent years, packet forwarding in sparse, frequently disconnected MANETs has received increasing attention from the research community. In such network scenarios, a complete end-to-end path from the source node to the destination node is usually unavailable in the space domain at any given moment. Therefore, mobile nodes have to store copies of packets in the memory and carry them across the network with nodal mobility, and later deliver the packet when the mobile nodes encounter with other nodes in the network. Such network scenarios are generally known as delay-tolerant networks (DTNs [4]). In DTNs, packet routing relies on not only the spatial connectivity, but also temporal change of nodal positions (i.e., mobility) to be successful[69]. Most of the previous work in this area focus on designing packet forwarding heuristics [54, 56, 57, 61, 71, 74, 93, 94]. In general, different degrees of knowledge of mobility pattern is assumed[54], or an *i.i.d. random* mobility model is used[56, 57]. Some protocols (e.g., [61, 74]) seek to discover promising leads to the destination node based on nodal

mobility or encounter patterns. This approach is suitable where the nodal mobility follows a non-*i.i.d.* pattern, which is generally the case in real life. There are also protocols leveraging simple strategy such as relying on the age of the last encounter events between nodes to discover promising leads to the destination node [93, 94]. An analysis on these protocols shows that each node has to encounter with a high percentage (i.e., more than 30%) of other nodes before the selected paths become stable [46]. As we discover empirically from the WLAN traces in chapter 6, this is usually not achieved in a diverse, large-scale environment such as university campuses, where on average a given node encounters only around 6% of the whole population. Hence it becomes an issue worth investigating that (1) would message delivery be successful in such a sparse (in terms of the available encounter events) network? and (2) how to design good message forwarding strategies in such environments?

In this dissertation we take a behavior-aware approach to message forwarding in the DTNs. In general, our goal is to make the forwarding protocol to be aware of the behavioral patterns of the individual users when making the forwarding decisions, and leverage the encounter patterns to facilitate the message forwarding. We split the task into several components in the dissertation – (a) we incorporate user behavioral patterns in a DTN message forwarding protocol, designed for a special case of sending messages to users who are similar to the sender in chapter 5, (b) we discuss about understanding of global user encounter patterns in chapter 6, and (c) we show how to leverage the encounter patterns in protocol design with more generic scenarios in chapter 7.

In chapter 5, we propose a new service paradigm named *profile-cast*. In *profile-cast*, the destination node(s) are not identified by their network identities (e.g., network addresses), but by their affiliations and behavioral patterns (i.e., the *profile* of the node). *Profile-cast* is related to multi-cast as both of them target groups of receivers. However, in *profile-cast* the intended receivers are defined by their intrinsic properties, and there would be no explicit join to subscribe to a group as in multicast. Managing group membership

in highly dynamic networks such as DTNs has attracted some attention recently [70] but it is still a hard problem to solve. The goal of our *profile-cast* service is to leverage underlying behavioral patterns (i.e., the *profiles*) to guide message delivery, which ties naturally to many context-centric services in mobile networks, such as searching and targeted announcement/advertisement.

We leverage *mobility-based profile-cast* as an example in the case study presented in chapter 5. The goal of our application is to deliver a message to the node(s) who have similar mobility profiles as the sender itself, without knowing their network identities beforehand. The forwarding protocol uses the characteristics of nodal mobility, which we referred to as the *mobility profiles* of users, to guide the propagation of messages among the nodes. Note that this application is different from *geo-cast*[72], which targets at the nodes *currently* within a geographical region as the receivers. Our target receivers are nodes with a certain mobility profile, *regardless* of their actual locations at the time the message is sent. With the case study of *mobility-based profile-cast* we show that understanding user behavioral pattern can be helpful in designing routing protocols or services. This success is directly based on the fact that *mobility profile* can be used as a distinguishing feature of the mobile users, as discussed in the first half of chapter 5 (based on the data mining approach to trace analysis). However, this case study applies only to sending to a group with similar *mobility profile* to the sender, a very specialized case.

We wish to further enlarge the scope of the *profile-cast* paradigm to include other types of user profiles as descriptors for potential destinations. In some cases, the target profiles could be even independent of the nodal mobility patterns. The fore-mentioned goal leads us naturally to the idea of leveraging the encounter patterns to disseminate copies of messages in the network efficiently. We seek a way to efficiently spread the message to the whole network so that the potential recipient nodes can easily retrieve a copy of the message.

In chapter 6, we first start from an empirical point of view, and investigate the issue of whether the store-and-forward model is potentially feasible with a simple forwarding strategy under the current encounter patterns of wireless devices derived from the WLAN traces. Our initial findings are encouraging: We first use the epidemic routing [71] to test the reachability of the network (i.e., if the current encounter patterns lead to a network in which most nodes are reachable). It turns out, not only most nodes are reachable, but the encounter patterns lead to a robust network – even if some nodes are uncooperative, or encounters with short durations are considered not useable, messages still propagate well in the network. This suggests the possibility of designing a learning protocol to identify nodes with different roles in the underlying Small World encounter pattern, and make the message dissemination more efficient (i.e., reducing the high overhead associated with the epidemic routing).

In chapter 7 we then discuss the design of this more generic message dissemination protocol. The difference between the *CSI* protocols in chapter 7 and the case study in chapter 5 is the following. In section 5.8 we focus on only sending messages to users with similar behavioral profile to the sender. In CSI we introduce the notion of the *target profile* to decouple the behavioral profile of the sender from the destination profile in the message. This significantly enhances the capability of the message dissemination schemes, by allowing the sender to specify target behavioral profile (in CSI:T mode), or even some target profiles that are orthogonal to the behavior based on which we measure the similarity between users (in CSI:D mode).

In the design process of the CSI protocols, we conduct the first detailed systematic study on the spatio-temporal stability of user behaviors in mobile societies, a new dimension that has not been considered before. Our effort on the extraction of behavioral profiles and behavior-based user classification is related to the reality mining project [16] and the work of Ghosh et al. [111]. We leverage the representation of mobility preference matrix defined in chapter 5, which reveals more detailed user behavior than the five

categories representation used in the reality mining [16] and the presence/absence encoding vector used by Ghosh et al. [111].

The major application considered in chapter 7 is to design a message dissemination scheme in decentralized environments. While several previous works exist in the delay tolerant network field, most of them (e.g. [61, 71, 74, 76, 107]) consider one-to-one communication pattern based on network identities. The *profile-cast* communication paradigm targeted at a behavioral group is a new paradigm in decentralized environments. Some of the previous works assume existing infrastructure: PeopleNet [110] uses specialized geographic zones for queries to meet. The queries are delivered to randomly chosen nodes in the corresponding zone through the infrastructure. Others (e.g., [74, 107]) rely on persistent control message exchanges (e.g., the delivery probability) for each node to learn the structure of the network, even when there is no on-going traffic. From the design point of view, our approach differs from them by avoiding such persistent control message exchanges to achieve better energy efficiency, an important requirement in decentralized networks.

The spirit of our design is more similar to the work by Daly et al. [76], in which each node learns the structure of the network locally and uses the information for message forwarding decisions. However, the learning process proposed in [76] still involves message exchanges about past encounters, even in the absence of actual traffic. Our goal, on the other hand, is to design the protocol so the nodes rely on the intrinsic behavioral pattern of individual users to "position" themselves in the behavioral space in a localized and fully distributed manner, without any message exchange between nodes. The use of user behavioral profiles to understand the structure of the space is similar to the mobility space routing by Leguay et al. [61] and the utility-based routing by Aiklas et al. [105]. The major differences between our work and [61, 105] are two fold: First, we design the CSI:D mode, in which the target profile need not be related to the behavioral profile based on which the message dissemination decisions are made. Second, we also provide a

non-revealing option in our protocol, thus no node has to explicitly reveal its behavioral pattern or interests to others, as opposed to [61, 105].

CHAPTER 3
DATA SETS

We introduce the data sets we include in the dissertation and the definitions of the terms we use throughout the dissertation, and discuss the normalization techniques to make data sets collected by different methods comparable. We also present the on-going efforts of collecting traces from the University of Southern California and the University of Florida campuses, and the available information from these two traces.

3.1 Data Sets Used

In this dissertation we mainly focus on wireless traces collected from university campuses and corporation networks. We obtain wireless traces from various sources, including totally over 15,000 distinct users and over 1,300 APs. To our best knowledge this is the most extensive study of user behavior in wireless networks so far. Among the traces, the USC and UF traces are collected specifically for the purpose of our studies, while Dartmouth [13], UCSD [11], and MIT [10] traces were collected by other research groups. We summarize the important characteristics of these traces in Table 3-1 and explain the major issues below. For comparison purposes, we also include one trace of encounters between portable wireless sensors deployed at INFOCOM 2005 by the researchers in the Haggle project [27], and one vehicle mobility trace obtained from the Cab-spotting project [17].

For longer traces such as the Dartmouth [13] trace, the USC trace, and the UF trace, we take parts of the data sets for each case study in the dissertation. We make such choices to facilitate the processing of data and focus our analysis on smaller, tractable parts of the data sets. The chosen parts are representative of the data set as a whole, and similar conclusions can be drawn if we had chosen other parts of the data set. We will explain our choices further in each case study. As shown in Table 3-1, we use different tags to refer to various parts of data sets from the same trace. In some cases, we apply multiple

post-processing techniques to the same data set and compare the results to understand the impact of the processing steps.

These traces are chosen to represent different environments. We study the differences and similarities of user behavior in these traces, and try to attribute them to the underlying differences in the corresponding environments as appropriate. In order to make the results comparable between traces, in chapter 4 and 6, we analyze selected one-month periods from the longer Dartmouth, USC, UF, and UCSD traces. For the UCSD trace, we choose the first month, as the user activity decreased during their study due to loss of interest in participation and some minor problems in trace collection[11]. We select two one-month periods from the Dartmouth trace: July 2003 (*Dart-03*, during the summer vacation) and April 2004 (*Dart-04*, during the spring quarter). For the USC trace, we pick the first available month for the detailed trace; for the UF trace, we pick the first month of spring 2008 and randomly pick $10,000$ users from the relatively large user population (see descriptions for USC and UF traces in section 3.4). The MIT, Dartmouth, USC, and UF traces collect measurements of generic wireless network users, including but not limited to laptops, PDAs, and VoIP devices. The UCSD trace is from a specific project to study the behaviors of PDA users. To further compare the association behaviors of smaller, handheld devices (e.g., PDA, VoIP devices) with generic wireless devices in the same environment, we also separate the PDA (*Dart-PDA*) and VoIP device (*Dart-VoIP*) users from the Dartmouth trace during April 2004, and study their behavior specifically. However, according to the device type information provided in Dartmouth trace archive[85], there are only 25 PDA users and 63 VoIP device users during this time period. The results we get from these small sample sizes may need to be verified by studies in larger scale.

Table 3-1. Statistics of studied traces

Trace source	Unique users	Unique APs	Unique buildings	Trace duration	User type	Environment	Trace collection method	Analyzed part in this dissertation	Users in analyzed part	Labels used in graphs
MIT[82]	1,366	173	3	Jul. 20 '02 to Aug. 17 '02	WLAN Generic	3 Engineer buildings	Polling	Whole trace	1,366	MIT-cons MIT-rel
Dartmouth[81]	10,296	623	188	Apr. '01 to Jun. '04	WLAN Generic	University campus	Event-based	Jul. 2003	2,518	Dart-03
								Apr. 2004	5,582	Dart-04 Dart-rel Dart-cons
					PDA only				25	Dart-PDA
					VoIP device				63	Dart-VoIP
					WLAN Generic			04/05/04 - 06/04/04 (Spring quarter 04)	6,582	Dart-04spring
UCSD[83]	275	518	N/A	Sep. 22 '02 to Dec. 8 '02	PDA only	University campus	Polling	Sep. 22 '02 to Oct. 21 '02	275	UCSD
USC[80]	4,548	79 ports	N/A	Dec 03 - Dec 04 (trap) Apr 20 05-Now (detail)	WLAN Generic	University campus	Event-based	Apr. 20, '05 to May. 19 '05	4,528	USC
	25,481	137 ports						01/25/06 - 04/28/06 (Spring semester 06)	5,000	USC-06spring
UF	44,751	728	N/A	August 2007 to Now	Generic	University campus	Event-based	Jan. 14, '08 to Feb. 13, '08	10,000	UF
Cambridge[84]	41 internal nodes	N/A	N/A	Mar. 7 '05 to Mar. 10 '05	iMOTE	conference	Polling	Whole trace	41	Cambridge-INFOCOM05
Cab-spotting [17]	549 taxis	N/A	N/A	Sep. 22 '06 to Nov. 1 '06	GPS devices installed on taxis	Greater San Francisco area	Polling	Whole trace	549	Vehicle-trace

All the WLAN traces, except the MIT trace, are collected from the entire campus wireless network. The MIT trace is collected from three engineering buildings in a corporation network, hence its user population is not as diverse as the other traces, and the geographic scope of trace collection is smaller. The USC trace is the only one that has coarser, per switch port location granularity, while the others have per AP location granularity. Each switch port in the USC trace has several APs connected to it. The geographic coverage of a switch port approximately corresponds to a building (or several small buildings in close vicinity) on the campus.

In chapter 5 and 7, we use longer traces to study the trends in user association preferences and how we could utilize such information to classify users. We use two semester-long traces from the longer Dartmouth and USC traces, since a semester (or a quarter) is the typical longest time period for which the behavior of users on university campuses remain consistent. For Dartmouth we pick the spring quarter of 2004, which includes the activities of $6,582$ users during 61 days. For USC we pick the spring semester of 2006, which includes the activities of $25,481$ users during 94 days. To make the task manageable, we analyze only the most active $5,000$ users for the USC trace.

In order to compare the WLAN user behaviors with the behaviors of users in other environments, we also include two additional traces in the study. The *Cambridge-INFOCOM* trace is collected from participants of an experiment carried out at INFOCOM 2005 conference[27]. Each participant is given a wireless sensor (an Intel iMote) and asked to carry this device throughout the conference. These devices keep track of other Bluetooth devices (including all 41 iMOTEs deployed for the experiment and other Bluetooth devices in the conference vanue as well) and record the devices that are within its communication range at regular intervals. In other words, this trace collects the *encounter* events between the wireless devices. The other trace, *Cab-spotting*, is available from the Cab-spotting project [17]. This is a project for tracking the locations of participating taxis in the greater San Francisco area using GPS devices installed on the

cabs. Specifically, we use this trace to compare some mobility characteristics between two very different groups – WLAN users on a university campus and taxis in a metropolitan area. As we shall see in chapter 4, these two seemingly very different groups share some similar mobility characteristics. In chapter 6, we compare the encounter patterns (see its definition below in section 3.3) between the WLAN users with the users in the encounter trace collected at INFOCOM 2005.

3.2 Trace Collection Methods

The methods of collecting WLAN traces can be categorized into two major categories: (i) Polling-based methods which record the association of the mobile nodes (MNs) at periodic time intervals, using SNMP (in the MIT trace[82]) or association tracking software on the MNs (in the UCSD trace[83]), and (ii) Event-based methods which record MN online/offline events using logging server (e.g. syslog) [80, 81]. For the Dartmouth trace we use the derived association history trace from their syslog trace[81], and for USC trace the logs are collected from the switch (i.e., the switch creates a log when a MN associates/disassociates with one of the APs connected to one of the switch ports). For the UF trace, the logs are collected at both the AP level (for user associations to APs) and the authentication server level (for user sign-in and sign-out). It is generally accepted that the event-based approach provides more accurate records of MN association with the APs in the network. However, there is no in-depth study to quantify the differences between these two approaches. In order to further understand the effects of different methods of trace collection on the user behavior metrics obtained from the traces, we also create an *emulated polling trace* as follows: For an event-based trace, we observe the trace at regular time intervals and emulate what would be recorded if the trace were taken by polling-based method. We then process the emulated polling trace as we do to a normal polling-based trace, and compare the findings with the original event-based trace. We use the April 2004 Dartmouth trace (*Dart-04*) to carry out this experiment, obtaining

Dart-cons and *Dart-rel* traces based on the conservative and relaxed assumptions detailed below.

For traces collected using polling-based methods, we obtain only "sample points" of MN association at regular time intervals in the trace, hence the duration of association must be derived from these samples. Here, an important assumption must be made about the association duration for each observed association sample. We test two different assumptions in this respect: (a) A conservative (*MIT-cons, Dart-cons*) approach, in which a MN is assumed to remain associated with the AP only until the next expected polling epoch (i.e., the expected time instant when the AP is polled again to record the associated MNs), unless indicated otherwise by new samples in the trace (i.e., if the MN is discovered associated with the same AP again at the next polling epoch, its association with the AP is "renewed" for the length of another polling interval). This approach reflects what is observed from the trace faithfully, but may have the drawback that inaccuracy in polling intervals or lost SNMP records (since SNMP uses UDP as the transport layer protocol) will lead to the conclusion that the MN has disassociated from the AP and later associates with the AP again, while in fact the MN has remained associated. (b) A more relaxed approach (*MIT-rel, Dart-rel*), in which a MN is assumed associated with the AP for four polling intervals after it is observed associated with the AP, unless indicated otherwise by the trace (i.e., if the trace reports the MN associates with another AP, the previous association with the old AP terminates before the assumed length of four polling intervals). This approach is more robust to imperfections (e.g., packet losses, wireless channel variation) in the trace collection process, however, it may erroneously increase the duration of association with APs after a MN in fact disassociates from the AP. The polling interval for the MIT trace is 5 minutes, and we use the same polling interval to obtain the samples for the emulated polling traces from the Dartmouth trace (described in the last paragraph). Hence the conservative and relaxed approaches assume a MN remains associated with an AP for 5 and 20 minutes, if a sample indicates the MN is with the

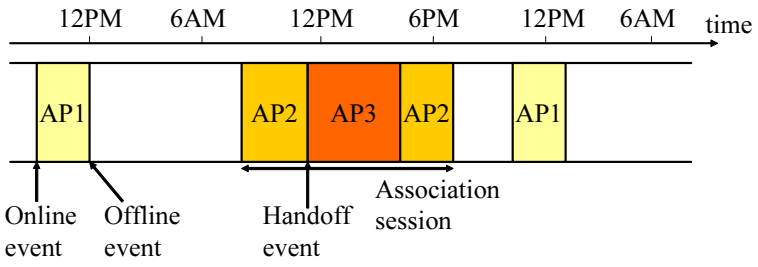


Figure 3-1. Illustration of the term definitions.

AP, respectively. For the UCSD trace the polling interval is 20 seconds. We use only the relaxed approach to process the UCSD trace.

3.3 Definition of Terms

In this section we first introduce some terminologies (refer to Fig. 3-1) we will use in the subsequent chapters. We use the notion **online** (or in short, **on**) to refer to the state when a MN is associated with any AP in the network (or equivalently, its current location is known, or it is "present" at the moment). On the contrary, the notion **offline** (or in short, **off**) refers to the state of a MN being absent at the moment (i.e., it is not associated with any AP currently). Related to the above definitions, an **online event** is defined as a MN starting a new association to any AP from the off state. An **offline event** is defined as a MN disassociating itself from the current AP and changing to off state (i.e., it does not roam to other APs, but goes offline directly). A **handoff event** or a **roaming event** is defined as a MN changing its association from one AP to another with no offline time in between. An **association session** is defined as the duration between an online event to the next offline event. There can be many handoff events within one association session. The **total online time** is the sum of the lengths of association sessions (i.e. the sum of the "shaded" intervals in Fig. 3-1), and the **existence time** is the time difference between a MN's first online event and its last offline event in the studied trace. We use the existence time as a conservative measure of the time duration for which the MN is a potential user of the network, since the MNs are not always online and the user population can change with respect to time on university campuses (for

example, there can be visitors who only use the WLAN on campus temporarily during the visit). Before a MN first shows up and after it last disassociates, we assume that it is not part of the network.

While the WLAN traces provide approximate location information for individual users, we leverage the information from these traces to further understand the potential opportunities of communication between the devices, when the infrastructure (i.e., the access points) is not present. We derive the **encounter events** (we sometimes refer to these events as **encounters** in short) between the MNs from the WLAN traces using the following assumption: The MNs can communicate with each other if they are associated with the same AP (or the same switch port in the USC trace). Following this assumption, the duration of the encounter events can be derived from the overlapped time intervals of association sessions of different MNs with the same AP. Nodal encounters in mobile network are important events as they provide opportunities for involved nodes to build up some relationship or to communicate directly. We acknowledge this assumption may be not completely accurate, as there can be scenarios that (i) MNs are associated with the same AP but still too far apart to communicate directly, (ii) MNs are able to communicate while associated with different APs, or (iii) MNs may encounter each other outside the coverage of any AP, hence some encounter events cannot be reconstructed from the WLAN association traces. However, we believe that the encounter events derived from WLAN traces capture a large portion of MNs within direct communication range under current deployments of WLAN (usually, the network administrators seek to provide ubiquitous coverage when possible). In addition, the WLAN traces capture another important factor one needs to incorporate when considering the inter-device communication opportunities: the usage pattern of the devices. According to our findings from the traces (and the understanding of how people use these devices), many devices are not *always on*. The empirical approach of deriving encounter events from the WLAN traces has the benefit of including such on-off usage patterns in the analysis of encounters

(and the resulting protocol performances on top of these communication opportunities). While omnipresent wireless mobile personal communication devices are envisioned to emerge in near future, it is still not clear how they are going to be used. Starting with the usage pattern of today's mobile devices is a reasonable starting point to investigate this futuristic scenario. Also, since the WLAN traces are the largest traces available today, such derived encounter traces provide a good opportunity to understand the communication opportunities in *current* large-scale wireless networks used by the generic public, compensating the small-scale encounter traces collected from special-purpose experiments (as in [27, 29, 30, 32, 33]).

3.4 Detailed Descriptions of Our Traces

In this section we describe the details of our efforts for trace collection from the University of Southern California (USC) and the University of Florida (UF) WLANs. We list the information we collect from these two networks.

The effort of trace collection from the USC campus WLAN started from December 2003. In the first stage, only the online events of the users were collected. We have this information from December 23, 2003 to December 13, 2004. However, this information is not sufficient to provide complete knowledge of user locations (in particular, we cannot identify when users leave the associated APs). Therefore, in the second stage started on April 20, 2005 (it is still an on-going effort), we collect both the online and offline events. On the USC campus the trace is collected from the network switches. The switch logs the start and end events of user associations with the APs connected to the switch. The locations of MNs are represented by the switch ports their associated APs connect to. Hence, the location granularity is per switch port (which corresponds to approximately a building or several buildings in geographic vicinity on the USC campus). In addition to the association events, the switch also logs netflow information (i.e., the source/destination IP/port and the protocols used in each traffic flow, with the size of traffic sent/received in the flow) of each user. In the USC trace, we identify the users by the MAC addresses,

assuming that each MAC address is a unique device, and each unique device corresponds to a unique user.

On the UF campus the trace is collected from several different network components. The WLAN users on the UF campus have to sign in to authentication servers before they can use the network. We keep the log from the authentication servers, which includes the DHCP IP assignments made by the server (these authentication servers are also DHCP servers at the same time), the start and end events of user authentication sessions. When a user finishes an authenticated session, the authentication server also reports the duration and the amount of traffic sent/received in the whole authenticated session. In addition to the authentication servers, we also obtain the association logs from the APs on campus (i.e., the association, disassociation, and roaming events), to keep track of the locations of the users. In the UF trace, a user can be identified in two different ways. Since both the authentication servers and the APs keep the logs by the MAC addresses, we can consider each unique MAC as a unique user. In addition to this typical assumption, since the authentication server logs also keep the UF user name (which is an unique identifier among UF network users), it is possible to identify the actual identity of a user even if the user uses several different devices. Thanks to this additional information, it is possible to further track the same person across multiple devices (although through this dissertation our study is based on the assumption that each unique MAC address is a unique user). This capability potentially allows us to go beyond device level and study individual users directly. However, there is also a potential danger of compromising the privacy of the users. Privacy-preserving processing techniques (e.g., to anonymize the traces so that it is still useful, but one cannot link the identities in the trace to real-life identities) are out of the scope of this dissertation, but is a current research agenda in our research group.

CHAPTER 4
CASE STUDY I: MODELING INDIVIDUAL USER MOBILITY

In this chapter we present the first case study in the dissertation. This first case study deals with individual users in the trace as independent entities, analyzing their microscopic behavior in terms of the association to the WLANs and mobility within this infrastructure. We first observe the mobility characteristics of individual users from multiple traces, comparing their similarity and differences, in section 4.1. We identify several major mobility characteristics, such as the *nodes not being always on*, the *skewed location visiting preferences* and the *periodical re-appearance* of nodes at the same location, as prominent mobility characteristics from the traces we study. Then, we propose a mobility model in section 4.2, which is flexible to capture the spatial and temporal dependency of the prominent mobility characteristics we observe empirically from the traces. In addition, this model is also mathematically tractable, hence it facilitates theoretical analysis of protocols in mobile networks.

4.1 On Modeling User Associations in Wireless LAN Traces

4.1.1 Introduction

Recently, wireless networks have been deployed ubiquitously in various environments, especially in university campuses and corporations, and gained popularity rapidly. With more users switching to wireless networks, the importance of understanding user behavior in such environments is becoming clearer. From the vast amount of wireless LAN (WLAN) traces available to the research community, one can obtain important and fundamental knowledge about its users. Among the vast space for potential investigation, we focus on the following question: How do we realistically model user[1] behavior and usage in campus WLANs? More specifically, if we are interested in modeling the mobility patterns of

[1] In this dissertation we use the terms *user*, *node*, and *mobile node (MN)* interchangeably. We assume that one MAC address in the trace corresponds to a unique device (MN), and a MN is always tied to the same unique user.

individual users in such environments, what characteristics are important to observe from the traces? And, how do users in different environments differ (or not) on these aspects? We seek to answer these questions by an extensive study of WLAN traces.

In this section we gain further understanding of realistic user behavior (e.g. usage and mobility) utilizing the most extensive wireless LAN traces collected to date from three university campuses (USC, Dartmouth, UCSD) and one corporate network. Such an understanding is important for several reasons: Most importantly, trace analysis is a necessary first step towards developing realistic mobility models that are crucial for the design, simulation and evaluation of wireless networking protocols. We will follow up on this task in section 4.2. Additionally, analysis of user behavior and network usage patterns enables accurate assessment of wireless network utilization and aids the development of better management techniques and capacity planning decisions. As new technologies evolve (e.g., variants of 802.11 WLANs, or ad hoc networks), fundamental understanding of user behavior becomes essential for the successful deployment of such emerging technologies.

Several studies have been previously conducted on the analysis of WLAN traces [10], [11], [13], and we borrow from these traces and studies. These studies are quite helpful, but each of them is based on a single campus with a different focus, and hence it becomes unclear whether their findings generalize beyond the studied campus. In our study, we go beyond previous work to compare user behavior across different traces, and try to observe the general trends and quantify the detailed differences among them. We look into the aspects that we consider important to model user behavior in WLANs, and reason about the commonalities and differences of these aspects between campuses. For the metrics we study, we find that in general most of the campuses follow similar trends, such as (1) Most nodes display on-off usage pattern. They are offline for non-negligible amount of time, and switch between online and offline states often. This fact is largely overlooked by previous researches on modeling WLAN users although it is an omnipresent phenomenon from all

traces. (2) Most nodes visit only a small portion of the access points (APs) on campus. Therefore, preference in user association is another important aspect to model users of WLANs. The above findings may be intuitive, but it is surprising to observe that the on-off pattern of users change significantly as the popularity of WLAN increases through years, but the ratio of visited APs hardly changes. In other words, there are varying and invariant user characteristics as one technology gains its popularity. (3) In most cases we identify repetitive patterns in user association over various time frames (e.g., days, weeks). Users re-appear at the same AP it previously associated with higher probabilities after time gaps of integer multiples of days. We propose the *network similarity index (NSI)* as a quantitative metric to capture such repetitive pattern. These findings point to unrealistic assumptions often made in user modeling (for both usage model and mobility model) and simulation, as the findings from traces are significantly different from the general assumptions (e.g., always-on users with no preferences in their association patterns). We later leverage the findings as guidelines for a realistic mobility model in section 4.2.

As one expects, the details of these user behavior metrics depend on the underlying campus environment and user device type; we will comment on the findings throughout the section. In addition to that, in this work we also compare two different trace collection methodologies, polling-based (e.g. SNMP) and event-based (e.g. syslog). We show the difference between these two trace collection methods by generating an *emulated* SNMP trace in post-processing from syslog traces (which have better time resolution than SNMP traces), and compare the differences among the two traces. Sometimes, major differences can be attributed to different trace collection methods used. This suggests the need for a standard methodology for trace collection to make data from different environments comparable.

4.1.2 Analysis of Individual User Behavior

In this section we propose metrics to describe and compare behaviors of individual users in the studied environments. These metrics correspond to different aspects of MN

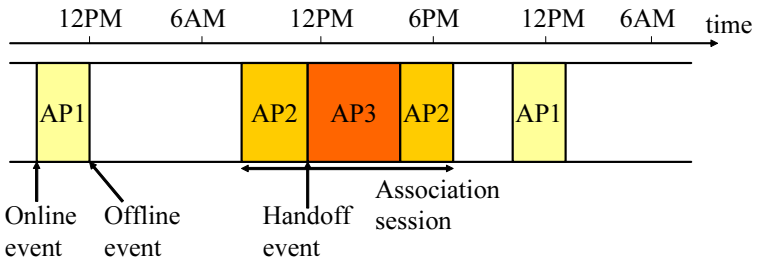


Figure 4-1. Illustration of a MN's association pattern with respect to time of the day.

association behaviors in a WLAN. We shall use Fig. 4-1 to illustrate. One could see the association pattern of a MN as a sequence of associated APs (shown by shades in Fig. 4-1), potentially with time segments during which the MN is *offline* (e.g. not associated with any AP) between associations. We look into four major categories to understand user behavior as follows:

(a) *Activeness of users*: This category captures the tendency of a user to be online (i.e., How actively the user shows up in WLAN?). In general wireless network users are not always on, but show up in the trace intermittently, as opposed to the always-on nodes assumed in the synthetic models.

(b) *Macro-level mobility of users*: This category captures how widely a MN moves in the network in the long run (i.e., for the whole trace duration), and how its online time is distributed among the APs. The intention is to capture overall long-run statistics and preference of a MN visiting APs. (i.e., How are the shades distributed in Fig. 4-1? Do we need many different intensities of shades for each user as it associates with many APs? Can we find a few "dominant" APs for each MN?)

(c) *Micro-level mobility of users*: This category captures how MNs move in the network while it remains associated with some AP (i.e., handoff). The intention here is to capture the mobility of a MN while *using* the wireless network, a different objective from the macro-level mobility. (i.e., How often does the MN change associations without leaving the network?)

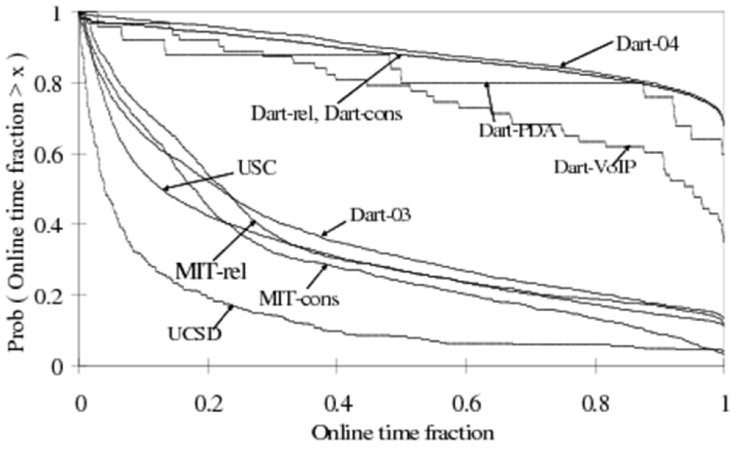


Figure 4-2. CCDF of online time fraction

(d) *Repetitive association pattern of users*: This category captures the user association behavior with respect to time. We expect users to show repetitive structure in association patterns during similar times of different days, as their mobility patterns are dictated by their daily schedule. This idea is also illustrated in Fig. 4-1: the user appears at AP1 during late evenings in both days. We propose the *network similarity index (NSI)* as a metric to quantify the tendency of users to show repetitive patterns in their associations.

4.1.2.1 Activeness of the users

Activeness of users is the first aspect we look into in attempt to compare the different traces. Activeness of users can be captured by either total online time fraction of a MN or the number of association sessions generated by a MN.

We choose to define the *online time fraction* as the ratio between MN's total online time to its *existence time*[2] , and plot the CCDF[3] of online time fraction of users in

[2] Note that, following this definition, MNs that associate with the APs for only one session have online time fraction of 1.0. This definition tend to over-estimate user activeness for one-time users.

[3] CCDF, or the complimentary cumulative distribution function, is the probability for a random variable to exceed a given quantity x. It is a non-increasing function taking values between the range $[0, 1]$.

various traces in Fig. 4-2. From Fig. 4-2 we observe that in all traces only a small portion of users are always on even though by definition the user activeness is already over-estimated, except for the Dart-04 trace. The average online time fraction is 87.68% for Dart-04 trace, and between 36.44% (Dart-03) and 14.12% (UCSD) for other traces. The standard deviation for online time fraction is large, varying from 0.24 to 0.36 for all traces. These observations argue strongly that **users have on-off usage patterns, where some of the users are heavy users (with high on time) while many are light users**. The distributions of the *on/off* times seem to depend heavily on the environments (i.e., campus) and the device types in the traces. UCSD trace, which focused only on PDA users, is the least active one among all traces. The other traces (MIT, USC, Dart-03) are not very different in online time fraction distribution. The activeness of MNs increase significantly from 2003 to 2004 in Dartmouth trace, which agrees with the findings in [13]. By comparing the curves of Dart-04, Dart-rel, and Dart-cons, we observe that **online time fraction is consistent for the same trace under different trace collection (or trace reconstruction) methods**. Comparison between Dart-04 with Dart-PDA and Dart-VoIP shows that during the same trace period, the handheld devices are less active than the average of the total population. However, handheld devices in the Dartmouth trace are much more active than the UCSD trace, but the reason is not clear at this point and warrants further investigation.

We also check whether the significantly higher online time fraction in Dart-04 trace is caused by users with only one short association session (hence its online time fraction is over-estimated by our definition). It turns out that the high online time fraction in Dart-04 trace is caused by significant increase of always-on users. In Dart-04 trace, there are 27.5% of users that initiate only one association session which lasts for the duration of 30 days, the whole trace period. The same number for Dart-03 trace is less than 0.04%. There are two possible reasons for the very different behavior in the two time periods. (1) July 2003 was during summer vacation, hence the activity was significantly lower,

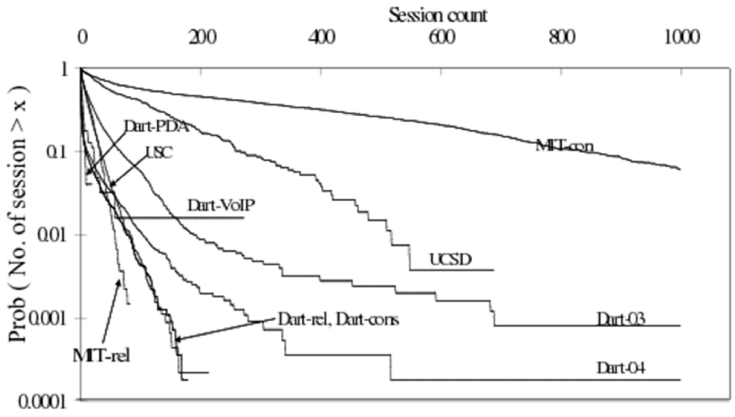


Figure 4-3. CCDF of number of association sessions by users

or (2) The way people use WLAN has changed between these two trace period at the Dartmouth College. Users in Dart-04 trace tend to use wireless LAN as a replacement for wired network, and keep their device associated with WLAN, instead of establishing the connection only when it is needed. If the later speculation is true, as we see this paradigm shift from using WLAN as a temporary connection to an always-on, permanent connection, it is possible that the online time fraction will also increase significantly for other deployments.

We further compare the CCDF of the number of association sessions generated by users in these traces in Fig. 4-3. We observe that the **PDA users in the UCSD trace generate more association sessions than users in other traces** (except MIT-con trace, explained below), which include generic wireless network devices (mainly laptop users) during comparable trace duration. This fact, together with the less online time fraction in Fig. 4-2, indicates that the UCSD PDA users are more likely to use the devices for shorter but more frequent sessions. However, this observation does not apply to the handheld devices in Dartmouth. Both PDAs and VoIP devices initiate less sessions than the general devices in Dartmouth. From the figure we also observe that count of association sessions is sensitive to the trace collection method. The emulated

polling traces (Dart-rel and Dart-cons) show very different distributions form the original Dart-04 trace, since **traces collected by polling at regular intervals will overlook association sessions shorter than the polling interval**. Comparing the CCDF curve of Dart-04 to Dart-rel or Dart-cons in Fig. 4-3, we see that the emulated polling traces observe only one fifth of sessions for the MN with the largest number of sessions (200 versus 1000). Another technical difficulty here is to adequately translate a sample seen in the polling-based traces to the duration of association appropriately, as we find the curves of MIT-cons and MIT-rel drastically different. A closer investigation into the MIT trace reveals that although SNMP polling intervals are typically 5 minutes, sometimes records of MN association are obtained at longer intervals, leading to bogus terminations and re-initiation of association sessions if the conservative assumption is used and hence the high association session counts shown by the curve MIT-cons.

4.1.2.2 Macro-level mobility of users

In this section we capture the long-term mobility of users by obtaining the overall statistics of AP association history during the whole trace period. We investigate the number of APs a user associates with and the fraction of online time it associates with each of the APs. The purpose of this section to understand the preference of MN association at the access point level. Note that the observation could not directly translate to the preference of user visits at geographic level, as APs are not uniformly deployed on the campuses. For example, popular locations on campus may have multiple APs deployed in anticipation of high usage, hence artificially reduce the load observed for each of these APs in the trace. Nevertheless, we can have some idea about how widely a MN visit (in terms of number of visited APs) from this section. If a user visits more APs, and stays at more APs with non-negligible fraction of its online time, it is an indication that the user visits wider range on the campus (i.e., more *mobile* in the long run) than another user who visits few APs and spend most of its online time at one or two APs.

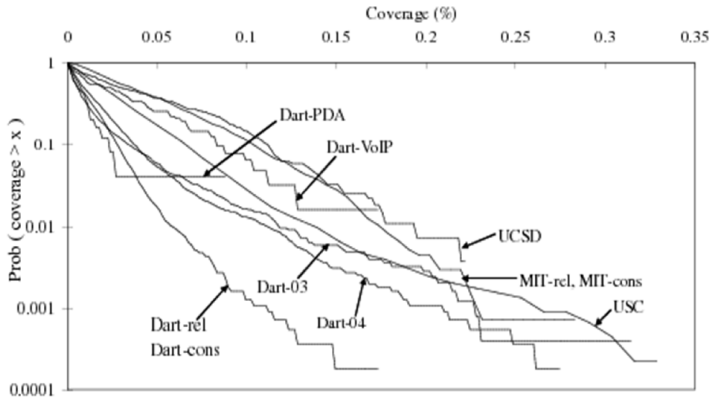


Figure 4-4. CCDF of coverage of users.

We define the *coverage* of a user as the *percentage* of APs on the campus the user associates with during the trace period. For the USC trace we use switch ports in place of APs. The distributions of the coverage of users in the traces are shown in Fig. 4-4. This metric captures how widely a user moves for the whole period of trace in the studied network environments.

We observe that **users have small coverage in all environments**. The average coverage is between 4.52% (UCSD) and 1.10% (Dart-cons/rel). None of these traces has even a single user visiting more than 35% of all APs. In the UCSD trace, the PDA users seem likely to visit a larger portion of campus than the generic users do in the other campus-wide traces, due to the portability of PDAs. Similar observation applies to the VoIP devices in the Dartmouth trace, which is the most mobile sub-user group in the Dartmouth trace. However, PDAs in the Dartmouth trace are less mobile than the generic users in the period we studied. We suspect that the result may be influenced by a few extreme users (there are only 25 PDA users identified during this period, and half of them visit only 4 or less APs). The MIT trace is collected from only three buildings, hence the relative coverage of users is a bit higher. It is important to note that the **coverage seems to remain stable with respect to time change**, although the activeness of users changes significantly (compare Dart-03 and Dart-04). The **coverage is sensitive**

to the trace collection method since the polling-based method overlooks short sessions and **under-estimates** the coverage metric. However, different re-construction methods of the polling-based trace (conservative or relaxed approaches) result in the same coverage, as the metric counts the number of APs a MN associates with, not the association duration.

We further study the average percentage of online time a user spends with every AP it visits. We order the APs a user ever visits during the trace period by the user's total association time with each AP, and average across users to get the average percentage of online time a user spends with its most visited AP to least visited AP. These results are shown in Fig. 4-5. From the figure we observe that for all environments, the general trend is that **each user has very few APs at which it spends most of its online time**. In particular, for all the traces, a MN spends on average more than 65% of its online time with *one* AP, and more than 95% of online time at as few as the top-5 APs combined. **The left-end of the curves are similar, but the tails vary**. The higher mobility of the UCSD PDA users translates into a longer tail, where in addition to those few most visited APs, the users also access the wireless network at much more locations with a small fraction of the user's online time as compared to other traces. Similar observations apply to Dart-VoIP and Dart-PDA traces. It is interesting why Dart-PDA trace shows small coverage in Fig. 4-4 but high average fraction of time associated with less popular APs here. These two points, however, do not contradict each other. A closer investigation reveals that although there are a small fraction of widely-visited PDAs (from Fig. 4-4), those who visit many APs contribute more of their online time to less popular APs. This metric is **robust** to different trace collection methods and assumptions of trace post-processing, as the curve for Dart-04 is close to Dart-cons or Dart-rel. Similar observations are made for the MIT trace.

4.1.2.3 Micro-level mobility of users

In this section we study the per-association session mobility of a user, which reflects its short-term mobility. This captures a different dimension of user mobility as compared

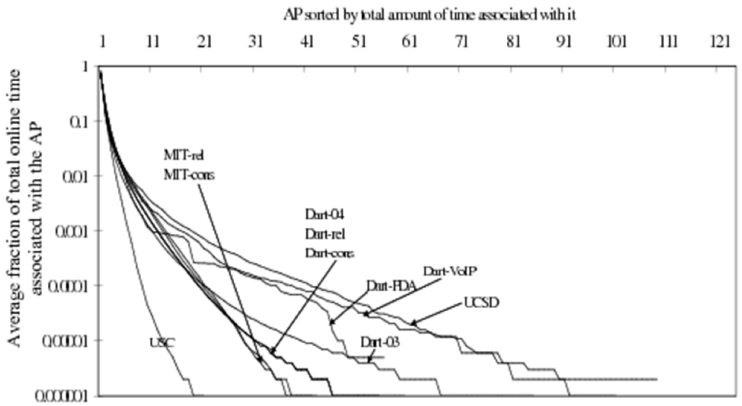


Figure 4-5. Average fraction of time a MN associated with APs. For each MN, the AP list is sorted based on association time before taking the average across users.

to the previous section: How mobile the user is while *using* the network. We use handoff statistics as a measure of user mobility while using the network. However, after the investigation of the handoff statistics, we discover a lot of handoff events are due to so-called "ping-pong effect" rather than real movements. The term "ping-pong effect" refers to the phenomenon of excessive handoff events due to disturbance in wireless channels while the MN itself might be stationary. Hence, we cannot directly link the handoff statistics to the micro-level mobility of the users. Development of better filters for ping-pong effects is needed before we can really understand the micro-level mobility form the WLAN traces.

First we show the CCDF curves for the total handoff event count during the whole trace period in Fig. 4-6. Our first intuition is that user mobility should be dependent on the device type, and handheld devices should display higher mobility than users in other traces. This is true for the Dart-VoIP trace, as the VoIP devices have the most per-user handoff count among all traces. However, the PDAs in both UCSD and Dartmouth trace do not have more handoff events than other traces. For the UCSD trace, this may be related to the fact that the PDAs are usually used for short sessions, hence they experience less handoff events. For the Dart-PDA trace, some of the PDAs are online for

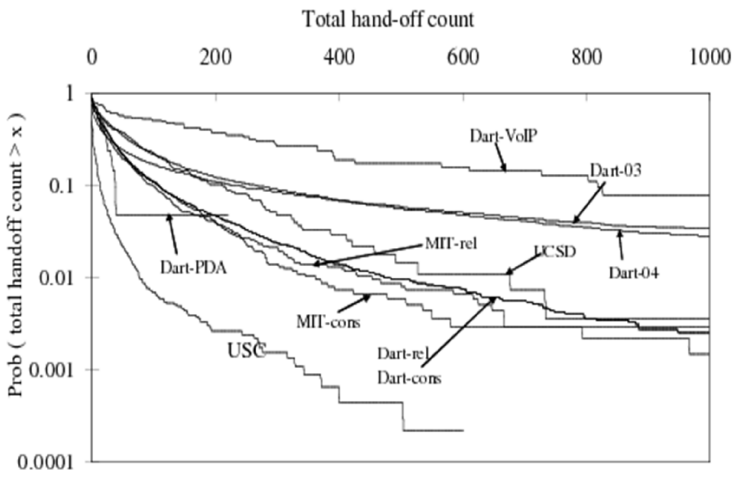


Figure 4-6. CCDF of total handoff count per MN.

long durations, but they do not have many handoff events. The reason is not clear at this point.

From Fig. 4-6 we observe that the exact number of handoff count depends heavily on the network environment (e.g., the deployment of the APs, etc). In the USC trace, the coarse location granularity directly leads to the lower handoff counts. On the other hand, the Dartmouth traces have much more handoff events than other traces. We also observe that the handoff counts in Fig. 4-6 are sensitive to the trace collection method, as the curve for Dart-04 differs significantly from Dart-rel and Dart-cons. This is again because the polling-based method overlooks quick changes of user associations between polling intervals and hence many handoff events are not captured. In addition to the above, we also observe that for all the traces, handoff counts vary significantly among the users - There are some users with many handoff events and some with few.

To better understand the cause of handoff events, we look into the relationship between session lengths and handoff events in the session for each trace. As an example, we show a scatter plot for session lengths (in minutes) and handoff counts for all sessions in the USC trace in Fig. 4-7. From the graph, we see that there is no clear trend between the session lengths and the handoff counts. In some cases, we see extremely long sessions

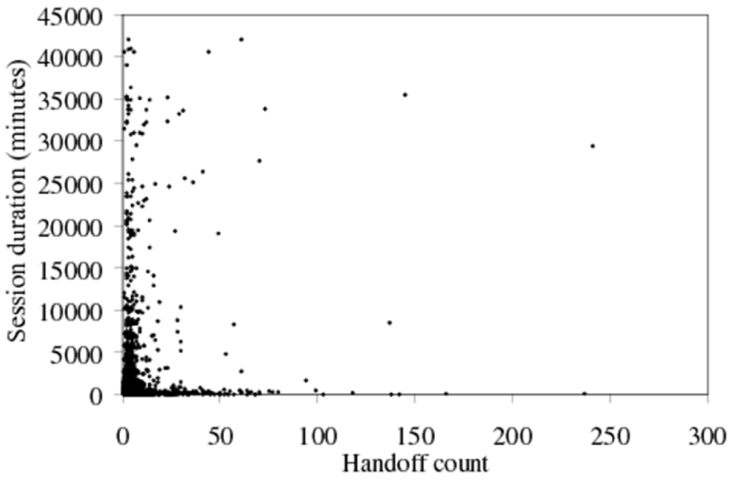


Figure 4-7. Scatter plot: Session durations versus handoff count in the session for the USC trace.

without any handoff events, or extremely many handoff events in a session with short duration. The correlation coefficients between session lengths and handoff counts for all the studied traces are between 0.377 and 0.030. So we can see that the session length and the handoff count have a weak linear correlation to each other in all traces.

We further look into the following statistics to observe whether the sessions with high handoff counts are all from a small set of extremely mobile users: For each user, we calculate the average handoff event per unit time (i.e. the *handoff rate*) for each of its sessions, and then calculate the mean and variance for the user's handoff rate from all the sessions the user initiates. If a high degree of mobility leading to the high handoff count is an intrinsic property for some users, we should see that those users show high average and low variance in their handoff rates. We use the coefficient of variation (the standard deviation divided by the mean) to understand the degree of variation in the handoff rates for users. In Fig. 4-8, we show the CDF of the coefficient of variation of the handoff rate for the studied traces. Only the users with more than one session and one handoff event are considered in the graph, since users with only one session automatically result in 0 variance for its handoff rate. From the figure, we see that **the handoff rate displays high variance for most of the users**. In all traces, more than 60% of users have its

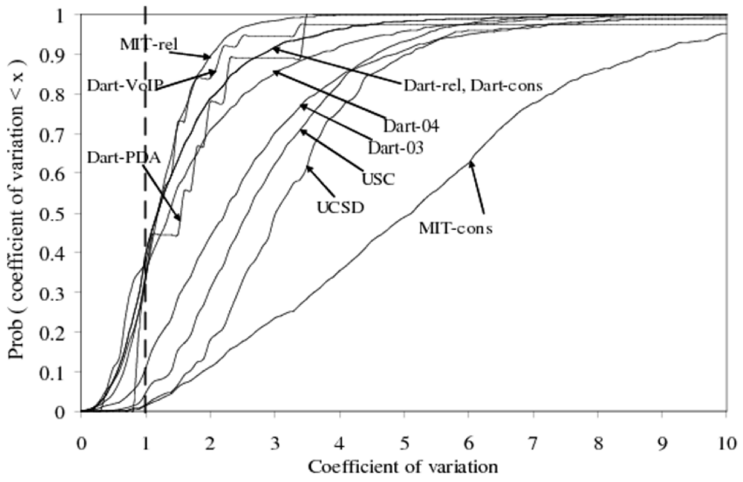


Figure 4-8. CDF for the coefficient of variation of the handoff rate of the users. Note that for all traces, coefficient of variation is larger than 1.0 for at least 60% of MNs with more than one session.

coefficient of variation of the handoff rate larger than 1.0 (i.e., the standard deviation being larger than the mean). This indicates even for a given MN, handoff rate varies drastically from session to session.

Combining the observations in the preceding paragraphs, we conclude that handoff events not only distribute unevenly between users, but also happen unevenly between the sessions for the same user. This indicates that the handoff events are greatly influenced by the environmental condition when a session is established rather than the property of the MN who initiates the session. We even observe that some MNs have hundreds, sometimes even thousands, handoff events between less than 5 APs within a session. Such a scenario is much more likely due to ping-pong effect rather than true user mobility. The reduction of ping-pong effect is an important issue to make better interpretation about the micro-level user mobility from the WLAN traces and warrants further study.

4.1.2.4 The repetitive association pattern of users

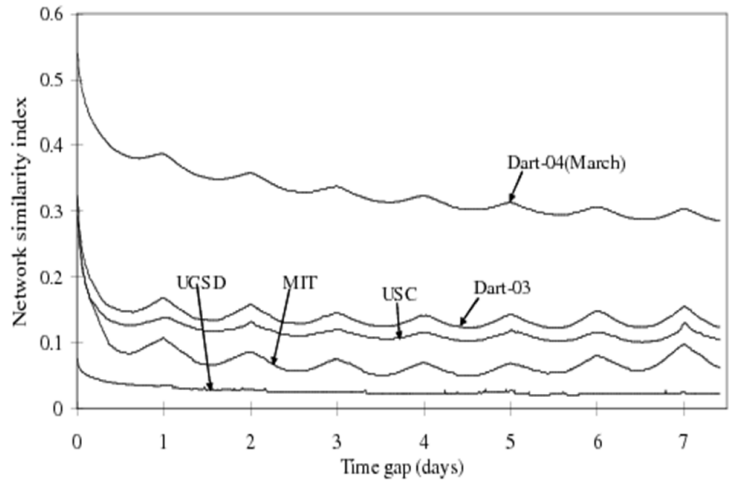
Naturally, user behavior changes with respect to time of the day and day of the week, as people follow daily and weekly schedules in their lives. In some cases, the user association pattern repeats itself day to day or week to week. In this section we try to

quantify such repetitive pattern by defining the *network similarity index (NSI)* below. We try to find the tendency of users displaying periodical association behavior by calculating the NSI of the traces.

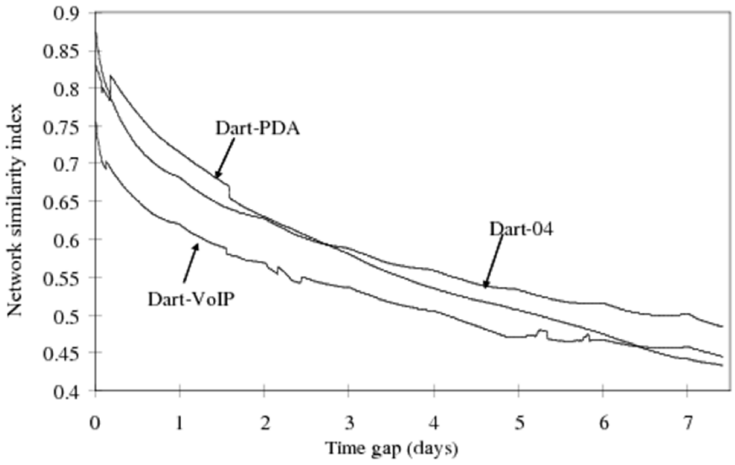
We start the definition with *location similarity index* for individual users. First we take snapshots of associated APs of the user every minute. To study the tendency of the user showing repetitive behavior after a certain time gap (e.g., every 24 hours), we consider all snapshot pairs that are separated by this time gap, and calculate the fraction of all such pairs where the user associates with the same AP in both snapshots. This is an indication of how likely this user re-appears at the same location after the chosen time gap. *Network similarity index (NSI)* for a given time gap is the average of *location similarity index* of all users for this time gap. Hence, NSI represents, *in average sense*, how likely would a node associates with the same AP after the given time gap for the trace under discussion.

In Fig. 4-9 we show the *NSI* for all the traces. To see the details better, we split the figure into two parts: curves with smaller absolute NSI values are shown in Fig. 4-9(a), and curves with bigger absolute NSI values are shown in Fig. 4-9(b). We will discuss the physical meaning of the absolute value of NSI later in this section.

From Fig. 4-9(a), we see that **in most of these traces** (i.e., USC, MIT, Dart-03) **we observe noticeably higher network similarity index if the time gap is close to integer multiples of a day**. This is an indication that **users have the strongest tendency to show repetitive association pattern at the same time of each day**. It is also interesting to observe that for these traces, the **network similarity index for the gap of 7 days (i.e., a week) is the second highest**, only slightly lower than that for the gap of 1 day. This indicates **weekly repetitive pattern is also strong** in these traces. On the other hand, the UCSD trace shows little repetitive pattern as there is almost no obvious spikes in its *NSI* curve. This can be attributed to its user population being PDA users. Unlike laptops, which are more related to work, PDAs are usually used



(a)



(b)

Figure 4-9. Network similarity indexes. The peaks represent intervals for which there is high similarity. (a) NSI curves with smaller absolute values (less always-on, stationary users), (b) NSI curves with larger absolute values (more always-on, stationary users).

in a more casual way in short, scattered durations. Hence it is expected that PDA users show less repetitiveness in their usage pattern.

However, we see that in Fig. 4-9(b), the NSI curves for the Dart-04 trace or its sub-groups of users[4] do not show strong patterns of periodicity as discussed above. We suspect that the periodical association behavior in the Dart-04 trace is hidden (only minor fluctuation is visible closer to integer multiple of days) due to the increase of always-on users (cf. section 4.1.2.1). In the 2004 trace, we have more always-on, stationary users using WLAN as a replacement of wired networks. This is reflected by the higher average value of the *NSI* curves, indicating larger fraction of users always stay at the same location. This may be attributed to the fact that Dartmouth traces include users in student dormitories, which are mainly stationary users and have contributed to high location similarity indexes. We further compare the NSI curve of the Dart-04 trace in Fig. 4-9(b) to the NSI curve of Dart-04-March (only used in this experiment) in Fig. 4-9(a). For the Dartmouth College, the month of March contains the spring break, when some of the stationary users in dorms are absent, and we see that the periodicity of association behavior is more visible in the March trace. From the above experiment, we argue that the periodic behavior in the average NSI curve comes from non-stationary users (e.g., those who come to work or classes during day time and follow a regular schedule), not the stationary users who use WLANs as a replacement of wired LANs. This point is partly supported by the findings in [36]: Most users displaying periodicity in association have home locations at academic buildings. The USC has not deployed WLAN in dormitories yet (for the one-moth trace we have chosen for this study), and the MIT trace is mainly focused on buildings for work. That may be the reason why periodic association behaviors are more obvious in those traces.

[4] Curves for Dart-cons and Dart-rel are not shown to make the graph more readable. They are not very different from the Dart-04 curve.

4.1.3 Conclusions and Future Work

Our contributions: The major contributions of this study are the following: First, by using WLAN traces from four different sources, comparing the results and highlighting both similarities and differences, it is the largest scale trace-based study in the literature as we are aware of. Although some of the findings in the study match with simple intuition of user behaviors, by extensive investigation we are able to further quantify and show the minor differences in detail systematically, and reason about the cause of those differences (e.g., methodology of trace collection, user population, network environment, time of trace collection, etc.). Second, by proposing metrics for describing individual MN behaviors, we propose a basis on which mobility models for individual MNs can be established. We also find several facts indicating that conventional, randomly generated synthetic mobility models (such as random waypoint, random walk, etc.) are not adequate for a heterogeneous environment such as university campuses and corporations. This work extends previous works [10], [13], [11] on analyses of WLAN traces by considering traces from multiple campuses and multiple aspects to model user behavior. We also make our own WLAN traces collected at USC campus available at [1], together with many pointers to existing WLAN trace archives.

To summarize, the findings from the traces point out important common features in all studied environments. Wireless network users in university campuses and corporate network are characterized by (1) limited number of visited APs in the network and a large proportion of online time spent at very few of its most visited APs. The coverage of users never exceeds 35% in all traces, and users spend more than 95% of their online time with as few as *five* APs. Furthermore, these numbers seem to remain relatively stable in a given environment, even if the WLAN gains popularity and users become more active. (2) Periodic association patterns with strong daily/weekly pattern. We believe that these metrics capture important characteristics about users in wireless networks that are largely overlooked by earlier work on mobility modeling and wireless network simulation. (3)

Large percentages of offline time. Except for the Dart-04 trace, there are less than 20% of users that are always on, and more than 68% of users are offline more than 50% of time. Even in the most active Dart-04 trace, there are more than 30% of users not always on. We will continue on to the development of a realistic *time-variant community mobility model* based on these characteristics in the next section.

By comparing the traces collected by event-based logging method and the emulated polling-based traces for the same environment, we find that they sometimes show dissimilar results. Hence, although polling-based trace collection is suitable for usage statistics, they are not very suitable for deriving the association patterns of users, as they tend to overlook details of association changes. Also, we need better heuristics to remove the ping-pong effects to make better interpretation about micro-level mobility events (i.e. handoff) from the traces.

Finally, the statistics obtained using the fore-mentioned metrics can be considered as characteristics or "fingerprint" for particular environments or user population. It should be interesting to develop mechanisms to inspect these "fingerprints" and argue about similarity/dissimilarity between environments.

4.2 Modeling Spatial and Temporal Dependencies of User Mobility in Wireless Mobile Networks

4.2.1 Introduction

In the *mobile ad hoc networks* (MANETs) [3], as the devices are usually easily portable and the scenarios of deployment are inherently dynamic, *mobility* becomes one of its key characteristics. It has been shown that mobility impacts MANETs in multiple ways, such as network capacity [48], routing performance [20], and cluster maintenance [79]. In short, the evaluation of protocols and services for MANETs seems to be inseparable from the underlying mobility models. It is, thus, of crucial importance to have suitable mobility models as the foundation for the study of ad hoc networks.

Our main contribution in this section is the proposal of a *time-variant community mobility model*, referred to as the TVC model, which is *realistic, flexible, and mathematically tractable*. The model captures several important mobility characteristics we, and other researchers, have observed empirically from various WLAN traces [66]. As we show in the previous section, one of the salient characteristics is *location preference*. In the TVC model, we extend the concept of communities from [52] to serve as popular locations for the nodes. Another important characteristic is the *time-dependent, periodical behavior* of many nodes. To capture this, we implement time periods in which the nodes move differently [67]. To our best knowledge, this is the first *synthetic mobility model* that captures non-homogeneous behavior in both *space* and *time*.

In addition to the improved realism, the TVC model can be mathematically treated to derive analytical expressions for important quantities of interest, such as the *nodal spatial distribution*, the *average node degree*, the *hitting time* (time required for a mobile node to hit a randomly selected coordinate) and the *meeting time* (time required for two mobile nodes to come within communication range of each other). These quantities are often fundamental to theoretically study issues such as routing performance, capacity, connectivity, etc. We show that our theoretical derivations are accurate through simulation cases with a wide range of parameter sets, and additionally provide examples of how our theory could be utilized in actual protocol design.

To establish the flexibility of our TVC model we also show that we can match its two prominent properties, *location visiting preferences* and *periodical re-appearance*, with *multiple* WLAN traces, collected from environments such as university campuses [80, 81] and corporate buildings [82]. More interestingly, although we motivate the TVC model with the observations made on WLAN traces, our model is generic enough to have wider applicability. We validate this claim by examples of matching our TVC model with two additional mobility traces: a vehicle mobility trace[17] and a human encounter trace[84]. In the former case, we observe that *location visiting preferences* and *periodical*

re-appearance are also prominent characteristics in vehicular movements. In the later case, we are able to match our TVC model with some other mobility characteristics, namely the inter meeting time and encounter duration between different users/devices. These latter quantities are particularly important for encounter-based (or "delay-tolerant" [4]) protocols. Despite these characteristics are not explicitly incorporated in our model by its construction, they can be still realistically reproduced.

To our best knowledge, this is the first synthetic mobility model proposed that matches measurement sets (traces) collected from multiple scenarios, and has also been theoretically treated to the extent presented here. Due to its strengths in both flexibility and theoretical tractability, the TVC model has two major applications: to generate realistic mobility patterns under a wide range of different scenarios and to facilitate performance analysis and prediction. We also make the code of the TVC model available at [9].

In this section, we first re-iterate the mobility characteristics we discovered from the traces, discuss how we construct a mobility model to capture them, and then formally introduce our *TVC model* in section 4.2.2. Then, in Section 4.2.3, we embark to present our theoretical framework and derive generic expressions of various quantities under the TVC model. The accuracy of these expressions is validated against simulations in Section 4.2.4. Finally we show the two major applications of the TVC model: in Section 4.2.5, we show how to generate realistic mobility scenarios with matching mobility characteristics in various traces; in Section 4.2.6, we motivate our theoretical framework further, by applying our analysis to provide guidelines and performance predictions in protocol design.

4.2.2 Time-variant Mobility Model

In this section, we first motivate the need of incorporating realistic mobility characteristics in the mobility models by contrasting the observations we make from the WLAN traces to the same properties generated by currently available mobility models. The results clearly display the failure of existing models to capture these

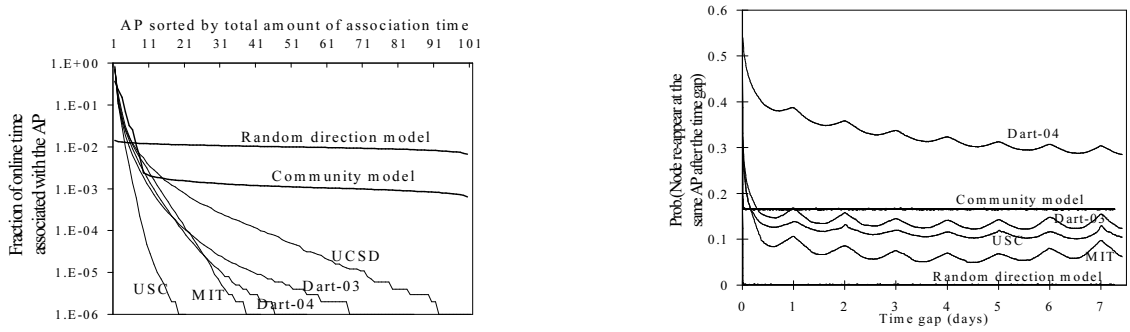
realistic observations. Then we present the design of our *TVC model* inspired by these observations.

4.2.2.1 Mobility characteristics observed in WLAN traces

Our main objective is to propose a mobility model that captures the important mobility characteristics observed in daily life. To better understand this mobility, we have chosen to study a number of wireless LAN traces collected by several research groups (e.g., traces available at [1] or [2]). The reason for this choice is that WLAN traces log information regarding large numbers of nodes, and thus are more reliable for statistical analysis. After analyzing a large number of traces, we have observed two important properties that seem to be recurrent in all of them: *skewed location visiting preferences* and *time-dependent mobility behavior* [66].

First, by *location visiting preference* we mean the amount of time that a node spends associated with a given access point (AP). In Fig. 4-10(a) we calculate for various traces the fraction of the total online time an average node spends with its most favorite AP, its second favorite AP, etc., up to its least favorite AP. (This is essentially the probability density function of the association time of a node with an AP, with the APs sorted in descending order of total association time.) It is clear from the plot that a node on average spends more than 65% of its online time associated with its favorite AP, and more than 95% of its online time at only five APs. We refer to this behavior by saying that the location visiting preference (or in short "location preference") of nodes is *skewed*.

Second, by *time-dependent mobility behavior* we refer to the fact that nodes tend to behave differently during different times of the day (or even during different days), and most specifically to exhibit some amount of periodicity in terms of the locations they choose to visit. In Fig. 4-10(b) we plot the probability of a node appearing in the same location at some time in the future, as a function of the difference in time. It is evident from the plot that nodes appear at the same AP with a higher probability after a time-gap of integer multiples of days. This creates the saw-tooth pattern in the curves. A slightly



(a) (b)

Figure 4-10. Two important mobility features observed from WLAN traces. (a) Skewed location visiting preferences. (b) Periodical re-appearance at the same location. Labels of traces used: MIT: trace from [82], Dart: trace from [81], UCSD: trace from [83], USC: trace from [80].

stronger weekly correlation could also be observed in some plots (see for example the slightly large peak in the MIT curve for a time gap of seven days). It is thus clear that nodes behave differently in different periods in time, and that similar behaviors tend be repeated on a daily basis.

Unfortunately, most existing mobility models fail to capture these two properties. For simple random models, like random direction, random waypoint, random walk, etc., there is obviously no preference in both space and time. This is demonstrated in Fig. 4-10 by a straight line (uniform distribution) for the Random Direction model for the respective probabilities. Even for more sophisticated models that try to capture other aspects of mobility, such as group mobility in the RPGM model [58] or a model considering obstacles and pathways [59], these two properties would also be straight lines in the plots as spatial and temporal preference is not a part of these models[5] . There do exist some more recent models (e.g., [21, 52, 62–64]) that aim at capturing spatial preference explicitly. An example of such a model is the simpler community model of [52]. As is shown in Fig. 4-10(a), with appropriately assigned parameters this model is able to capture the *skewed*

[5] In the case of the obstacle-based model, some locations are not allowed to be visited at all; yet among all the permissible ones, no particular preference is assigned to any node.

location visiting preference, to some extent. However, time-dependent behavior is not captured, and thus the *periodical re-appearance* property cannot be reproduced, as shown by the flat curve labeled *community model* in Fig. 4-10(b).

It is our goal to design a mobility model that successfully captures both of these two properties, observed in the majority of traces. One could argue that a potential shortcoming of this approach is that WLAN traces do not register continuous movement of the devices, but rather associations of users/nodes with specific APs. What is more, some devices are not always on, and typically there are some gaps in the coverage of access points in these networks. However, we believe that the two main properties we observed, namely *skewed location preference* and *time-dependency*, are prevalent in real-life mobility. This belief is further supported by observing typical daily activities of humans: most of us tend to spend most of the time at a handful of frequently visited locations, and a recurrent daily or weekly schedule is an inseparable part of our lives. As a result, a model supporting location-preference and time-dependent mobility should be able to capture human mobility in many contexts, if carefully designed. Comparison with non-WLAN traces in Section 4.2.5 confirms our argument.

4.2.2.2 Construction of the time-variant community model

Skewed location preferences arises naturally due to extended stay at locations that bear importance to us, such as homes and offices, cafeterias and libraries. To capture this phenomenon, we construct popular location(s) for the nodes in the simulation field. These locations, or rather geographical areas, we call *communities*, and we make a node visit its own community more often than other areas. Different nodes can pick different communities, creating nodes with very diverse behaviors. Furthermore, multiple communities could be defined if a node tends to visit multiple locations with high probability, some of which could also be shared by more than one nodes (e.g. people working in the same building, libraries, etc.).

Table 4-1. Parameters of the time-variant community mobility model

For all parameters, we follow the convention that the subscript of a quantity represents its community index, and the superscript represents the time period index.

N	Edge length of simulation area
V	Number of time periods
T^t	Duration of t-th time period
S^t	Number of communities in time period t
C_j^t	Edge length of community j in time period t
$Comm_j^t$	The j-th community during time period t
π_j^t	Probability that the next epoch is performed in community j during time period t
$v_{min}, v_{max}, \overline{v}$	Minimum, maximum, and average speed[6]
$D_{max,j}, \overline{D_j}$	Maximum and average pause time after each epoch[6]
$\overline{L_j}$	Average epoch length for community j
$P_{move,j}^t \vert P_{pause,j}^t$	Probability that a node is moving | pausing when being in community j during period t
P_j^t	Fraction of time the node is in state j ($P_j^t = P_{move,j}^t + P_{pause,j}^t$)
K	Transmission range of nodes
$A(a_j^t, b_k^t)$	The overlapped area between $Comm_j^t$ of node a and $Comm_k^t$ of node b
w^t	A specific relationship between a target coordinate and the communities in time period t
Ω^t	The set of all possible relationships between a target coordinate and the communities in time period t
$P(w^t)$	Probability of a given relationship w^t in time period t
$A(w^t)$	Area corresponding to the relationship w^t
$P_h(w^t)$	Unit-time hitting probability under the specific scenario w^t
$P_H(w^t)$	Hitting probability for a time period t under specific scenario w^t
P_m	Unit-time meeting probability
P_M	Meeting probability for a time period t
$HT(case)$	Expected hitting time under the given "case"
$MT(case)$	Expected meeting time under the given "case"

Periodical re-appearance at a given location is related to omnipresent schedules in our lives. Almost everyone follows recurrent daily or weekly schedules, and different behaviors based on time-of-day have been observed in many contexts. To capture time-dependent behaviors, we introduce structure in the time domain by the use of *time periods*. For a given node, we assign several *time periods* during which it behaves differently. For example, a node may have different communities during different periods or the same communities but different mobility parameters to move between them. To further ensure periodicity, the time period assignment follows a *recurrent structure*, with the same "time-period" and its respective statistical characteristics occurring, say, for all weekday mornings.

We illustrate the model with an example in Fig. 4-11. We also use this example to introduce the notations we use (see Table 4-1) in the rest of the section. As shown in

the example, there are $V = 3$ (where V denotes the total number of time periods) time periods TP1, TP2, and TP3 (of duration T^1, T^2, and T^3, respectively). During each time period, there are a number of communities, that is, geographical areas that are heavily visited. These communities can be chosen differently in each time period, as shown by the three sub-plots. Within a given time period t, the j-th community is denoted as $Comm_j^t$.[6] This is a square geographical area of edge length C_j^t. Note that by construction the communities can overlap (as in TP1 in Fig. 4-11), or one community can even contain the other (as in TP2 in Fig. 4-11). Finally, the number of communities in each time period may vary. For example, there are 3 communities in total in the first period, 2 in the second one, and 4 in the third[7] . The total number of communities in period t is denoted as S^t. This construction allows for maximum flexibility when designing the simulation setup for nodes with different behaviors. As for the structure in time domain, we need to arrange time periods in a re-current sequence (see Fig. 4-11 or Fig. 4-12) that corresponds to a daily or weekly schedule.

We now describe how a node moves inside the above construction. Node movement consists of a sequence of *epochs*. Each of these epochs is a Random Direction movement[8] . In a typical Random Direction epoch, a node chooses at the beginning its speed uniformly in $[v_{min}, v_{max}]$, and a direction (angle) uniformly in $[0, 2\pi]$; it also chooses the length (distance) of movement (usually distributed exponentially with average in the order of the network dimension), and moves towards this direction with the chosen speed and for the

[6] For all parameters used in this work, we follow the convention that the subscript of a quantity represents its community index, and the superscript represents the time period index. Note that all parameters used in the TVC model can be set differently for *each node*. When necessary, we use a pair of parentheses to include the node ID for a particular parameter, e.g., $C_j^t(i)$ denotes the edge length of the j-th community during time period t for node i.

[7] To allow a node to move randomly among the whole simulation field sometimes, we often allocate one community to be the whole simulation field (e.g. $Comm_3^1$ in period TP1 in Fig. 4-11).

[8] Note that we could also choose random waypoint or random walk models for the type of movement during each epoch.

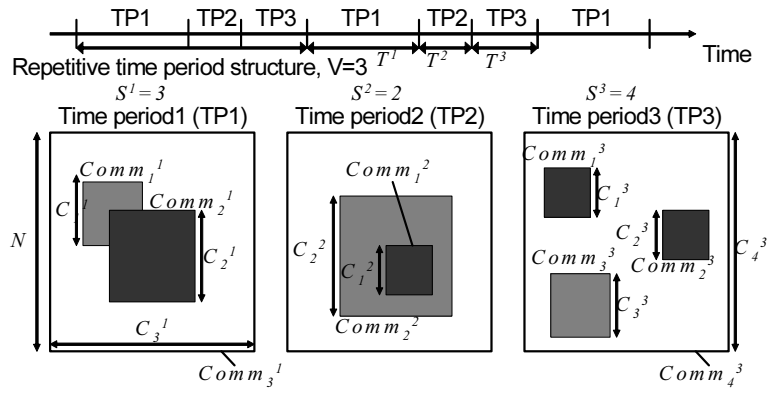


Figure 4-11. Illustration of a generic scenario of the time-variant mobility model, with three time periods and different numbers of communities in each time period.

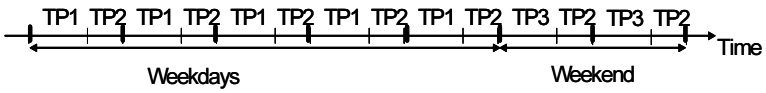


Figure 4-12. An illustration of a simple weekly schedule, where we use time period 1 (TP1) to capture weekday working hour, TP2 to capture night time, and TP3 to capture weekend day time.

chosen distance; at the end of the epoch, the node picks a pause time randomly and then proceeds to the next epoch.

The difference between our community model and the Random Direction model is that in addition to all other parameters, in the community model the node also chooses randomly the community in which the next epoch will be performed. That is, with probability π_j the next epoch takes place inside the node's j-th community ($\sum_{j=1}^{S^t} \pi_j = 1$), rather than moving around the whole simulation area randomly, as in the standard Random Direction model. (Note that we usually add superscript t in the notation, i.e. π_j^t, to denote that these probabilities might change between time periods.) We say the node is in *state* j when it has an epoch in the j-th community. Further, to ensure that a local move is compatible with the local community size, we also scale the local epoch length by drawing it from an exponential distribution with average length $\overline{L_j}$, that is, in the order

of the given community size[9] . It is important to note that a node can still perform some of its epochs in the whole simulation area, by assigning an additional community that corresponds to the whole simulation field (e.g. $Comm_3^1$ in period TP1 in Fig. 4-11). We refer to such epochs as *roaming epochs*. Finally, after an epoch, a node pauses for a time uniformly chosen in $[0, D_{max,j}]$, where the maximum pause time is again dependent on the community.

As a final note, one may argue that capturing location preference and time-dependencies could plausibly be achieved with a mobility model constructed with different ways than the one we propose. However, our choices are largely guided by the fact that most of the building blocks we utilize to create our mobility model (e.g. random direction epochs, communities, etc.) are easy to understand, and have been shown to be amenable to theoretical analysis [52]. The benefits will become evident in Section 4.2.3. At the same time, we will also show in section 4.2.5 that these choices do not compromise our model's ability to accurately capture real life mobility scenarios.

4.2.3 Theoretical Analysis of the TVC Model

So far, we have established the general framework of the TVC model. We make the framework very flexible in order to create a model that can be used in many realistic contexts. Yet, one of the biggest advantages of our model is that, in addition to the realism, it is also analytically tractable with respect to some important quantities which determine protocol performance. We focus on demonstrating this last point in this section.

We start here by deriving the theoretic expressions of various properties of the proposed mobility model. We first calculate the *nodal spatial distribution*. This can be represented as a two-dimensional probability density function of nodes at any given point in the space. The expected number of nodes in an area is then calculated by integrating

[9] To avoid boundary effects, we assume that if the node hits the community boundary it is re-inserted from the other end of the area (i.e., the boundaries are "torus" boundaries).

the density function. This property provides a basic demographic profile of the simulation area, and can also help to evaluate whether the model reflects the target environment. Then, we derive the *average node degree*, which is the average number of nodes residing within the communication range of a given node. This is a quantity of interest due to its implication on the success rate of various tasks in mobile ad hoc networks [86, 87]. Finally, we derive the expected *hitting* and *meeting times* for our model. The hitting time is the time it takes a node, starting from the stationary distribution, to move within transmission range of a fixed, randomly chosen target coordinate in the simulation field. The *meeting time* is the time until two mobile nodes, both starting from the stationary distribution, move into the transmission range of each other. These two quantities are of interest due to their close relationship to the performance of DTN routing protocols, or in general the performance of processes that rely on node encounters. Knowing the meeting time for a mobility model is, for example, crucial when using a "mobility-assisted" or "store-carry-and-forward" protocol to deliver a message [55–57], while hitting times might be needed if some nodes in the network are static (e.g. sensors, base stations, etc.).

We note that a preliminary version of some of the theoretical derivations presented here appear for a special case of our TVC model only in [67] (that model included one community and two time periods only). Here, we generalize all derivations for any community and time-period structure. Moreover, we present some additional results regarding the spatial distribution and the average node degree that are relevant to various wireless communication protocols, as we show in Section 4.2.6. We start with a useful lemma that calculates the probability of a node to reside in a particular state.

Lemma 4.1. *The probability that a node moves or pauses (after the completion of an epoch) in state j, at any given time instant during time period t, is:*

$$P^t_{move,j} = \pi^t_j(\overline{L^t_j}/\overline{v^t_j}) / \sum_{k=1}^{S^t} \pi^t_k(\overline{L^t_k}/\overline{v^t_k} + \overline{D^t_k}), \tag{4–1}$$

$$P^t_{pause,j} = \pi^t_j \overline{D^t_j} / \sum_{k=1}^{S^t} \pi^t_k (\overline{L^t_k}/\overline{v^t_k} + \overline{D^t_k}). \tag{4–2}$$

Proof. The result follows from the ratio of the average durations of the moving part ($\overline{L^t_j}/\overline{v^t_j}$) and the pause part ($\overline{D^t_j}$) of each state, weighted by the probabilities of choosing the state. □

Note that the above stationary probabilities can be calculated for each time period and node separately. We use $P^t_j(i)$ to denote the probability that node i is in state j during time period t (i.e., $P^t_j(i) = P^t_{move,j}(i) + P^t_{pause,j}(i)$).

4.2.3.1 Nodal spatial distribution

We start with the derivation of the nodal spatial distribution. This becomes relatively straight-forward after we observe that a node follows a basic *random mobility model* (i.e., random direction) in each community. Hence, when a given node is in state j, it appears equally likely in any point within $Comm_j$.

Theorem 4.2. *For a given area A, the probability for a node to appear in A at any given time instant during time period t is*

$$\iint_A p(x,y) \ \ dxdy, \tag{4–3}$$

where the function $p(x,y)$ is the spatial density function,

$$p(x,y) = \sum_{\{\forall j | (x,y) \in Comm^t_j\}} P^t_j / C^{t2}_j. \tag{4–4}$$

Proof. A node could appear at a given point in space when it is in state j if and only if the j-th community includes the point. Within the community, the appearance probability of the node is uniformly distributed. Considering a given point (x,y), the probability for a node to appear at the point is the sum of the contributions from all of its communities that contain the point. □

Note that the nodal spatial distribution for each time period is independent, hence can be calculated separately with the above Theorem.

4.2.3.2 Average node degree

The average node degree of a node is defined as the expected number of nodes falling within its communication range. Each node contributes to the average node degree independently, as nodes make independent movement decisions.

Lemma 4.3. *Consider a pair of nodes, a and b. Assume further that, in time period t, community j of node a and community k of node b overlap with each other for an area $A(a_j^t, b_k^t)$. Then, the contribution of node b to the average node degree of node a, when a resides in its j-th community and b resides in its k-th community, is given by*

$$\frac{\pi K^2}{C_j^{t\,2}(a)} \frac{A(a_j^t, b_k^t)}{C_k^{t\,2}(b)}, \tag{4–5}$$

where K is the communication range of the nodes.

Proof. Since nodes are uniformly distributed within each community, the probability for node b to fall in the j-th community of node a is simply the ratio of the overlapped area over the size of the k-th community of node b. Node a covers any given point in its community equal-likely, hence given node b is in the overlapped area, it is within the communication range of node a with probability $\pi K^2 / C_j^{t\,2}(a)$. □

Following the same principle in Lemma 4.3, we include all community pairs and arrive at the following Theorem.

Theorem 4.4. *The average node degree of a given node a is*

$$\sum_{\forall Comm_j^t(a)} P_j^t(a) \sum_{\forall b} \sum_{\forall Comm_k^t(b)} P_j^t(b) \frac{\pi K^2}{C_j^{t\,2}(a)} \frac{A(a_j^t, b_k^t)}{C_k^{t\,2}(b)}. \tag{4–6}$$

Proof. Eq. (4–6) is simply a weighted average of the node degree of node a conditioning on its states. For each state with probability $P_j^t(a)$, the expected node degree is a sum

over all other nodes' probability of being within the communication range of node a, again conditioning on all possible states. □

Corollary 4.5. *In the special case when all nodes choose their communities uniformly at random among the simulation field, Eq. (4–6) degenerates to*

$$\begin{aligned} &\sum_{\forall b} \sum_{\forall Comm_k^t(b)} P_k^t(b) \frac{\pi K^2}{{C_k^t}^2(b)} \frac{{C_k^t}^2(b)}{N^2} \\ =&\sum_{\forall b} \frac{\pi K^2}{N^2} \sum_{\forall Comm_k^t(b)} P_k^t(b) = \sum_{\forall b} \frac{\pi K^2}{N^2}. \end{aligned} \tag{4–7}$$

Proof. This result follows from that a randomly chosen community is anywhere in the simulation field equally likely. If nodes pick their communities randomly and independently, the actual location of node a would not make any difference in its average node degree. Regardless of the location of node a, it falls within the k-th community of node b with probability ${C_k^t}^2(b)/N^2$. Within the community, node b appears uniformly, and with probability $\pi K^2/{C_k^t}^2(b)$ it appears within the communication range of node a. Note that the equation reduces to each node b contributing $\pi K^2/N^2$ to the average node degree, which is the same *as if node b roams around the whole simulation area without any preference in space (i.e., communities).* □

Similar to the nodal spatial distribution, the average node degree can be calculated for each time period separately.

4.2.3.3 Hitting time

The sketch of the derivation of the hitting time is as follows: (1) We first condition on the relative location of the target coordinate with respect to a node's communities (e.g. target inside community i, target outside community, etc.). We thus have to derive the hitting time for each sub-case separately. (2) We then derive the *unit-step* probabilities of hitting a target, P_h, for a given sub-case. The unit-step probability is the probability of encountering the target exactly within the next time-unit (rather than within the duration of a whole epoch). In other words, we approximate the continuous mobility with a discrete

version of it where nodes move in discrete steps. It has been shown in [52] that the latter provides a good approximation for the continuous version, and is easier to analyze for our purposes. (3) The expected hitting probability for a whole time period, P_H, is then calculated for each sub-case from the unit-step probability, by assuming "hitting" occurs independently in each time step[10] . (4) Finally, taking the weighted average of each sub-case (i.e. weighted by probability of a given target being located inside a given community) we get the overall hitting time.

The most influential factor for the hitting time is whether the target coordinate is chosen inside the node's communities. We denote the possible relationships between the target location and the set up of communities during time period t as the set Ω^t. Note that the cardinality of set Ω^t is at most 2^{S^t} (i.e. for each of the S^t communities, the target coordinate is either in or out of it). Also, not all of the 2^{S^t} combinations are always valid. For example, in the set up of time period 2 in Fig. 4-11, the communities are overlapped, hence if the target is within $Comm_1^2$ it must be within $Comm_2^2$.

Lemma 4.6. *By the law of total probability, the average hitting time can be written as*

$$HT = \sum_{w^1 \in \Omega^1, ..., w^V \in \Omega^V} P(w^1, ..., w^V) HT(w^1, ..., w^V), \tag{4–8}$$

where $w^1, w^2, ..., w^V$ denote one particular relationship (i.e. a combination of $\{out, in\}^{S^t}$) between the target coordinate and the community set up during time period $1, 2, ..., V$, respectively. Functions $P(\cdot)$ and $HT(\cdot)$ denote the corresponding probability for this scenario and the conditional hitting time under this scenario, respectively. Note that each sub-case $\{w^1, w^2, ..., w^V\}$ is disjoint from all other sub-cases.

[10] This assumption of independence is shown in [52] to be a good approximation, when the expected length of an epoch is in the order of the square root of the area of the community the epoch takes place in.

To evaluate Eq. (4–8), we need to calculate $P(w^1, ..., w^V)$ and $HT(w^1, ..., w^V)$ for each possible sub-case $(w^1, ..., w^V)$.

Lemma 4.7. *If the target coordinate is chosen independent of the communities and the communities in each time period are chosen independently from other periods, then*

$$P(w^1, ..., w^V) = \Pi_{t=1}^{V} P(w^t), \tag{4–9}$$

where $P(w^t) = A(w^t)/N^2$, i.e., the probability of a sub-case w^t is proportional to the area $A(w^t)$ that corresponds to the specific scenario w^t, which is a series of conditions of the following type: $(\{target \in comm_1^t\}, \{target \notin comm_2^t\}, ..., \{target \in comm_S^t\})$.

Proof. The result follows from simple geometric arguments. □

The first step for calculating $HT(w^1, ..., w^V)$ is to derive the unit-time hitting probability in time period t under target coordinate-community relationship w^t, denoted as $P_h^t(w^t)$.

Lemma 4.8. *For a given time period t and a specific scenario w^t,*

$$P_h^t(w^t) = \sum_{j=1}^{S^t} I(target \in Comm_j^t | w^t) P_{move,j}^t 2K\overline{v_j^t}/C_j^{t2}, \tag{4–10}$$

where $I(\cdot)$ is the indicator function.

Proof. The overall unit-time hitting probability is the sum of the hitting probabilities contributed by epochs in each state. Note that the hitting event can only occur when the node is physically moving, and the node can hit the target when it is moving in its j-th community only if the target coordinate is within the community[11] . When a node moves with average speed $\overline{v_j}$ in community j, on average it covers a new area of $2K\overline{v_j}$ in unit time. Since a node following random direction movements visits the area it moves about

[11] We neglect the small probability that the target is chosen out of the community but close to it, and make the contributions from epochs in state j zero if the chosen target coordinate is not in community j.

with equal probability, and the target coordinate is chosen at random, it falls in this newly covered area with probability $2K\overline{v_j}/C_j^{t2}$ [52]. Hence the contribution to the unit-time hitting probability by movements made in state j is $P_{move,j}^t 2K\overline{v_j^t}/C_j^{t2}$, i.e., when the node moves in community j *and* the target is in the newly covered area in the time unit. □

Note that the movement made in each time unit does not increase or decrease the probability of hitting the target in the subsequent time units, therefore each time unit can be considered as an independent Bernoulli trial with success probability given in Eq. (4–10). The corollary below immediately follows.

Corollary 4.9. *The probability for at least one hitting event to occur during time period t under scenario w^t is*

$$P_H^t(w^t) = 1 - (1 - P_h^t(w^t))^{T^t}. \tag{4–11}$$

Finally, using the law of total probability, we derive the conditional hitting time under a specific target-community relationship, $HT(w^1, ..., w^V)$.

Theorem 4.10.

$$\begin{aligned} HT(w^1, ..., w^V) = \sum_{t=1}^{V} HT(w^1, ..., w^V | first\ hit\ in\ period\ t) \cdot \\ P(w^1, ..., w^V, first\ hit\ in\ period\ t), \end{aligned} \tag{4–12}$$

where the probability for the first hitting event to happen in time period t is

$$P(w^1, ..., w^V, first\ hit\ in\ period\ t) = \frac{\Pi_{i=1}^{t-1}(1 - P_H^i(w^i)) \cdot P_H^t(w^t)}{P}, \tag{4–13}$$

and the hitting time under this specific condition is

$$HT(w^1, ..., w^V | first\ hit\ in\ period\ t) = \sum_{i=1}^{V} T^i \cdot (\frac{1}{P} - 1) + \sum_{i=1}^{t-1} T^i + \frac{1}{P_h^t(w^t)}, \tag{4–14}$$

where $P = 1 - \Pi_{t=1}^{V}(1 - P_H^t(w^t))$ is the hitting probability for one full cycle of time periods.

Proof. The probability for the first hitting event to happen in time period t can be derived as follows: we consider the occurrence of hitting events in each type of time periods as independent coin toss trials, which give head with probability $P_H^t(w^t)$ for time period t. The probability of the first hitting event occurring in time period t is equivalent to the probability of getting the first head from t-th coin, when we flip these coins in turns, following the same order as the time periods appearing in the structure. The success probability for each full cycle is $P = 1 - \Pi_{i=1}^{T}(1 - P_H^i(w_i))$. The probabilities for the first hitting event to occur in time period t is as given in Eq. (4–13), since in each cycle of time periods follows the same repetitive structure. The first term in Eq. (4–14) corresponds to the expected duration of full time period cycles until the hitting event occurs. Since for each cycle the success probability of hitting the target is P, in expectation it takes $1/P$ cycles to hit the target, and there are $1/P - 1$ full cycles. The second term in Eq. (4–14) is the sum of duration of time periods before the time period t in which the hitting event occurs in the last cycle. Finally, the third term is the fraction of the last time period before the hitting event occurs. Note that the last part is an approximation which holds if the time periods we consider are much longer than unit-time. □

4.2.3.4 Meeting time

The procedures of the derivation of the meeting time is similar to that of the hitting time detailed in the last section. In short, we derive the unit-step (or unit-time) meeting probability, P_m, and the meeting probability for each type of time period, P_M, and put them together to get the overall meeting time in a similar fashion as in Theorem 4.10.

Similar to Lemma 4.8, we add up the contributions to the meeting probability from all community pairs from node a and b in the following Lemma.

Lemma 4.11. *Let community j of node a and community k of node b overlap with each other for an area $A(a_j^t, b_k^t)$ in time period t. Then, the conditional unit-time meeting probability in time period t when node a and b are in its community j and k, respectively, is*

$$
\begin{aligned}
P_m^t(a_j^t, b_k^t) = {} & P_{move,j}^t(a) P_{move,k}^t(b) \hat{v} \frac{2K\overline{v}}{A(a_j^t, b_k^t)} \frac{A(a_j^t, b_k^t)}{C_j^{t\,2}(a)} \frac{A(a_j^t, b_k^t)}{C_k^{t\,2}(b)} \\
& + P_{move,j}^t(a) P_{stop,k}^t(b) \frac{2K\overline{v}}{A(a_j^t, b_k^t)} \frac{A(a_j^t, b_k^t)}{C_j^{t\,2}(a)} \frac{A(a_j^t, b_k^t)}{C_k^{t\,2}(b)} \\
& + P_{stop,j}^t(a) P_{move,k}^t(b) \frac{2K\overline{v}}{A(a_j^t, b_k^t)} \frac{A(a_j^t, b_k^t)}{C_j^{t\,2}(a)} \frac{A(a_j^t, b_k^t)}{C_k^{t\,2}(b)}.
\end{aligned} \tag{4–15}
$$

Proof. Equation (4–15) consists of two parts:

(I) Both of the nodes are moving within the overlapped area. This adds the first term in Eq. (4–15) to the meeting probability. The two ratios, $\frac{A(a_j^t, b_k^t)}{C_j^{t\,2}(a)}$ and $\frac{A(a_j^t, b_k^t)}{C_k^{t\,2}(b)}$, capture the probabilities that the nodes are in the overlapped area of the communities. The contribution to the unit-time meeting probability is the product of probabilities of both nodes moving within the overlapped area and the term $\frac{2K\overline{v}}{A(a_j^t, b_k^t)}$, which reflects the covered area in unit time. We use the fact that when both nodes move according to the random direction model, one can calculate the effective (extra) area covered by assuming that one node is static, and the other is moving with the (higher) *relative speed* between the two. This difference is capture with the multiplicative factor $\hat{v}$ [52].

(II) One node is moving in the overlapped area, and the other one pauses within the area. This adds the remaining two terms in Eq. (4–15) to the unit-time meeting probability. These terms follow similar rationale as the previous one, with the difference that now only one node is moving. The second term corresponds to the case when node a moves (and b is static), and the third term corresponds to the case when node b moves (and a is static).

The derivation of the unit-time meeting probability between nodes a and b for time period t includes all possible scenarios of community overlap. If node a has $S^t(a)$ communities and node b has $S^t(b)$ communities, there can be at most $S^t(a)S^t(b)$ community-overlapping scenarios in time period t. □

Note that (4–15) is the general form of Equation (13) and (14) in [67]. If we assume perfect overlap and a single community from both nodes, we arrive at (14). If we assume no overlap, we result in (13). Also note in the general expressions presented in this work, the whole simulation area is also considered as a community. Therefore we do not have to include a separate term to capture the roaming epochs.

Corollary 4.12. *The probability for at least one meeting event to occur during time period t is*

$$P_M^t = 1 - \sum_{\forall(j,k)} \{P_{ov}(a_j^t, b_k^t) \cdot (1 - P_m^t(a_j^t, b_k^t))^{T^t}\}, \tag{4–16}$$

where $P_{ov}(a_j^t, b_k^t)$ is the probability that the community j of node a overlaps with community k of node b. This quantity is simply 1 if the communities have fixed assignments and $A(a_j^t, b_k^t) \neq 0$. If the communities are chosen randomly, this probability can be derived by Lemma 4.5 in [67]. The Lemma is re-produced below for completeness.

Lemma 4.13. *For a specific time period t, if the j-th community of node a and the k-th community of node b are randomly chosen within the simulation area, they overlap with probability*

$$P_{ov}(a_j^t, b_k^t) = \frac{(C_j^t(a) + 2K)^2}{N^2}. \tag{4–17}$$

Proof. As shown in Fig. 4-13, when a mobile node moves within its community, the area covered by the node (i.e., the area that could fall in the communication range of the node) actually extends out of the community by the transmission range of the node. Hence, the "footage" of the community is larger than C_j^{t2}. We approximate this area by $(C_j^t + 2K)^2$, ignoring the small differences at the corners. Finally, since each node selects its community at random within the simulation area, the probability that part of the footage of the community of node a is chosen as part of the community of node b is simply $\frac{(C_j^t(a)+2K)^2}{N^2}$. □

Finally, similarly to Theorem 4.10, the expected meeting time can be calculated using the results in the Lemmas in this section.

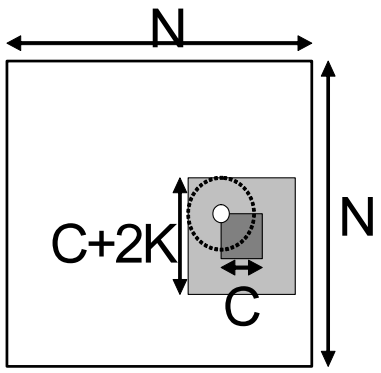


Figure 4-13. Illustration of the expansion of the "footage" of community.

Theorem 4.14. *The expected meeting time is*

$$MT = \sum_{t=1}^{V} MT(\textit{meet in period } t)P(\textit{meet in period } t). \tag{4–18}$$

Where the quantities in the above equation are calculated by

$$P(\textit{meet in period } t) = \frac{\Pi_{i=1}^{t-1}(1 - P_M^i) \cdot P_M^t}{Q}, \tag{4–19}$$

$$MT(\textit{meet in period } t) = \sum_{i=1}^{V} T^i \cdot (\frac{1}{Q} - 1) + \sum_{i=1}^{t-1} T^i + \frac{1}{P_m^t}, \tag{4–20}$$

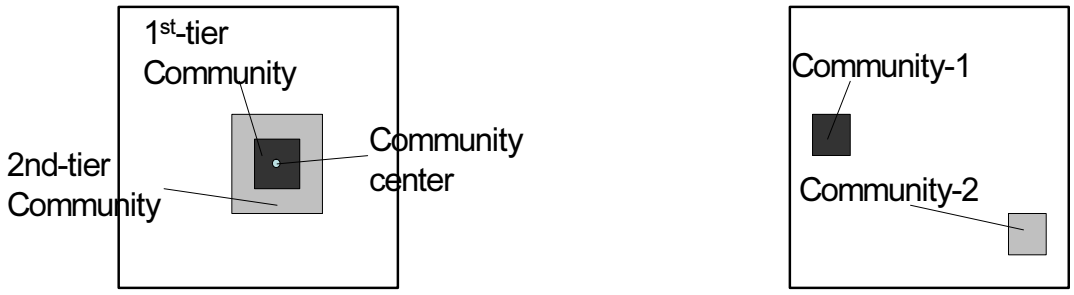
where $Q = 1 - \Pi_{i=1}^{V}(1 - P_M^i)$ *is the meeting probability for one full cycle of time periods.*

Proof. The proof is parallel to that of Theorem 4.10 and is omitted. □

4.2.4 Validation of the Theory with Simulations

In this section, we compare the theoretical derivations of the previous section against the corresponding simulation results, for various parameter settings. Through extensive simulations with multiple scenarios and parameter settings, we establish the accuracy of the theoretical framework.

We summarize the parameters for the tested scenarios in Table 4-2. Table 4-2 (a) lists the parameters we use for a simplified model (two time periods with two communities in each time period, where one of the communities is the whole simulation field). For more complex models, we try out the setup of *tiered communities* and *multiple randomly*



(a) (b)

Figure 4-14. Illustration of the community setup for the generic cases of TVC model. (a) Concentric multiple-tier communities setting. (b) Multiple randomly placed communities setting.

placed communities. In the *tiered communities* layout, as illustrated in Fig. 4-14(a), a randomly chosen point in the simulation field serves as the *center* of the communities, and multiple tiers of communities with different sizes share the same center. This construction is suggested by a common observation from our daily lives: People visit the vicinity area of locations that bear importance to them more often than roam far away. When we assign the tiered community structure, it naturally makes sense to have the node visit the outer tiers less frequently than the inner tiers, although this is not required for the theoretical derivation. In the simulations, we use two alternative time periods with a two-tier local community in each time period, and the parameters are listed in Table 4-2 (b). In the *multiple randomly placed communities* layout, as illustrated in Fig. 4-14(b), multiple communities are instantiated randomly to show that our theory is not limited to a single community. We use two time periods with two randomly placed communities each for this scenario. Other than the difference in community setup and sizes, we again use the parameters in Table 4-2 (b) for this case. Our discrete-time simulator is written in C++, and nodes move as described in Section 4.2.2. More details about the simulator, as well as the simulator code, can be found at [9].

Table 4-2. Parameters for the scenarios in the simulation

Common parameters: For simplicity, we use the same movement speed for all node: $v_{max} = 15$ and $v_{min} = 5$ in all scenarios. In all cases we use two time periods and they are named as time period 1 and 2 for consistency. In the simple model we use a single local community (with subscript l) in each time period. For the generic model, we test with two different configurations: (1) A two-tier community in each time period - in this scenario the inner tier community and the outer tier community has edge length C_{l1} and C_{l2}, respectively. (2) Two randomly placed communities in each time period - in this scenario the communities both have edge length C_{l1}, but the parameters correspond to the two communities are different (i.e., correspond to subscript $l1$ and $l2$ in the table). In all cases, there is also a roaming state (with subscript r) in which the node moves about the whole simulation area (i.e. the whole simulation area is a community).

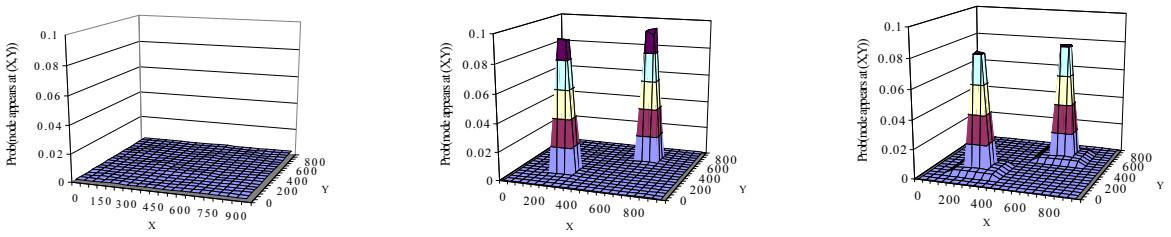
(a) The simple model.

Model name	Description	N	C_l^1	C_l^2	$D_{max,l}$	$D_{max,r}$	$\overline{L_l}$	$\overline{L_r}$	π_l^1	π_r^1	π_l^2	π_r^2	T^1	T^2
Model 1	Match with the MIT trace	1000	100	100	100	50	80	520	0.714	0.286	0.8	0.2	5760	2880
Model 2	Highly attractive communities	1000	200	50	100	200	52	520	0.667	0.333	0.889	0.111	3000	2000
Model 3	Not attractive communities	1000	100	100	50	200	80	800	0.5	0.5	0.667	0.333	2000	1000
Model 4	Large-size communities	1000	200	250	50	100	200	800	0.7	0.3	0.889	0.111	2000	1000

(b) The generic model.

Model name	N	C_{l1}^1	C_{l2}^1	C_{l1}^2	C_{l2}^2	$D_{max,l1}^1$	$D_{max,l2}^1$	$D_{max,r}^2$	$D_{max,l1}^2$	$D_{max,l2}^2$	$D_{max,r}^2$
Model 5	1000	100	300	100	300	25	15	1	30	25	3
Model 6	1000	150	450	150	450	50	20	15	30	20	30
Model 7	1200	160	480	160	480	50	20	15	30	20	30

Model name	$\overline{L_{l1}^1}$	$\overline{L_{l2}^1}$	$\overline{L_r^1}$	$\overline{L_{l1}^2}$	$\overline{L_{l2}^2}$	$\overline{L_r^2}$	π_{l1}^1	π_{l2}^1	π_r^1	π_{l1}^2	π_{l2}^2	π_r^2	T^1	T^2
Model 5	300	500	1000	200	300	1000	0.6	0.3	0.1	0.85	0.1	0.05	5760	2880
Model 6	100	300	1000	200	500	1000	0.5	0.35	0.15	0.7	0.2	0.1	5760	2880
Model 7	140	600	1500	200	500	1600	0.8	0.15	0.05	0.7	0.2	0.1	5760	2880



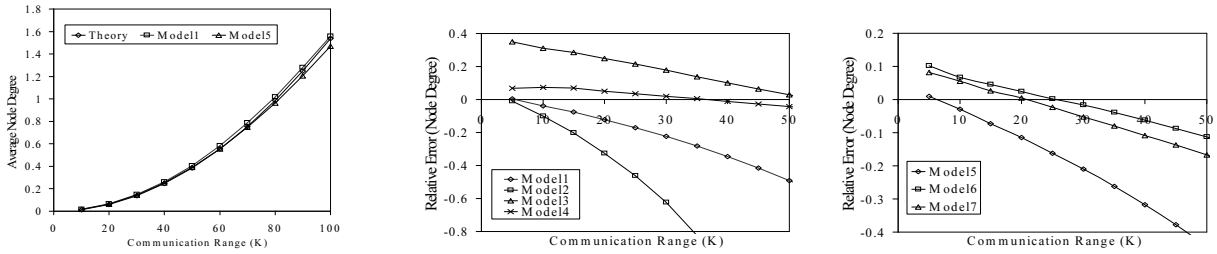
(a) (b) (c)

Figure 4-15. Spatial distribution of the node (shown as the probability for a node to appear in each 50x50 grid block). (a) Randomly placed community. (b) Single-tier community centered at $(300, 300)$ or $(700, 700)$ with one half probability. (c) Two-tier community centered at $(300, 300)$ or $(700, 700)$ with one half probability.

4.2.4.1 Nodal spatial distribution

To observe the nodal spatial distribution, we divide the simulation area into a 20-by-20 grid and count the average number of nodes in each grid block during the simulation. The results presented in this subsection is the average of 5000 runs of independent simulations.

If the communities are randomly chosen, the node should appear at each of the 400 evenly divided grid equal-likely, with probability 1/400. We observe that the spatial distribution of node varies a bit about this value in the simulation, as shown in Fig. 4-15 (a). The minor discrepancy is due to the finite number of samples. To make the scenario more interesting, we also generate the spatial distribution for nodes when the communities are fixed. We use the parameter sets of *Model-1* (one community in each time period) and *Model-5* (two-tier community in each time period) from table 4-2, and assign the center of the community at either $(300, 300)$ or $(700, 700)$ with one half probability. The resulting nodal spatial distributions are shown in Fig. 4-15 (b) and (c), respectively. The node appears with higher probability where the communities are assigned. From Eq. (4–3), for the scenario in Fig. 4-15 (b), the node appears in the community with probability 0.0864 and in other area with probability 0.0008, respectively. For the scenario in Fig. 4-15 (c), the node appears in the first-tier community, the second-tier community, and the other



(a) (b) (c)

Figure 4-16. Comparison of theoretical and simulation results (the average node degree). (a) Randomly placed community. (b) Relative error for scenarios with fixed single-tier community. (c) Relative error for scenarios with fixed two-tier communities.

area with probability 0.0759, 0.0039, and 0.0004, respectively. In both cases the simulation results follow the theoretical results reasonably well, within about 10% error for the area in the communities.

4.2.4.2 Average node degree

For the average node degree, we create simulation scenarios with 50 nodes in the simulation area, and calculate the average node degree of each node by taking the time average across snapshots taken every second during the simulation, and then average across all nodes. All the runs last for 60000 seconds in this subsection.

As we show in Corollary 4.5, when the communities are randomly chosen, the average node degree turns out to be the average number of nodes falling in the communication range of a given node, *as if all nodes are uniformly distributed*. Hence the average node degree does not depend on the exact choices of community setup (i.e. single, multiple, or multi-tier communities) or other mobility parameters. In Fig. 4-16 (a), we compare the evolution of the theoretical average node degree versus the communication range (K) to the simulation results for some of the models listed in Table 4-2. The simulation curves follow the prediction of the theory well. Other configurations we tried (not listed here) also show similar trends.

Again, to make the scenario a bit more realistic, we simulate some more scenarios when the communities are fixed. Among the 50 nodes, we make 25 of them pick the

community centered at $(300, 300)$ and the other 25 pick the community centered at $(700, 700)$. We simulate scenarios for all seven sets of parameters listed in Table 4-2. Models 1 through 4 correspond to scenarios with single-tier communities in each time period, and models 5 through 7 correspond to scenarios with multi-tier communities. We show the relative errors, calculated as $Error = (Theory - Simulation)/Simulation$, in Fig. 4-16 (b) and (c). A positive error indicates the theoretical value is larger than the simulation result, while a negative error indicates the converse. In the simulations, when the communication ranges are small as compared to the edge of the communities, the relative errors are low, typically below 10% except for *Model-3*, indicating a good match between the theory and the simulation. However, as the communication range increases, the area covered by the communication disk becomes comparable to the size of the community and Eq. (4–5) is no longer accurate since the communication disk extends out of the overlapped area in most cases. That is the reason for the discrepancies between the theory and simulation. Besides *Model-3*, we observe at most 20% of relative error when the communication disk is less than 20% the size of the inner-most community, indicating that our theory is valid when the communication range is relatively small.

4.2.4.3 Hitting time and meeting time

We perform simulations for the hitting and the meeting times for $50,000$ independent iterations for each scenario listed in Table 4-2, and compare the average results with the theoretical values derived from the corresponding equations (i.e. (4–8) and (4–18)). To find out the hitting or the meeting time, we move the nodes in the simulator indefinitely until they hit the target or meet with each other, respectively.

Again we show the relative errors between the theoretical values and the simulation results for various scenarios in Fig. 4-17. We see that for all the scenarios, the relative errors are within acceptable range. These results display the accuracy of our theory under a wide range of parameter settings. The absolute values for the error are within 16% for the hitting time and within 20% for the meeting time. For more than 70% of the tested

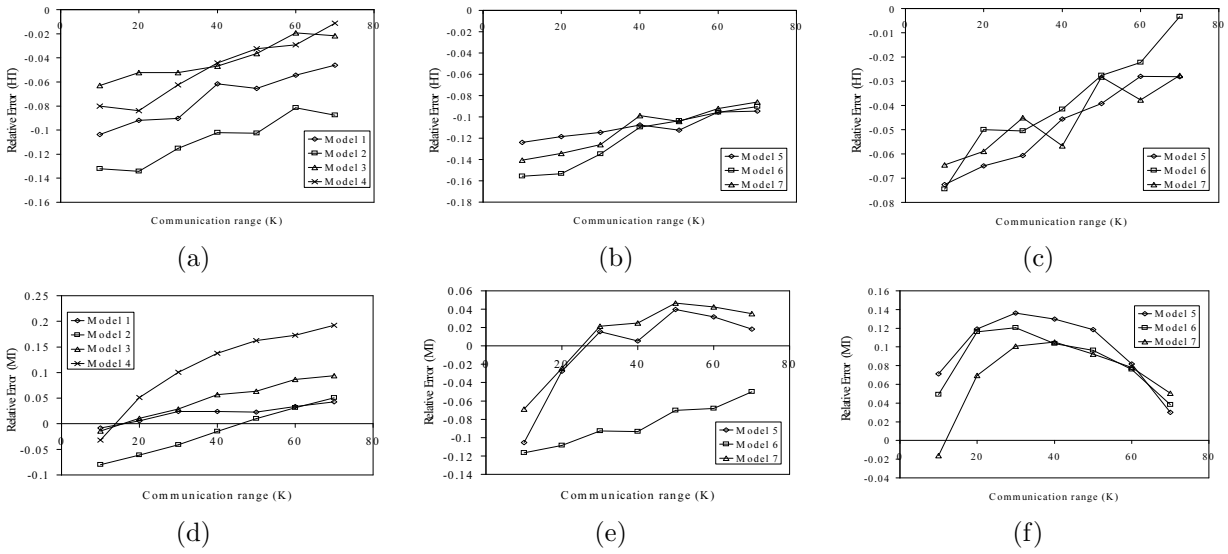


Figure 4-17. Relative error between theoretical and simulation results (the hitting time and the meeting time). (a) Hitting time, simple model. (b) Hitting time, multi-tier communities. (c) Hitting time, multiple random communities. (d) Meeting time, simple model. (e) Meeting time, multi-tier communities. (f) Meeting time, multiple random communities.

scenarios, the error is below 10%. The errors between the theoretical and simulation results are mainly due to some of the approximations we made in the various derivations. For example, there exist some border effects with respect to the hitting and meeting probabilities within a community. When a node is close to the border of a community, it could also "see" some other nodes outside of the community if its transmission range is large enough. However, we have chosen to ignore such occurrences to keep our analysis simpler. Furthermore, the approximation of the hitting and meeting processes with discrete Bernoulli trials is valid only for the epochs that are large enough (in the order of community size). Nevertheless, as shown in the figures, the errors are always within acceptable ranges, justifying our simplifying assumptions.

4.2.5 Application I: Generation of Mobility Scenarios for Simulation

After establishing the theoretical results in previous sections, in this and the next sections we display the usefulness of the TVC model. The TVC model is flexible to match the mobility characteristics we obtain from several qualitatively different traces,

hence provide a good platform for researchers to evaluate the protocols and services they propose for various environments. The theoretical tractability, on the other hand, assists system operators to make management decisions about a given protocol operated under an environment described by the TVC model.

In the first application, we show that the TVC model provides a general framework to model a wide range of mobility scenarios, and provides a powerful tool for simulation-based protocol or service evaluations in MANETs. We have made our mobility trace generator available at [9]. The tool provides mobility traces in both ns-2[88] compatible format and time-location (i.e., (t, x, y)) format.

In this section, our aim is twofold: (i) first, we would like to demonstrate the model's flexibility and how it can be configured to generate mobility instances that are representative of various target wireless networks such as WLANs, VANETs, etc.; (ii) at the same time, we would like to validate the model's "realism" or "accuracy" by explicitly comparing mobility instances produced by our model with real mobility instances captured in well-known, publicly-available traces. However, it is important to note that the use of such a model is not merely to match it with any specific trace instance available; this is only done for validation and calibration purposes. Rather, the goal is to be able to reproduce a much larger range of realistic mobility instances than a single trace can provide.

We first outline here some general guidelines about how to use the model in order to construct specific mobility scenarios. Then, we show how to explicitly configure the TVC model in order to match the mobility characteristics observed in three case studies: a wireless LAN trace, a vehicular trace, and a human-encounter trace. All the parameter values we use in the examples in this section are also available at [9].

STEP 1: Determine the structure in space and time: The first step to construct the TVC model for a given scenario is to setup the communities and the time period structure. If the map of the target environment is available, one should observe the map

and identify the points of attractions in the given environment and how they vary with respect to time (e.g., restaurants on campus during lunch time, hotels in an amusement park during nights), and assign the communities/time periods in the TVC model accordingly. If the map is not available, alternatively, one could use the general mobility characteristics observed in typical traces for the particular target network (as shown, for example, in Fig. 4-10) as guidelines to assign the structure in space and time. For example, from location preference curves like the one in Fig. 4-10(a), one can determine the number of communities one needs to explicitly create; as a very simple example, if in most WLAN traces it is observed that the typical node spends say 95% of its time at around 2 to 5 preferred locations (depending on the node), then one could assign each node to have from 2 to 5 local communities in the network (with the actual number and locations of communities randomly chosen for each node), with a larger (roaming) community representing the rest of the 5% of mobility time[12] . Similarly, from curves like the ones in Fig. 4-10(b), one may observe the *re-appearance* periodicity and decide on the time period structure accordingly. If a finer time granularity is necessary (e.g. time-of-day) one could additionally observe the mobility characteristics (e.g. location preferences) on an hour-by-hour basis and identify clear changes in a node's daily behavior.

STEP 2: Assign community-related parameters: Ideally, for a given environment, once the communities are identified, the related parameters (e.g., $\pi_j^t, \overline{D_j^t}, \overline{L_j^t}$, which represent the probability, average pause time, and average epoch length, at community j during time period t) could be assigned according to the mobile nodes' behavior in each community (e.g., how long does a typical person spend at the cafeteria for lunch?). Nevertheless, in most cases such information is not available, or extremely difficult to obtain. Hence, one could again resort to measured statistics from typical traces to guide

[12] In reality, one may be able to capture more complex structures with more communities or structure between them, by combining knowledge from the actual network area (map), generic mobility characteristics, and other information about the network.

the assignment of the parameters. It is not difficult to see that, typically, the attraction of the communities (π_j^t) and the time spent in each community (related to $\overline{D_j^t}$, $\overline{L_j^t}$) determine the shape of the location visiting preference curve. Thus, one can use basic probability theory to calculate the expected fraction of time a node spends in a given community as a function of these parameters, and derive from it the values needed to obtain a given location preference profile observed in a curve in Fig. 4-10(a). We calculate these community stay probabilities later in Lemma 4.1.

STEP 3: Assign user on-off behavior: Note that the mobility trace generated by the TVC model is an "always-on" mobility trajectory of the mobile node (i.e., the node is always present somewhere in the simulation field). Depending on the target environment, this always-on behavior may not be realistic. In many empirically collected traces, not all nodes are present all the time (i.e., some of the nodes are "off" or not in the observed area sometimes). This is the case in two of the scenarios we discuss below - in the WLAN traces, nodes are "on" only when they are not moving; in the vehicle mobility trace, nodes are "on" when they are moving. Thus, before producing the final synthetic trace, assumptions about when the user is considered "on" should sometimes be made and superimposed to the TVC mobility traces generated. We have applied this step to the traces of the two scenarios mentioned above.

Next, we look into three specific case studies, namely a set of WLAN traces, a vehicular trace, and a trace of inter-node encounters. We show how to apply the fore-mentioned procedure in each case, and show that synthetic mobility traces produced by the TVC model successfully match the characteristics observed in the real traces.

4.2.5.1 Matching mobility characteristics with WLAN traces

In the first example, we show that the TVC model can re-create the *location preferences* and *re-appearance probability* curves observed in WLANs. We construct our synthetic trace from the TVC model with the following steps: (STEP1) We divide the simulation area into a 10-by-10 grid and use these 100 grid cells as the locations for the

purpose of measuring mobility statistics for the simulated nodes. For each node, we assign some of the grids to serve as the node's communities during each time period, according to the method described earlier. (STEP2) We use the mobility characteristics obtained from the WLAN traces (i.e., curves in Fig. 4-10(a)) to calculate the attraction from the communities (π_j^t) and the pause times of the node ($\overline{D_j^t}$) to shape a decaying tail of *location visiting preference*. (STEP3) Since devices are usually turned off when users move them in the real WLANs, we make a similar assumption that the mobile nodes are considered "on" only when they are not moving. When the simulated node moves, we assume that it is not associated with the grid. Note that the curves in Fig. 4-10(b) represent the probability of an "on" node associated with the same community after the given time gap, and the peaks appear when the considered points in time are in the same type of time period. Therefore, the peak value is $\sum_{j=1}^{S^t}(\pi_j^t)^2(P_{on,j}^t)^2$, where $P_{on,j}^t$ denotes the probability a node is considered "on". Hence, the fraction of time nodes spend on moving ($\overline{L_j^t}/\overline{v}$) and pause ($\overline{D_j^t}$) can be adjusted (note that in this case, $P_{on,j}^t = \overline{D_j^t}/(\overline{D_j^t} + \overline{L_j^t}/\overline{v})$) to change the peak values in the curve of *periodical re-appearance* property to match with the curves in Fig. 4-10(b).

We use the MIT WLAN trace[82] as an example to display the match between the synthetic trace derived from the TVC model and the real trace. We also achieved good matching with the USC[80] or the Dartmouth[81] traces, but do not show it here due to space limitations (see [9]). We show the *skewed location visiting preferences* and the *periodical re-appearance* properties in Fig. 4-18 (a) and (b), respectively. We first try a simple synthetic model (labeled as *model-simplified*, using the parameters of *Model-1* in Table 4-2) with one community in two time periods. While this simple model captures the major trends in the mobility characteristics, there are several noticeable differences. First, since there is only one community, the tail in the *model-simplified* curve in Fig. 4-18(a) is "flat" as opposed to the exponentially diminishing tail of the *MIT* curve (notice the Y-axis in Fig. 4-18(a) is in log scale). Second, the peaks in the *model-simplified* curve in Fig.

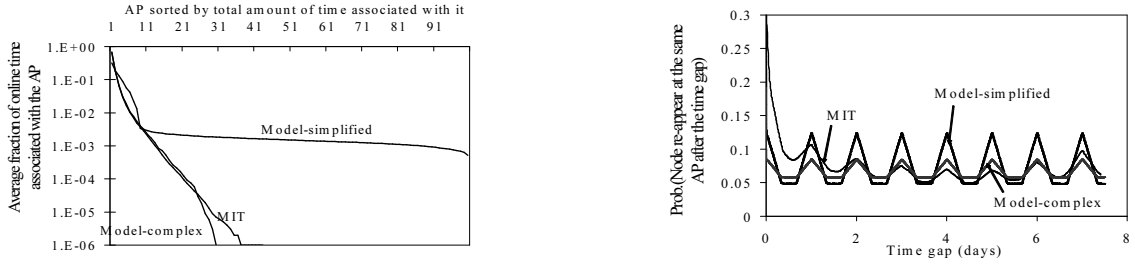
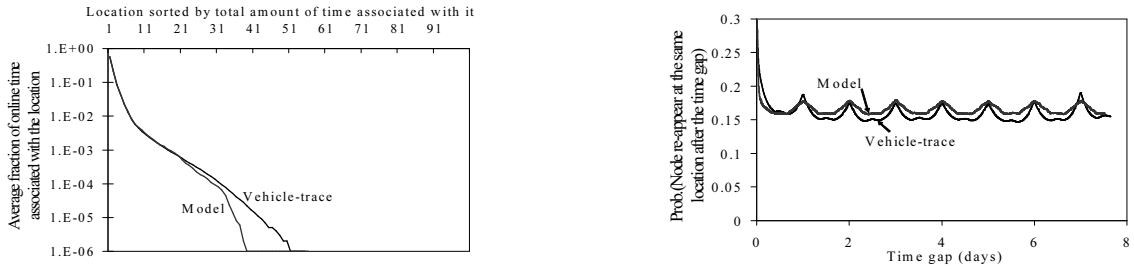


Figure 4-18. Matching the MIT WLAN trace with the synthetic trace. (a) Skewed location visiting preferences. (b) Periodical re-appearance at the same location.

4-18(b) are of equal heights, due to the simple two-alternating-time-period structure, as opposed to varying peak values of the *MIT* curve. We can improve the matching between the synthetic trace and the real trace by adding complexity in both space (using more communities) and time (using more complex schedule, such as the weekly schedule shown in Fig. 4-12). In a refined model labeled as *Model-complex* in Fig. 4-18, we show that the resulting mobility characteristics match very closely with the MIT trace. This also demonstrates the flexibility of our model - the user can adjust its complexity by choosing the number of communities and time periods needed to achieve a desired level of matching with the mobility characteristics.

4.2.5.2 Matching mobility characteristics with vehicle mobility traces

In this example we display that *skewed location visiting preferences* and *periodical re-appearance* are also prominent mobility properties in vehicle mobility traces. We obtain a vehicle movement trace from [17], a website that tracks participating taxis in the greater San Francisco area. We process a 40-day trace obtained between Sep. 22, 2006 and Nov. 1, 2006 for 549 taxis. We obtain the mobility characteristics of the taxis by the following steps. For each taxi, we first identify its movement range within the 40-day period, then draw a rectangular area that bounds the movement of the taxi, and divide this area into equal-sized 10-by-10 grids. We tally the mobility statistics of the taxis using these 100 grids as locations, and show the results in Fig. 4-19 (a) and (b), respectively, with the



(a) (b)

Figure 4-19. Matching the vehicle mobility trace with the synthetic trace. (a) Skewed location visiting preferences. (b) Periodical re-appearance at the same location.

label *Vehicle-trace*. It is interesting to observe that the trend of vehicular movements is very similar to that of WLAN users in terms of these two properties.

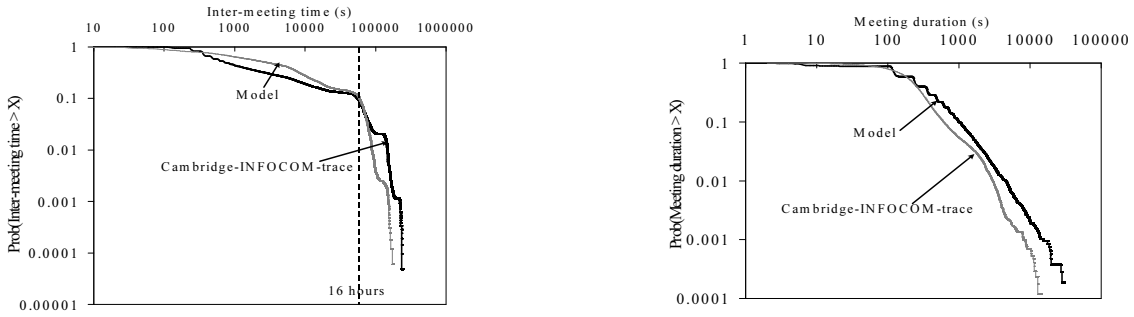
We further show that, using the outlined procedures, we can generate a synthetic trace with similar mobility characteristics as the vehicle mobility trace. After observing the trace closely, we discover that the taxis are offline (i.e., not reporting their locations) when not in operation. Hence in the synthetic trace we make the corresponding assumption (in STEP3) that the nodes are associated with the current grid they reside in only when they are moving; we then consider the pause times as breaks in the taxi operation (hence $P^t_{on,j} = (\overline{L^t_j}/\overline{v})/(\overline{D^t_j} + \overline{L^t_j}/\overline{v})$ in this case), from which we can calculate or adjust the respective model parameters. The curves in Fig. 4-19 with label *Model* correspond to the mobility characteristics of the synthetic trace. As a final note, although vehicular movements are generally constrained by streets and our TVC model does not capture such microscopic behaviors, designated paths and other constraints could still be added in the model's map (for vehicular or human mobility) without losing its basic properties. We defer this for future work.

4.2.5.3 Matching contact characteristics with encounter-based traces

In this example, we show that the TVC model is generic enough to also reproduce the distributions of the inter-meeting time and the encounter duration observed from a human encounter trace [84], by setting up its parameters properly. Specifically, we

tune our mobility model to mimic the behaviors observed in an experiment performed at INFOCOM 2005 [27]. In this experiment, wireless devices were distributed to 41 participants of the conference, with appropriate software installed that could log encounters between nodes (i.e. coming within Bluetooth communication range), as they moved around the premises of the conference area.

The inter-meeting time and the encounter duration distributions of all 820 pairs of users obtained from this trace are shown in Fig. 4-20 with label *Cambridge-INFOCOM-trace*. To mimic such behaviors using our TVC model, we observe the conference schedule at INFOCOM, and set up a daily recurrent schedule with five different types of time periods (STEP 1): technical sessions, coffee breaks, breakfast/lunch time, evening, and late night (see [9] for the detailed parameters). For each time period we set up communities as the conference rooms, the dining room, etc. We also generate a community that is far away from the rest of the communities for each node and make the node sometimes isolated in this community to mimic the behavior of patrons skipping part of the conference. It is interesting to note that the inter-meeting time distribution has a sharp drop (the "knee" in the curve) at 16 hours, which is approximately the time gap between the end of the day and the beginning of the subsequent day at the conference. This suggests the nodes (naturally) meet with lower probability during the nights, and thus the time-dependent mobility provided by our TVC model is appropriate. We can naturally achieve this by assigning nodes to disjoint communities (i.e., the "hotel rooms") during the nights. In STEP2, we use the theory presented in section 4.2.3 to adjust the parameters and shape the inter-meeting time and encounter duration curves. For example, a stronger tendency for nodes to choose roaming epochs (setting larger π_r^t) would increase the meeting probability (see, e.g., Eq. (4–15)), hence reducing inter-meeting times. Since the devices used to collect the encounter traces are always-on, we do not apply any changes to the synthetic trace in STEP 3. We randomly generate 820 pairs of users and obtain their corresponding distributions of the inter-meeting time and the encounter



(a) (b)

Figure 4-20. Matching *inter-meeting time* and *encounter duration* distributions with the human encounter trace. (a) Inter-meeting Time. (b) Encounter duration.

duration. These distributions are shown in Fig. 4-20 with label *Model*. It is clear that our TVC model has the capability to reproduce the observed distributions, even if it is not constructed explicitly to do so. This displays its success in capturing the decisive factors of typical human mobility.

It is clear from the cases studied here that, once we observe the target environment closely and come up with the right underlying parameters, the TVC model is able to capture the consequent mobility characteristics well. In addition, with the respective configuration, it is possible to generate synthetic traces with much larger scale (i.e., more nodes) than the empirical ones while maintaining the same mobility characteristics. It is also possible to generate multiple instances of the synthetic traces with the same mobility characteristics to complement the original, empirically collected trace. Although other proposed models have also managed to match some sets of collected measurements [62–65], none of the existing works has been shown to capture the variety of qualitatively different traces (e.g. WLAN, vehicles, inter-contacts) that the TVC model does.

4.2.6 Application II: Using Theory for Performance Prediction

Although the various theoretical quantities derived for the TVC model in Section 4.2.3 are interesting in their own merit, they are particularly useful in predicting protocol performance, which in turn can guide the decisions of system operation. We illustrate this point with two examples in this section.

4.2.6.1 Estimation of the number of nodes needed for geographic routing

It has been shown in geographic routing that the average node degree determines the success rate of messages delivered [87]. Thus, using the results of Section 4.2.3.2 we can estimate the number of nodes (as a function of the average node degree) needed to achieve a target performance for geographic routing, for a given scenario.

We consider the same setup as in Section 4.2.4.2, where half of the nodes are assigned to a community centered at $(300, 300)$ and the other half are assigned to another community centered at $(700, 700)$. We are interested in routing messages across one of the communities, from coordinate $(250, 250)$ to coordinate $(350, 350)$ with simple geographic routing (i.e., greedy forwarding only, without face routing [89]). Using simulations we obtain the success rate of geographic routing under various communication ranges when 200 nodes move according to the mobility parameters of *Model-1* (Table 4-2). Results are shown in Fig. 4-21 (each point is the percentage of success out of 2000 trials). If we assume now that the mobility model was different, say *Model-3*, we would like to know how many nodes we would need to achieve similar performance. Using Eq. (4–6) we find that 760 nodes are needed to create a similar average node degree for *Model-3*. To validate this, we also simulate geographic routing for a scenario where 760 nodes follow *Model-3*. Comparing the resulting message delivery ratio for this scenario to the original scenario (200 nodes with *Model-1*) in Fig. 4-21, we see that similar success rates are achieved in both scenarios under the same transmission range, which confirms the accuracy of our analysis.

4.2.6.2 Predicting message delivery delay with epidemic routing

Epidemic routing is a simple and popular protocol that has been proposed for networks where connectivity is only intermittent (often referred to as Delay Tolerant Networks) [71]. It has been shown that message propagation under epidemic routing can be modeled with sufficient accuracy (assuming the number of nodes is large enough) using a simple fluid-based model [91]. (Note that its performance has also been analyzed

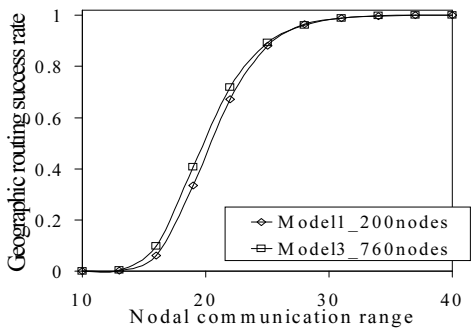


Figure 4-21. Geographic routing success rate under different mobility parameter sets and node numbers.

using Markov Chain [33, 92] and Random Walk [56] models.) This fluid model has been borrowed from the Mathematical Biology community, and is usually referred to as the SI (Susceptible-Infected) epidemic model. The gist of the SI model is that the rate by which the number of "infected" nodes increases ("infected" nodes here are nodes who have received a copy of the message) can be approximated by the product of three quantities: the number of already infected nodes, the number of susceptible (not yet infected) nodes, and the pair-wise contact rate, β (the implicit assumption there of course being that nodes meet independently). This contact rate in the SI model is equivalent to the unit-step meeting probabilities calculated in Section 4.2.3.4. Thus, one could in essence plug-in these meeting probabilities into the SI model equations and calculate the delay for epidemic routing. Yet, in the TVC model (and often in real life) there are multiple groups of nodes with different communities, and thus different pair-wise contact rates that depend on the community setup. For example, nodes with the same or overlapping communities tend to meet much more often than nodes in far away communities. For this reason, we extend the basic SI model to a more general scenario that is applicable to the TVC model.

We consider the following setup in the case study: We use *Model-3* (Table 4-2) for the mobility parameters. A total of $M = 50$ nodes are divided into two groups of 25 nodes each. One group has its community centered at $(300, 300)$ and the other at $(700, 700)$. One packet starts from a randomly picked source node and the time needed until it reaches all

other nodes in the network using epidemic routing is calculated. The propagation of the message can be described by the following equations:

$$\begin{cases} \frac{dI_1(t)}{dt} = \beta_{ov} I_1(t) S_1(t) + \beta_{no\_ov} I_2(t) S_1(t) \\ \frac{dI_2(t)}{dt} = \beta_{ov} I_2(t) S_2(t) + \beta_{no\_ov} I_1(t) S_2(t) \\ S_1(t) + I_1(t) = M/2 \\ S_2(t) + I_2(t) = M/2. \end{cases} \tag{4–21}$$

where $S_x(t)$ and $I_x(t)$ denote the number of susceptible and infected nodes at time t in group x, respectively. Parameters β_{ov} and β_{no_ov} represent the pair-wise unit-time meeting probability when the communities are overlapped (i.e., for nodes in the same group) and not overlapped (i.e., nodes in different groups), respectively. We use Eq. (4–15) to obtain these quantities. This model is an extension from the standard SI model [91] and similar extensions can be made for more than two groups [90]. The first equation governs the change of infected nodes in the first group. Notice that the infection to susceptible nodes in the group ($S_1(t)$) can come from the infected nodes in the same group ($I_1(t)$) or the other group ($I_2(t)$). We can solve the system of equations in (4–21) to get the evolution of the total infected nodes in the network. As can be seen in Fig. 4-22, the *theory* curve closely follows the trend in *simulation* curve (the non-perfect matching between the two curves, is due to the fact that fluid models become more accurate approximations of the actual stochastic spreading for large numbers of nodes). This indicates first that scenarios generated by our mobility model are still amenable to fluid model based mathematical analysis (SI), despite the increased complexity introduced by the concept of communities. It also shows that results produced thus can be used by a system designer to predict how fast messages propagate in a given network environment. This might, for example, determine if extra nodes are needed in a wireless content distribution network to speed up message dissemination.

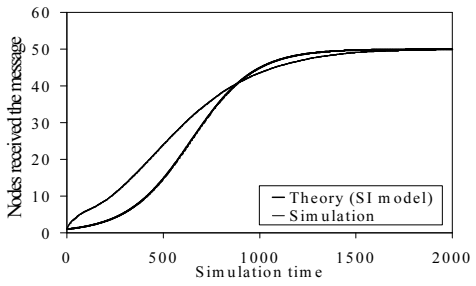


Figure 4-22. Packet propagation with epidemic routing. The total population is divided into 2 groups with different community.

As a final note, in addition to the epidemic routing, the theoretical results for the hitting and meeting times could be applied to predict the delay of various other DTN routing protocols (see e.g. [52, 56, 91]), for a large range of mobility scenarios that can be captured by the TVC model.

4.2.7 Conclusions and Future Work

Our contributions: We have proposed a *time-variant community mobility model* for wireless mobile networks. Our model preserves common mobility characteristics, namely *skewed location visiting preferences* and *periodical re-appearance at the same location*. We have tuned the TVC model to match with the mobility characteristics of various traces (WLAN traces, a vehicle mobility trace, and an encounter trace of moving human beings), displaying its flexibility and generality. A mobility trace generator of our model is available at [9]. In addition to providing realistic mobility patterns, the TVC model can be mathematically analyzed to derive several quantities of interest: the *nodal spatial distribution*, the *average node degree*, the *hitting time* and the *meeting time*. Through extensive simulation studies, we have verified the accuracy of our theory.

The TVC model can be easily generalized to provide scenarios in which nodes display more heterogeneous behavior. Nodes may have different set of parameters and even the time period structure can be different for different nodes. With these extensions, we have a mobility model to describe an environment including users with diverse mobility characteristics. We believe such a model is a very important building block

for understanding protocol performances in real-life settings. To demonstrate this last point, we also provide some examples of how our theory can be used in practice to predict protocol performance and guide design decisions.

In the future we would like to further analyze the performance of various routing protocols (e.g., [56, 57]) under the time-variant community mobility model. We also would like to construct a systematic way to automatically generate the configuration files, such that the communities and time periods of nodes are set to capture the inter-node encounter properties we observe in various traces (for example, the Small World encounter patterns observed in WLAN traces [60], see chapter 6 for the details).

CHAPTER 5
CASE STUDY II: MINING BEHAVIORAL GROUPS IN THE TRACES

After we have analyzed the individual user mobility in the last chapter, in this chapter we widen the scope of analysis to consider the relationship between multiple users. Specifically, we seek to answer a different question: given a WLAN trace, how can we identify users with similar behaviors from the trace? We rely on unsupervised learning techniques (i.e., clustering) as our main tool to investigate this question, and establish a systematic method to identify the underlying groups of similar users from large-scale user traces. We define a metric for similarity between user behaviors in the process, and we further leverage the similarity metric to guide message delivery in a new service paradigm – *profile-cast*, which delivers messages to groups implicitly identified by the behavior of the nodes.

5.1 Introduction

In recent years, we have witnessed the mass deployments of portable computing and communication devices (e.g., cellphones, laptops, PDAs) and wireless communication infrastructures. As the adoption of these technologies becomes an inseparable part of our lives, we expect fundamental behavioral change among the users. To estimate the impact of this new paradigm shift in user behavior, it is not sufficient to study the change exclusively from a technology perspective. There is a pressing need to capture and understand the user behavioral patterns when these new technologies are adopted. This understanding will also play a crucial role in solving a multitude of technical issues, ranging from better network management to designing of behavior-aware protocols, services, and user models.

Consider wireless LANs (WLANs) on university campuses as an example. One could imagine the major work places (e.g., offices and classrooms) and the informational hubs (e.g., libraries and computer centers) would dominate users' behavioral patterns in terms of network usage (in terms of the locations they access the network from). However, as the

WLAN deployments prevail, the location from where people access information may start to change. While the traditional "hot spots" still play an important role, we can expect users to display diverse behavioral patterns that reflect their personal preferences (e.g., A small group may prefer to work at a coffee shop), as these wireless devices become tiny and personalized. We need to understand such behavioral patterns to better characterize the users within a social context. The technique to discover such patterns from collected data is the focus of this section. This is very different from the overall user mobility statistics we look into in chapter 4, as we previously do not seek to distinguish users as similar or dissimilar based on their behavioral patterns, which we will do in this chapter.

Specifically, we take a first step toward understanding and characterizing the structure of behavioral patterns of users within large WLANs. We develop methods to identify groups of users that demonstrate similar and coherent behavioral pattern. This is important for several reasons: (1) From the application or service perspective, the groups identify different existing major behavioral modes in the network, and, hence, can be potentially utilized to identify targets for group-aware services. (2) From the network management perspective, it helps us to understand the potential interplay of the user groups with the network operation and reveals insight previously unavailable by looking at the mere aggregate network statistics. (3) From a social sciences perspective, the results unravel the relationships between users (i.e., their "closeness" in terms of network usage behavior) when they embrace a new lifestyle.

We apply our analysis framework on long-term WLAN traces obtained from two university campuses[80, 81] across the coasts of USA. We represent a user's behavioral features by constructing a normalized association matrix to which we apply our analysis. While the applicability of our methods is not specific to WLANs, these are the most extensive wireless user behavioral traces available today. We leverage unsupervised learning (i.e., clustering) techniques [95] to determine groups of users displaying similar behavior. While clustering has been widely-applied in other areas and in some

cases[24, 38] to WLAN traces, the main contribution of the study is to construct proper representations for our data sets and design novel distance metrics between users. These two aspects are fundamental in the application of clustering techniques and determine the quality of the results we obtain. The key challenge in designing a good distance metric is to accurately and succinctly summarize the trends in the data, so the distances are not influenced by noise and can be evaluated efficiently. We show that a singular-value decomposition (SVD) based scheme not only provides the best summary of the data, but also leads to a distance metric that is robust to noise and is computationally efficient. The succinct summaries also help to reduce the processing, storage, and exchange (i.e., when nodes communicate with each other to convey their behavioral summaries) overhead. Furthermore, we validate our methods and explain its significance.

We leverage our *TRACE* approach (outlined in the chapter 1) to understand user grouping in this study. Specifically, the work starts with the WLAN *T*races that capture realistic user behavior. We then focus on a specific *R*epresentation distilled from the traces that captures important aspects of user behavior, as we introduce in section 5.2. We then *A*nalyze the *clustering* of the users based on the chosen representation, *normalized association vectors*, from section 5.3 to section 5.6. We first show the need for a good distance metric for clustering in section 5.3. To achieve that goal, we conduct further analysis to understand the nature of user association patterns, and evaluate and contrast various summaries to capture its major trend in section 5.4. We then utilize a feature-based approach to achieve meaningful user clustering in section 5.5 and discuss its interpretation in section 5.6, where we show the *C*haracteristics of user groups we observed from the traces. We briefly discuss how to *E*mploy the of the methods and findings we develop in the user-clustering effort in section 5.7, and take one application, the behavior-aware message delivery, as the major focus in section 5.8 to show the usefulness of the understanding of user grouping and our similarity metric.

5.2 Preliminaries

We first introduce the traces we analyze in the study and the *normalized association vector* representation we choose. We also briefly introduce the necessary background knowledge about clustering in the section.

5.2.1 Choice of Data Sets and Representations

The widespread deployments of large-scale wireless LANs on university campuses have attracted high adoption from its community. These deployments have outgrown experimental networks and become commodities. Due to its high penetration and diversity in users (as compared to corporate WLANs), campus networks are good platforms to study the behavioral pattern of WLAN users. We elect two WLAN traces collected from large populations for long durations for the study (namely, the semester-long *USC-06spring* and quarter-long *Dart-04spring* traces). The details for the selected traces are listed in Table 3-1. While the devices logged in the WLAN traces are mainly laptop computers, we note that our methods are not limited to the specific data sets we choose, and it would be of great interest to study traces from other mobile devices (e.g., cellphones, iPods), if available for a large population.

To understand user behavior from wireless network traces, the first fundamental task is to choose a representation of the raw data. This chosen representation should have significance to the network and in the greater social relationship context. We choose the patterns of users visiting WLAN access points (APs) for the analysis. Visiting pattern is important to WLANs as mobility is one of its defining characteristics. When a WLAN user moves within the campus and *associates* with APs across the network, the set of APs with which the user associates is considered an indicator of the user's physical location. From a social context, the places a person visits regularly and repeatedly usually have a stronger connection to her identity and affiliation. It is perhaps one of the important distinguishing factors for people with different social attributes.

We represent a user's visiting pattern by what we refer to as the *normalized association vectors*[1] . The association vector is a summary of a user's AP association during a given time slot. Note that there are potentially many ways to represent user behavior from a rich data set. Different representations certainly provide different insights. We focus first on the *normalized* representation for *daily association vectors* to illustrate our analysis, and briefly discuss about other alternatives in section 5.10. We choose to use a day as the time slot since it represents the most natural behavior cycle in our lives. The *association vector* for each time slot is an n-entry vector, $(x_1, x_2, ..., x_n)$, where n is the number of unique locations (i.e., buildings[2]) in the given trace. Each entry in the vector, x_i, represents the *fraction* of online time the user spends at the location during the time slot, i.e., we normalize the user association time with respect to his online time in the considered time slot. With this representation, the conclusions we draw are not influenced by the absolute value of online time, which varies across a wide range among different users and different time slots of a given user. Note that the sum of the entries in the association vector, $\sum_{i=1}^{n} x_i$, is always 1 if the user has been online during the time slot. We use a zero vector to represent the association vector when the user is completely offline for the time slot. To represent a user's association preference for the long run, we construct the *association matrix* X for the user, as illustrated in Fig. 5-1, i.e., we concatenate the association vectors for each time slot (day). If there are n distinct locations and the trace period consists t time slots, the *association matrix* for a user is a t-by-n matrix.

[1] For brevity, we sometimes use the shortened term *association vector* to refer to the *normalized association vector* unless stated otherwise.

[2] We aggregate APs in the same building as a single location for better interpretation of user behavior.

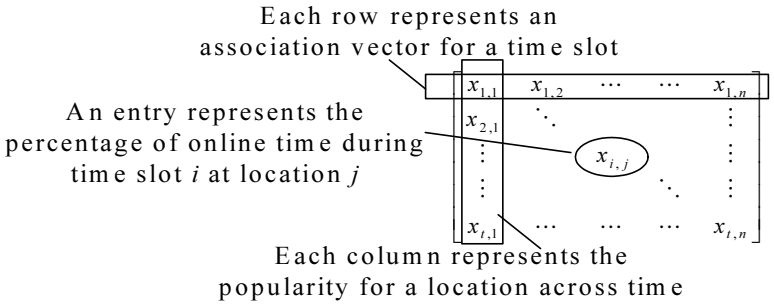


Figure 5-1. Illustration of association matrix representation.

5.2.2 Preliminaries of Clustering Techniques

Clustering (one of the key methods in unsupervised learning) is a widely-applied technique to discover patterns from data sets with unknown characteristics. It can be roughly classified into hierarchical or partitional schemes [95]. We use the hierarchical clustering, in which each element is initially considered as a cluster containing one member. Then, at each step, based on the distances between the clusters[3] , two clusters that are the closest to each other among all cluster pairs are merged into one cluster with larger membership. This process continues until a *clustering threshold* has been reached, when all the inter-cluster distances for the remaining clusters are larger than a given distance threshold, or the remaining cluster number reaches a given target.

One major issue in applying clustering to a data set with unknown characteristics is that it is hard to pre-select a proper clustering threshold in advance. The indication of a good clustering result is that the distances between elements belonged to the same cluster are low, and the distances between elements in different clusters are high. (i.e., there is a clear separation between inter-cluster and intra-cluster element distance distributions.) Usually the clustering threshold comes from the domain knowledge or trial-and-error.

[3] Among several alternatives, we use the average distance of all element pairs between the clusters. Use of other methods does not change the results significantly.

Often the decisive factor for the quality of the clustering results is the selection of the *distance metric*, which is our main contribution as we show in the following sections.

5.3 Challenges

As mentioned previously, the most important step in clustering is to define the *similarity* or *distance metric* between users[4] . We highlight the challenges in selecting a proper distance metric with an example in this section.

An intuitive distance function between user association patterns of two individuals is to consider all the association vector pairs. Formally, we define the *average minimum vector distance (AMVD)* between users A and B, $AMVD(A, B)$, as

$$AMVD(A, B) = \frac{1}{|A|} \sum_{\forall A_i \in A} \arg \min_{\forall B_j \in B} d(A_i, B_j), \tag{5–1}$$

where A_i and B_j denote an association vector of user A and B, respectively. $|A|$ denotes the cardinality of set A. $d(A_i, B_j)$ denotes the Manhattan distance, defined as[5]

$$d(a, b) = \sum_{i=1}^{n} |a_i - b_i|, \tag{5–2}$$

where a_i and b_i are the i-th element in vector a and b, respectively. $AMVD(A, B)$ is the average of distances from each of the vectors in set A to the closest vector (or the nearest neighbor) in set B. Note that, with this definition, $AMVD(A, B)$ is not necessarily equal to $AMVD(B, A)$. We define a symmetric distance metric between users A and B as $D(A, B) = (AMVD(A, B) + AMVD(B, A))/2$.

We apply the hierarchical clustering algorithm to users with the distance metric derived from $AMVD$. As mentioned earlier, a clustering algorithm requires properly

[4] $d(x, y)$ is a distance function if $d(x, x) = 0$ and $d(x, y)$ is small if x and y are similar and large otherwise. Similarity can be considered to be the opposite of distance i.e. $sim(x, y) = 0$ means x,y are dissimilar.

[5] We use Manhattan distance, or the $L1$ norm, since it is robust to statistical noise. Note that by our representation, $0 \leq d(a, b) \leq 2$ for normalized association vectors a and b.

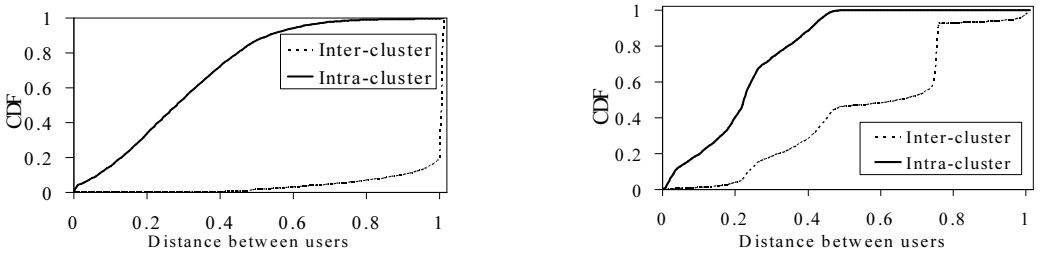


Figure 5-2. Cumulative distribution function of distances for inter-cluster and intra-cluster user pairs (AMVD distance). (a) USC. (b) Dartmouth.

chosen thresholds, and the particular choice is data-dependent. We experiment with various thresholds, and discover that for the USC trace, we can group the populations into 200 clusters with a clear separation between inter and intra cluster distance distributions (Fig. 5-2 (a)), which is a qualitative indicator for a tight clustering. However, the distance metric works poorly for the Dartmouth trace, as shown in Fig. 5-2 (b). The separation between inter and intra cluster distance distributions is not clear, *regardless* of the cluster thresholds we use (we have tried several).

One problem with the $AMVD$ metric is that it considers all association vectors, i.e., it includes not only the important trends, but also the noise vectors when the users deviate from the dominant trend, leading to bad clustering results. A meaningful distance metric should capture the major trends of user behavior and be robust to noise and outliers. Another problem of the $AMVD$ metric is its computational complexity. We have to calculate the distances between all t^2 pairs of association vectors for each user pair. If there are N users the computation requirement is of order $O(N^2t^2)$. Furthermore, it requires significant space to store t association vectors for all N users. Thus we would like to design a metric that is both (1) robust to noise and (2) computation and storage efficient. In order to achieve both goals, we start by studying the characteristics of the association patterns of a single user to validate the repetitive patterns or modes of behavior. We show that this study leads us to the appropriate distance metric.

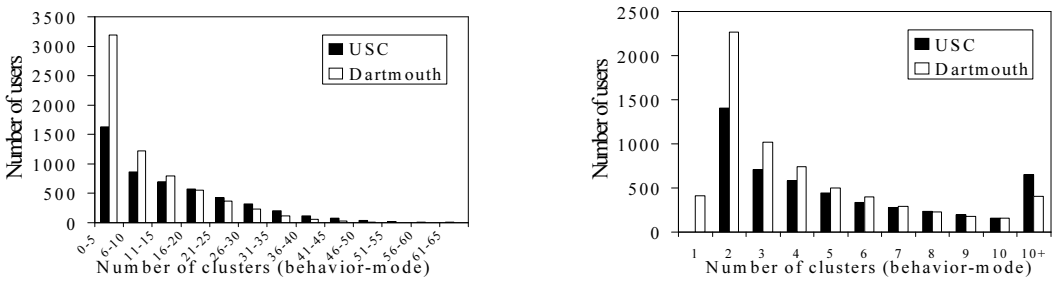


Figure 5-3. Distribution of number of clusters (behavioral modes) for users. (a) Clustering threshold = 0.2. (b) Clustering threshold = 0.9.

5.4 Summarizing the Association Patterns

In this section, we identify association trends of an individual and construct a compact representation of her association matrix, which is suitable for distance computations used in clustering.

5.4.1 Characteristics of Association Patterns

We first understand the repetitive trend in a single user's associations pattern, and how dominant the trend is (i.e., are there dominant *behavioral modes*?). We obtain this understanding by applying hierarchical clustering to all the association vectors of a single individual.

Consider the clustering of the association vectors, X_i for $i = 1, ..., t$ (i.e., row vectors of an association matrix X) of a single user. The identified clusters represent distinct *behavioral modes* of the user. Similar association vectors will be merged into a cluster and its size indicates its dominance – a large cluster with many association vectors implies that the user follows consistent association patterns in many time slots (in our case, *days*) as its major behavioral modes.

We apply hierarchical clustering to the association vectors of each user in the USC and the Dartmouth traces using various clustering thresholds. The distribution of number of clusters (or *behavioral modes*) obtained are shown in Fig. 5-3. In Fig. 5-3(a), we use a small clustering threshold (0.2), with which only very similar association vectors are

merged. We see that for the USC and the Dartmouth traces, respectively, about 50% and 67% of users have less than 10 different clusters or behavioral modes (much fewer than total number of time slots, 94 and 61) with this low clustering threshold. This indicates the users have distinct repetitive trends in its association vectors. On the other hand, if we consider a moderate clustering threshold (0.9), we see in Fig. 5-3 (b) that users still show multiple behavioral modes. On average, with 0.9 as the clustering threshold, the number of behavioral modes for USC and Dartmouth users are 5.57 and 4.32, respectively, and the users with the most behavioral modes have 32 clusters in both cases.

Most of those users with two behavioral modes have a consistent association pattern: One mode corresponds to the association vectors when the user is offline, and the other one corresponds to the association vectors when the user is online. These users switch between online and offline behaviors from day to day, and when they are online, the association vectors are consistent and fall in a single behavioral mode. We refer to these users as *single-modal* users. On the other hand, we also observe many *multi-modal users*. These users show a more complex behavior: their association vectors form more than two clusters, which indicate that they display distinct behavioral modes when they are online. 71.9% of users in USC and 59.4% of users in Dartmouth are classified as multi-modal when the clustering threshold is 0.9. Hence, we conclude that although users in WLANs are not extremely mobile, they do move and display various association patterns over a period of time.

To examine the degree of dominance of the most important behavior modes of users, we compare the most important behavioral mode and the second most important one (i.e., the largest and the second largest clusters) in terms of their sizes. In Fig. 5-4 we plot the size (i.e. number of vectors) distributions of the first and the second behavioral modes under clustering threshold 0.2 (solid lines) and 0.9 (dotted lines) for USC users. We see that there is a clear separation between the sizes of these two behavioral modes. (i.e., the most dominant behavioral mode is much more important than the second most

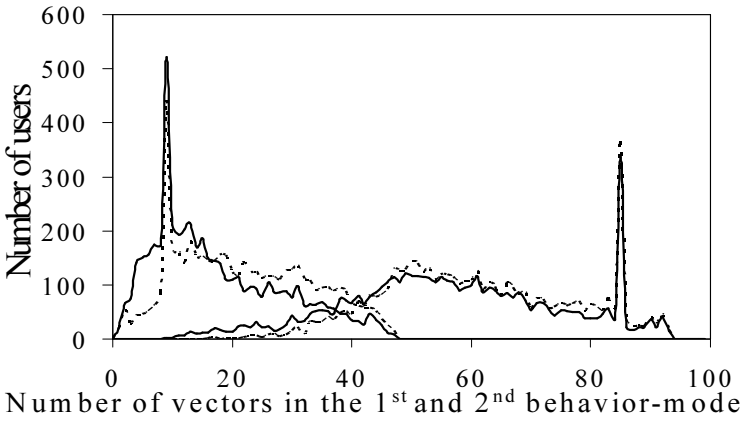


Figure 5-4. Distribution of association vectors in the first and the second behavioral modes for the USC trace. Right: the first cluster, Left: the second cluster.

important one for most users.) Different clustering thresholds do not change the results much. In other words, observations of the most dominant behavioral mode could reveal user characteristic to a good extent for many users. Similar observations also hold for the Dartmouth users.

We show the distribution of the size ratio between the largest and the second largest cluster in Fig. 5-5. Here we see for USC and Dartmouth, respectively, 36% and 31% of users have the two most important behavior modes with comparable sizes (i.e., with size ratio smaller than 2.0 - The second most important behavioral mode is followed at least one half as often as the most important behavioral mode). Hence looking at the most dominant cluster exclusively could still be sometimes misleading and we might be ignoring information about the user's detailed behavior. It is therefore desirable to have a summary that takes not only the dominant behavioral mode, but also the subsequent ones into account.

5.4.2 Summarization Methods

Now we investigate various ways to summarize the association vectors, and then judge their quality based on a specific metric – the *significance score*.

Average of association vectors: This is the simplest way to calculate a summary. Averaging naturally emphasizes the dominant behavioral mode (as there are more vectors

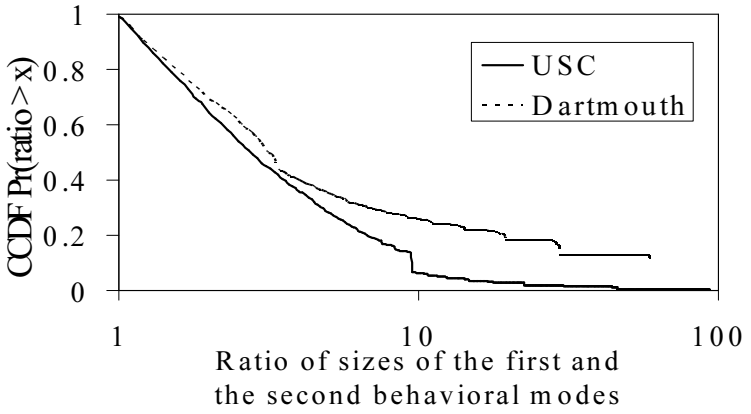


Figure 5-5. Complementary CDF for the ratio of the first behavioral mode size to the second behavioral mode size. Note that the X-axis is in log scale to make the graph more visible.

in this mode). As users are not always online, the average should include only the online days and ignore the zero vectors. It is defined as

$$X_{onavg} = \frac{\sum_{i=1}^{t} X_i}{\sum_{i=1}^{t} \parallel X_i \parallel_1}, \tag{5–3}$$

where $\parallel X_i \parallel_1$ is the L1 norm of vector X_i (recall that for online days, the elements in association vectors sum to 1, hence $\parallel X_i \parallel_1 = 1$ if the user is online for the time slot i or 0 if the user is offline.).

Centroid of the first cluster: We observe for many users, the first behavioral mode is dominant. Hence we can use the centroid of vectors in the first *non-trivial* behavioral mode (i.e., if the first behavioral mode is the cluster of zero vectors, we take the second behavioral mode instead) as a summary. Formally,

$$X_{centroid1} = \frac{\sum_{X_i \in C_1} X_i}{\sum_{i=1}^{t} I(X_i \in C_1)}, \tag{5–4}$$

where C_1 denotes the largest non-trivial behavioral mode for the user and $I(\cdot)$ is the indicator function. Intuitively, it works well if the first behavioral mode is dominant, but less so if there are multiple behavioral modes with comparable importance for the user. We experiment with two different clustering thresholds, 0.5 or 0.9, when we obtain the

Table 5-1. The average significance score for various summaries of user association vectors

	X_{onavg}	$X_{centroid1}$ threshold 0.5	$X_{centroid1}$ threshold 0.9	SVD
USC	0.646	0.716	0.702	0.764
Dartmouth	0.690	0.757	0.747	0.789

clusters of different behavioral modes from the association vectors of the user and identify the dominant behavioral mode.

In order to quantitatively compare the quality of the summary techniques, we propose to measure the *significance score* of a summary vector with respect to a user by summing the projections of all association vectors on the summary vector, normalized by the online days of the user.

$$SIG(Y) = \frac{\sum_{i=1}^{t} |X_i \cdot Y|}{\sum_{i=1}^{t} \parallel X_i \parallel_1}, \tag{5–5}$$

where Y is any summary vector. The physical interpretation of the *significance score* is the percentage of power in the association vectors X_i's explained by the summary vector Y. Following the definition, we calculate the average score of the *significance* for X_{onavg} and $X_{centroid1}$, and list them in Table 5-1. We observe that the centroid of the first cluster better explains the behavioral pattern of a given user than the average, since averaging sometimes lead to a vector that falls between the behavioral modes.

Singular value decomposition: We revisit our definition of the *significance score* in Eq. (5–5), and pose it as an optimization question: Given the association vectors X_i's, what is the best possible summary vector Y to maximize its significance? Mathematically, we want the vector Y to be

$$Y = \arg\max_{|v|=1} \sum_{i=1}^{t} |X_i \cdot v|. \tag{5–6}$$

This is exactly the procedure to obtain the first singular vector if we perform singular value decomposition (SVD) [41] to the association matrix X. In other words, if we want the summary vector Y to capture the maximum possible power in the association vector X_i's, the optimal solution is to apply singular value decomposition to extract the first singular vector of the association matrix X. We apply this technique and calculate the

significance score in the last column in Table 5-1. It is evident from the numbers that among all the candidates, SVD provides the best summary. Hence we focus on the use of the SVD-based summary, and defer the discussion of other summary techniques to section 5.10.

5.4.3 Interpreting Singular Value Decomposition

In this subsection we explain other important properties of SVD as applied to the association matrices.

From linear algebra [41], we know that for any t-by-n matrix X, it is possible to perform singular value decomposition, such that

$$X = U \cdot \Sigma \cdot V^T, \tag{5–7}$$

where U is a t-by-t matrix, Σ is a t-by-n matrix with r non-zero entries on its main diagonal, and V^T is an n-by-n matrix where the superscript ^{T} in V^T indicates the transpose operation to matrix V. r is the rank of the original association matrix X. The column vectors of the matrix V are the eigenvectors of the covariance matrix X^TX, and Σ is a diagonal matrix with the corresponding singular values to these eigenvectors on its diagonal, denoted as σ_1, σ_2, ..., σ_r. These singular values are ordered by their values (i.e. $\sigma_1 \geq \sigma_2 \geq ... \geq \sigma_r$). We can re-write Eq. (5–7) in a different form:

$$\tilde{X}_k = \sum_{i=1}^{k} u_i \sigma_i v_i^T. \tag{5–8}$$

Here u_i's and v_i's are the column vectors of matrix U and V. They are used as the building blocks to reconstruct the original matrix X. With this format, SVD can be viewed as a way to decompose a matrix: It breaks the matrix X into column vectors u_i, v_i and real numbers σ_i. If we retain all these components (i.e., $k = rank(X)$), SVD is a lossless operation and the matrix X can be reconstructed accurately. However, in practical applications, SVD can be treated as a lossy compression and only the important components are retained to give a rank-k approximation of the matrix X. The percentage

of power in the original matrix X captured in the rank-k reconstruction in Eq. (5–8) can be calculated by

$$\frac{\sum_{i=1}^{k} \sigma_i^2}{\sum_{i=1}^{Rank(X)} \sigma_i^2}. \tag{5–9}$$

For our data sets, the users have much fewer behavioral modes than the number of association vectors, and for most users the dominant behavioral modes are much stronger than the others (c.f. Fig. 5-4). Hence we expect SVD to achieve great data reduction on the association matrices. This is indeed the case, as we show in Fig. 5-6: Most of the users have a high percentage of power in their association matrices X explained by a relatively low-rank reconstruction – For example, in the USC trace (Fig. 5-6(a)), if we use a rank-1 reconstruction matrix, it captures 50% or more of power in the association matrices for more than 98% of users, and a rank-3 reconstruction is sufficient to capture more than 50% of power in the association matrices for all users. Even if we consider an extreme requirement, capturing 90% of the power in the association matrices, it is achievable for 68% of users using a rank-1 reconstruction matrix, and for more than 99% of users using at most a rank-7 reconstruction matrix. Similar observations can be made for the Dartmouth users (in Fig. 5-6(b)). For both campuses, five components are sufficient to capture 90% or more power for most (i.e., more than 90%) of the users. This indicates although users show multi-modal association pattern, for most users the top behavioral modes are relatively much more important then the remaining ones.

If a low-rank reconstruction of the association matrix is achievable, it is natural to ask for the representative vectors for the behavioral modes of a user. For this purpose, SVD can be viewed as a procedure to obtain representative vectors that capture the most

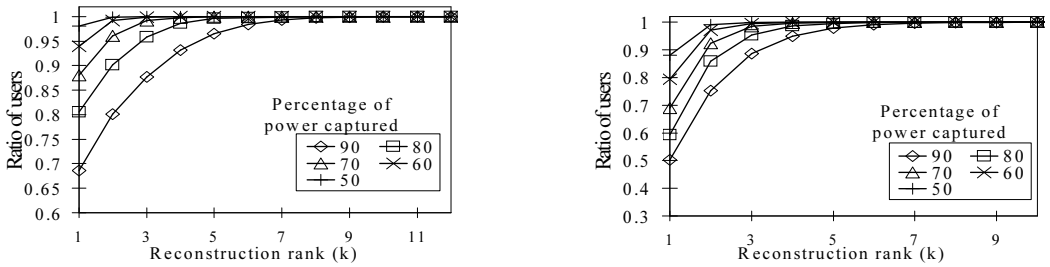


Figure 5-6. Low association matrices dimensionality: A high target percentage of power is captured with a low rank reconstruction matrix for many users. (a)USC. (b)Dartmouth.

remaining power in the matrix. Mathematically[6] ,

$$\begin{aligned} u_1 &= \arg\max_{\|u\|=1} \|X \cdot u\| \\ u_k &= \arg\max_{\|u\|=1} \|(X - \sum_{i=1}^{k-1} X u_i u_i')u\| \quad \forall k \geq 2. \end{aligned} \tag{5–10}$$

We can interpret the singular vectors, u_j's, as the vectors that describe the user's behavioral modes in decreasing order of importance in the association matrix X, with its relative weight (or the importance) quantified by $\sigma_j^2 / \sum_{i=1}^{r} \sigma_i^2$, following Eq. (5–9). We refer to these vectors as the *eigen-behavior* vectors for the user.

The *eigen-behavior* vectors, u_j's, are unit-length vectors. The absolute values of entries in an *eigen-behavior* vector quantify the relative importance of the locations in the user's j-th behavioral mode. For example, suppose a given user visits location l almost exclusively, then in his first eigen-behavior vector, the entry corresponds to location l would carry a high value (i.e. close to 1), and the weight of the first eigen-behavior vector, $\sigma_1^2 / \sum_{i=1}^{r} \sigma_i^2$, shall be high. With a set of *eigen-behavior* vectors and their corresponding weights, we can capture and quantify the relative importance of a user's behavioral modes.

[6] SVD on matrix X can be viewed as calculating the eigenvalues and eigenvectors of the covariance matrix, $X^T X$. This is also the procedure typically used to perform Principal Component Analysis (PCA) for matrix X.

There are several benefits of applying SVD to obtain the summary as compared to other schemes: (1) SVD provides the optimal summary that captures the most remaining power in the original matrix with each additional component. (2) The components can be used to reconstruct the original matrix, while the calculation of average or centroid vectors are non-reversible. Thus SVD provides a way to "compress" user association vectors and helps us save storage space. (3) Not only the most important behavioral mode, but also the subsequent ones can be systematically obtained with SVD, with a quantitative notion of their relative importance.

5.5 Clustering Users by Eigen-Behavior Vectors

In this section, we define our novel distance measure between WLAN users based on the *eigen-behavior* vectors and then use it for user clustering.

5.5.1 Eigen-Behavior Distance

Suppose u_i's and v_j's are the eigen-behavior vectors of two users, where $i = 1, ..., r_u$, and $j = 1, ..., r_v$. r_u and r_v are the ranks of the corresponding association matrices. The similarity between the two users can be calculated by the sum of pair-wise inner products of their eigen-behavior vectors u_i's and v_j's, weighted by w_{u_i} and w_{v_j}[7] . Our measure of similarity between two sets of eigen-behavior vectors, $U = \{u_1, ..., u_{r_u}\}$ and $V = \{v_1, ..., v_{r_v}\}$, is defined as:

$$Sim(U, V) = \sum_{i=1}^{r_u} \sum_{j=1}^{r_v} w_{u_i} w_{v_j} |u_i \cdot v_j|. \tag{5–11}$$

Higher similarity index $Sim(U, V)$ indicates that the eigen-behavior vectors U and V are more similar, and hence the corresponding users have similar association patterns. We

[7] w_{u_i} represents the weight of the eigen-behavior vector u_i, calculated by $w_{u_i} = \sigma_{u_i}^2 / \sum_{k=1}^{r_u} \sigma_{u_k}^2$. The weights w_{u_i}'s sum up to 1, and w_{v_j}'s are defined similarly.

define the *eigen-behavior distance* between users U and V as $D'(U,V) = 1 - (Sim(U,V) + Sim(V,U))/2$.[8]

Using the *eigen-behavior distance* also reduces the computation overhead. If we use only the top-5 components (which captures more than 90% power in the association matrices for most users, as shown in Fig. 5-6), instead of going through t-by-t pairs of original association vectors as we did in the $AMVD$ distance in section 5.3, we reduce the distance calculation to 5-by-5 vector pairs. Since we have at least 61 days in the traces, this is at least a $(61/5)^2 \approx 148$ fold saving for all N^2 pair of users. By paying the pre-processing (i.e., SVD for all N users) overhead of $O(Nt^2)$, we can reduce the distance calculation complexity from $O(N^2t^2)$ to $O(c \cdot N^2)$. Since users follow repetitive trends in the association patterns, its total *eigen-behavior* vectors would not grow with the number of time slots, t. If we consider longer traces or association vector representations in finer time scale, the reduction can be even more significant. In the following computations, we consider only the eigen-behavior vectors that capture at least 0.1% of total power in the user's association matrix.

5.5.2 Significance of the Clusters

We cluster users based on the eigen-behavior distance and again validate the results by plotting the intra-cluster and inter-cluster distance distributions, when we consider 200 clusters. With the eigen-behavior distance, for both the USC and Dartmouth traces, there is a better separation between the CDF curves (shown in Fig. 5-7) as compared to the results with the $AMVD$ distance (shown in Fig 5-2), indicating a meaningful clustering. This proves the eigen-behavior distance is a better metric than the $AMVD$ distance as it helps us to group users into well-separated behavioral groups based on their WLAN association preferences, for both campuses.

[8] We normalize the similarity indices from user U to all other users between $(0,1)$. Among all users, we find the user K such that $Sim(U,K) = \mathbf{max}_{\forall N} Sim(U,N)$. We than normalize $Sim(U,V) = Sim(U,V)/Sim(U,K)$ for all users V.

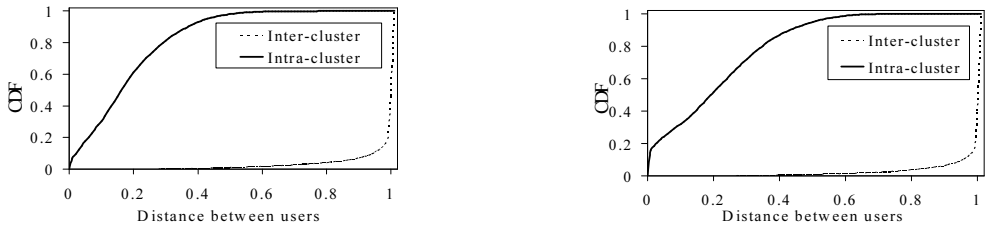


Figure 5-7. Cumulative distribution function of distances for inter-cluster and intra-cluster user pairs (eigen-vector distance). (a) USC. (b) Dartmouth.

We further validate whether the resulting clusters indeed capture users with similar behavioral trends. For this test, we construct the *joint association matrix* by concatenating the daily association vectors of a cluster of m similar users in a larger mt-by-n matrix, where n is the number of locations and t is the number of time slots. When we perform SVD to the *joint association matrix*, the top eigen-behavior vectors represent the dominant behavioral patterns within the group. If the users in the group follow a coherent behavioral trend, the percentage of power captured by the top eigen-behavior vectors should be high. On the other hand, if association vectors of users with different association trends are put in one *joint association matrix* (i.e., if dissimilar users are put in one cluster by mistake), the percentage of power captured by its top eigen-behavior vectors should be much lower. Among all clusters, we pick those with more than five users, and compare the cumulative power captured by the top four eigen-behavior vectors of these clusters with random clusters of the same size (i.e., we randomly pick the same number of users from the population and construct another *joint association matrix*) in scatter graphs, shown in Fig. 5-8. Clearly, most the dots are well above the 45-degree line for both campuses. This indicates the users in the same cluster follow a much stronger coherent behavioral trend than randomly picked users, pointing to the significance of our clustering results.

We would also like to see if each cluster from the population shows a distinct behavioral pattern. To quantify this, we obtain the first eigen-behavior vector from

(a) (b)

Figure 5-8. Scatter graph: Cumulative power captured in top four eigen-behavior vectors of random matrices (X) and the joint association matrices formed by users in the same cluster (Y). Only clusters with 5 or more members are included. (a) USC(129 clusters). (b) Dartmouth(136 clusters).

each group and calculate its *significance score*, defined in Eq. (5–5), for all the groups. The results confirm with our goal of identifying groups following different behavioral trend: For the USC trace, the first eigen-behavior vectors obtained from the *joint association matrices* have an average *significance score* of 0.779 for their own clusters and an average score of 0.005 for other clusters, indicating the dominant behavioral trends from each cluster is distinct. The corresponding numbers for the Dartmouth trace are 0.727 and 0.004, respectively.

We conclude that we have designed a distance metric that effectively partitions users into groups based on behavioral patterns. In addition, these clusters are unique with respect to their major behavioral trends.

5.6 Interpretation of the Clustering Results

In this section we analyze and interpret the results of clustering for both university campuses from social perspective.

First we analyze the group size distribution, as shown in Fig. 5-9. We observe the distributions of group sizes are highly-skewed for both campuses. There are dominant behavioral groups that many users follow: the largest groups in the campuses include 504 and 546 members, out of the population of 5000 for USC and 6582 for Dartmouth, respectively. The ten largest groups combined account for 39% and 33% of the total

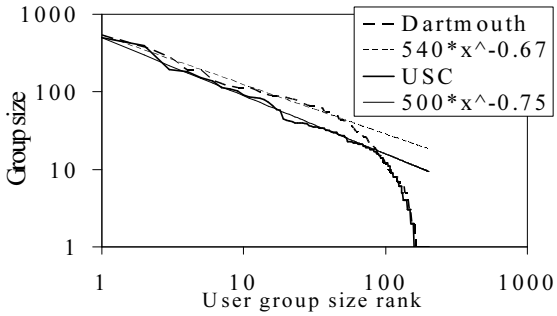


Figure 5-9. Rank plot (group size ranking v.s. group size) in log-log scale. User group size follows a power-law distribution.

population, respectively. On the other hand, there are also many small groups, or even singletons, for both populations: out of the 200 clusters, there are 68 and 57 of them with less than five members, respectively, and in both campuses about half of the groups have less than 10 members. More interestingly, we observe that besides these small clusters, the distribution of the cluster size seems to follow a power-law distribution. In Fig. 5-9, we plot the straight lines that illustrate the best power-law fits. The slopes for these lines are -0.67 for Dartmouth and -0.75 for USC, respectively. The power law distribution of group sizes may be related to the skewed popularity of locations on campuses - it has been shown that the number of patrons to various locations differ significantly[13]. However, the link between the distributions of number of patrons and the distribution of group sizes is not direct. While the most-visited locations on both campuses easily attract thousands of patrons, these people are broken into different behavioral groups depending on their association preferences.

We now study the detailed behaviors of each cluster by using the eigen-behavior vectors and their relative weights to understand the detailed preferences of the groups. We discover for most of the groups, their top eigen-behavior vectors dominate, i.e., the contribution of the second-most important location is almost invisible in the first eigen-behavior vector. Similar relationship holds between the second-most important location and the third-most important one, and so on. Hence the association behavior of

the group can be described by a sequence of locations of decreasing importance with a clear ordering. This observation matches with the current status of WLAN usage: people tend to access WLAN at only a limited number of locations, and the preference of visiting locations is heavily skewed (c.f. chapter 4). For such users, its most visited locations might be sufficient to classify them.

Most large user clusters belong to the fore-mentioned case. The largest clusters on both campuses include the library visitors, as expected, since libraries are still the most visited area on university campuses. For the USC campus, the largest user cluster visits the library (the first eigen-behavior vector has a single high-value entry corresponding to the library, and this eigen-behavior vector captures 83% of the power in the joint association matrix for the group), followed by a couple locations around the Law school (4.45%) and the school of Communication (4.5%), both are popular locations on campus. For the Dartmouth campus, the largest user cluster visits LibBldg2 (72.85%), followed by LibBldg1 (5.13%), SocBldg1 (3.56%), and LibBldg3 (1.93%). It seems this group consists of library patrons who mainly move about the public area on the campus and access the WLAN from these locations.

While libraries are popular WLAN hot spots, we also discover many user clusters that rarely visit these locations. The second largest cluster for USC consists of users visiting mostly the Law school (89.73% of power), school of accounting (6.37%), and a couple of locations close to the Law school (0.59%). For Dartmouth, the second largest cluster visits AcadBldg18 (56.38%), AcadBldg6 (13.4%), ResBldg83 (10.15%), AcadBldg31 (3.5%), AcadBldg7 (3.12%), which seems to be a group of students going to classes at multiple academic buildings. We have also observed various clusters featured different dorms and classrooms as their most visited location from both campuses.

On the other hand, we have also discovered groups with multiple high-value entries in its top eigen-behavior vectors from both campuses. One prominent example from the USC trace consists of 32 users, who visit buildings VKC and THH, two major classrooms

on the USC campus. The top two eigen-behavior vectors of the cluster both consist of two high-value entries corresponding to these two buildings[9] , and they combined capture 63.14% of power in the joint association matrix. This cluster consists of users who visit these two locations with similar tendency, according to the eigen-behavior vectors, and such a distinct behavioral trend exists for 32 users in the population. This cluster is a good example to show why it is not sufficient to merely use the most dominant behavioral mode (or the most-visited location) of a user to classify it. If the centroid of the dominant behavioral mode (i.e., Eq (5–4)) is used to classify users, the behavioral trend of visiting multiple locations with similar tendency will not be revealed. Instead, among the 32 users, 13 are classified with others who visit VKC frequently, 10 are classified with those who visit THH frequently, and the rest are put into various groups. As portable wireless devices gain popularity, we expect to see more users displaying diverse behavioral trends in terms of network usage. To fully capture such behavior, average-based summary is not sufficient, and this is where SVD shows its strength the most.

Interestingly, we also discover many small clusters with unique behavioral patterns that deviate from the "main stream" users in both traces. For example, in the USC trace, there is a small cluster of eight users who visit exclusively a fraternity house. Probably these are the people who live there. In the Dartmouth trace, there is a cluster of eight users who visit mostly athletic buildings (AthBldg5 (90.9%), AthBldg10 (4.62%), AthBldg2 (3.14%), AthBldg3 (0.8%), and ResBldg26 (0.54%)). These are probably either athletes or management staffs of the athletic facility. Such findings substantiate our motivation of the study: as the wireless technology prevails, we can expect users to display diverse behavioral patterns that reflect to their personal preferences, and it is important to capture such behavioral trend and quantify its significance.

[9] One of the eigen-behavior vectors has positive values for both entries, and the other has one positive and one negative value, in order to adjust the ratio between these two locations in the association vectors.

To sum up, we have demonstrated a systematic way to identify distinct behavioral groups within on-campus populations, by using clustering techniques based on association features obtained from large-scale wireless network traces. The method and findings are useful for various applications, as we discuss the next.

5.7 Potential Applications

The insights of user grouping obtained from our analysis can be applied in many different ways. We discuss some of these applications briefly in this section, including (1) behavior-aware services and group-casting, (2) user modeling, and (3) network management. We will further work on the details of designing a protocol for behavior-aware group-casting in section 5.8.

Behavior-aware services: In the future, we expect the wireless devices to be very portable and personalized. Hence, the services provided could be highly personalized, or at least customized based on the *interest groups*. Our method would facilitate to identify the dominant groups. Certainly, different *representations* of users (e.g., hobbies, interests) that fit into the context might also be utilized rather, but our method would still be applicable. Based on the targeted group of a given service, the service providers could assign a *target behavioral vector* to describe the property of target users, and the user devices could easily determine potential customers using a *significance score* (i.e., Eq. (5–5)) to compare its eigen-behavior vector to the target behavioral vector. We refer to this scenario as interest-based grouping and profile-casting. We will design a protocol for this service in section 5.8.

In addition to facilitating clustering of the users, the eigen-behavior vectors could also provide an efficient mechanism for users to exchange their behavioral features in order to *make new friends*. Such social profiles could be applied in applications in social networking, such as behavior pattern oriented matching.

User modeling: Results from the clusters of users could help us to propose more realistic models for WLAN users, which is a challenge and a necessity for evaluating

network protocols. Although mobility models with groups of user is not a new idea[58], there has been little work in realistic models based on groups. Our decomposition approach provides two pieces of important information: (1) the distribution of group sizes follows a power-law distribution and (2) the detailed eigen-behavior vectors of the groups. With such information, one can set up a generative model with the proper group sizes and the weights for frequently visited locations (e.g., its *communities* in the TVC model presented in chapter 4) to evaluate their impact on the network.

Network management: Our analysis provides a different view of network management. WLAN management and planning could be done by monitoring the activities of individual APs in order to identify the busy ones. From our clustering technique, the manager can identify user groups and the relative importance of locations to each group. Such information can be helpful in terms of load prediction and planning. For example, if the business school is going to expand, by checking the behavioral groups of business school students, it maybe possible to predict its impact on the load of different parts of the WLAN. For better understanding, one may also observe the change in the group structure with time and across semesters.

As large-scale city-wide WLAN deployments become commonplace, solutions to issues in management, service design, and protocol validation could immensely benefit from insight into the behavioral patterns of the users or the *society*. We believe that our framework will be able to provide the behavioral patterns and help find solutions to several problems ranging from wireless network management to understanding basic social behavior of users.

5.8 Profile-Cast: Behavior-Aware Mobile Networking

In this section we focus on a new class of service named *profile-cast*. In this service, instead of targeting a particular end-point or host, the message is to be delivered to *all* hosts with a certain property (i.e., those who match with the specified *profile* are the intended receivers). There exist a wide variety of ways by which a *profile* can be defined.

The *profile* can be based on the user's interest (e.g., movie-goers or baseball lovers), social affiliation (e.g., graduate students in the computer science department, students from a particular foreign country), or other behavioral patterns. Potential applications of such a service include notification or advertisement for a scoped group within the general population, or a matching service trying to find people with certain characteristics or interests. Note that the notion of *profiles* refers to the implicit, intrinsic properties one discovers from the behavioral patterns of users. This distinguishes the *profile-cast* from the traditional multi-cast where users join multicast groups explicitly: In the *profile-cast*, a user does not join particular groups to receive messages. Instead, it is a new service paradigm in which the groups are implicitly defined by the intrinsic properties of the users, and revealed by the way the users utilize the network.

In this section, we focus on providing the *service* of *profile-cast* within the *communication framework* of *DTN*. We believe this is a promising new direction – as the small hand-held devices (e.g., smart-phones, PDAs) equipped with short-range radio (e.g., bluetooth) gain popularity, they provide a channel for information to propagate within the *mobile society* independent of the existing infrastructure. The *Profile-cast* services provide a new paradigm to navigate the messages through the mobile network, reaching the targeted groups defined by their underlying properties or behavioral patterns (i.e., the chosen *profiles*).

We consider user *mobility profile* as a case study to demonstrate the efficacy of the *profile-cast* paradigm. We borrow from the user clustering results in the previous sections, and target the message propagation to the identified groups. **We analyze only a** *special case* **in the generic framework of** *profile-cast* **here, focusing on delivering messages to the same group in which the sender resides. A more generic form of** *profile-cast* **service will be introduced later, in chapter 7, after we understand the global structure of the network in chapter 6.** In this case study, we propose a *similarity-based* profile-cast protocol that makes the message forwarding decision based

on the distance between users in the multi-dimensional *profile space*. We show that by incorporating user mobility profiles, we can limit the scope of message delivery in DTNs to a specific behavioral group. Thus we avoid the high overhead of the epidemic routing [71] (i.e., we can eliminate more than half of the transmissions with a little reduction in the delivery success rate) and out-perform random-walk based protocols in terms of the delivery delay (for at least 30%).

5.8.1 Profile-Casting in Delay Tolerant Networks

One particular important decision to make for nodes in a DTN is whether to forward a packet to other nodes they encounter with, as illustrated in Fig. 5-10(a). Such decisions have implications on many aspects of how efficiently the routing strategies work, such as delay, overhead, and message delivery success rate. There exists a tradeoff between these performance metrics, and a well-designed protocol should provide a mechanism for its users to strike a right balance for the given environment. The key research challenge in designing the routing protocols is to make an intelligent decision with the *local* information available to the two encountering nodes, assuming no knowledge about the *global network properties*, which is usually unavailable in decentralized networks such as DTNs.

For our *profile-cast* service, the goal is to reach a set of nodes with a certain similar property. The conceptual view of the problem is illustrated in Fig. 5-10(b). We consider a virtual, high-dimensional *profile space* where each node is represented by a point in the space. The nodes that are similar with respect to the property we use to construct the profile space should be close to each other in this space, and dissimilar nodes should sit far apart. Our specific application we consider here corresponds to a scoped-flooding in the *profile space*: The goal is to reach all similar nodes (with respect to the profile we choose to construct the *profile space*) to the sender. The nodes should keep forwarding the message to those who are similar to them under the considered profile, but ignore those who are dissimilar. Linking the figures in Fig. 5-10, they point out a need for nodes to evaluate their mutual similarity in the considered profile space when they encounter, and

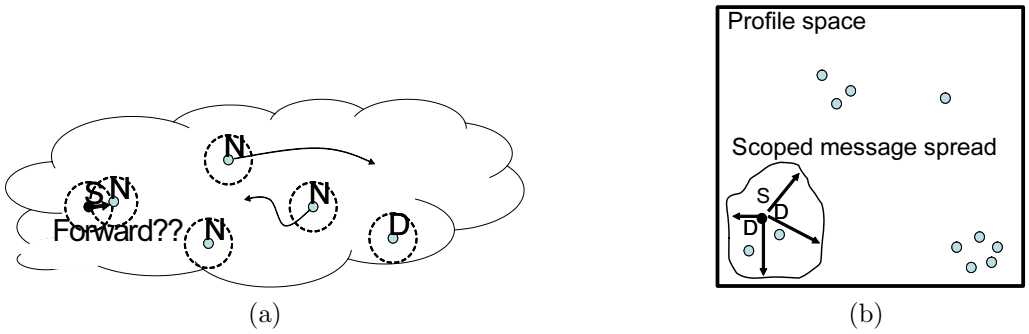


Figure 5-10. Two different views of the profile-cast service in the DTN. (a) Physical view: Forwarding decisions in the DTN. (b) Conceptual view: Scoped-flooding in the profile space. The conceptual view of scoped flooding in the profile space has to be implemented through message forwarding decisions at nodal encounter events.

use this piece of information to guide the routing decisions in the *DTN*. We propose a *similarity-based protocol* for this purpose in sub-section 5.8.2.

We use *mobility profile* as an example to illustrate the usefulness of the *profile-cast service paradigm*. We choose the mobility profile for the study for the following reasons. First, it has been shown in the previous sections that mobility is one of the distinguishing feature to differentiate users from a large population. Groups with distinct behavioral patterns can be identified with respect to the long-run mobility patterns, and we use these groups as our targets in the *profile-cast* protocol. Second, *mobility-profile-cast* ties with some new services in the ad hoc network. For example, a student loses a wallet and wishes to send a message to other fellow students who visit similar places often as he does to look for it. Or, the manager of the library may want to send an announcement about power shutdown only to its frequent patrons. These services are mobility pattern specific, and none of the existing service paradigms serves the need of identifying the intended message recipients from a diverse population well. Third, to evaluate the effectiveness of our proposed protocol realistically, we need detailed traces of user behavior with respect to

the profile we choose. User mobility data is more available than other network traces (e.g., user interest or social affiliation), hence we choose to leverage these data sets first[10] .

5.8.2 A Similarity-Based Profile-Cast Protocol

In this section we explain the details of the similarity-based profile-cast protocol, using *mobility profile* as the example. The goal here is to reach other nodes with similar mobility preferences to the sender itself. The protocol contains two phases. (1) *Profiling*: Each mobile node keeps track of its own *mobility profile* as it moves around the given environment. This is an individual effort made by each node independently – every node is responsible only for keeping its own *mobility profile*. (2) *Forwarding decision*: When nodes encounter with each other, they exchange the *mobility profiles* to determine whether a message forwarding should take place.

Profiling user mobility: To enable *mobility-profile-cast* services, it is important to first have a descriptive representation for user mobility profiles. We choose to construct the *association matrix*, as illustrated in Fig. 5-1, to describe the long-run mobility trend of a mobile user. For each time slot, each node generates an *association vector* that summarizes its association with visited locations during this time slot, as described in section 5.2.1. The *association matrix* representation captures the relative importance of locations on the campus to each user (i.e., the *preference* in the user mobility process). Based on this representation, we classify the whole user population into distinct behavioral groups with clustering methods detailed in the previous sections. These groups correspond to users with unique *mobility profiles*. In the protocol evaluation presented later, we take these behavioral groups as the targets for *mobility profile-cast*.

Evaluation of user similarity based on the mobility profiles: When nodes encounter with each other, they need to exchange the *mobility profile* for the evaluation of

[10] Note, however, if other data sets were available, similar protocols as the one proposed in section 5.8.2 could be used for other types of user *profile* as well.

their similarity. However, the raw *association matrix* is too large in size to be exchanged efficiently. Hence we need a good method for summarizing the association matrix. We have established that singular value decomposition (SVD) provides an efficient way for this purpose in section 5.4.2. The *eigen-behavior* vectors (defined in Eq. (5–10)) and its corresponding weights provide a concise yet highly accurate representation of user *mobility profile* for exchange when the users encounter with each other.

When two users meet with each other, they exchange the summarized *mobility profiles* (i.e., *eigen-behavior vectors* with their weights) of their previously collected mobility pattern and decide whether they are similar at the spot. The similarity index is calculated as the weighted sum of inner products of the *eigen-behavior* vectors, as defined in Eq. (5–11). If the similarity index is larger than a threshold, they exchange the message. Note this decision is solely local, involving only the two encountered nodes. The philosophy behind the protocol is, if each node delivers the message only to others with high similarity in mobility profile, the propagation of the message copies will be scoped within a group of similar users. The threshold that triggers the message transmission provides a control for the protocol user to adjust the tradeoff between the performance metrics. A high-valued threshold favors low transmission overhead, while a low-valued threshold leads to short delivery delay and high delivery success rate.

5.8.3 Evaluation and Comparison

5.8.3.1 Evaluation setup

In this section we describe the experiment setup to evaluate our similarity-based profile-cast protocol presented in the previous section. We utilize the USC trace (i.e., *USC-06spring*) to study the message transmission schemes *empirically*. Some logistic details of the data set can be found in Table 3-1. We assume that two nodes are able to communicate (i.e., encounter with each other) when they are associated to the same location in the WLAN. Note that the WLAN infrastructure is merely used to collect user

location information, and the messages can be transferred only between the users without using the infrastructure.

We compare the performance of our *similarity-based* protocol with several alternative protocols described below based on the following metrics: (1) *Delivery ratio*: The number of nodes receiving the message over the number of intended receivers. (2) *Delay*: The average time taken to deliver the messages to the recipient nodes. (3) *Overhead*: The total number of transmissions involved in the process of message delivery.

Flooding (epidemic routing): This is a simple decision rule for message forwarding in DTNs. All nodes in the network are oblivious to the mobility profiles and blindly send out copies of the message to nodes who have not received it yet. This scheme is also known as the *epidemic routing*[71] in DTN, using the analogy that the message propagates in the network like an epidemic. This is also the most aggressive forwarding strategy in DTN. Under an idealistic environment (i.e., no packet drop due to wireless contention or insufficient buffer size), this is also the strategy that achieves the shortest possible delay and best delivery success rate.

Centralized: In this ideal scenario, we assume that all nodes acquire perfect knowledge of the group membership through an oracle with no additional cost. In order to reduce the transmission overhead, nodes only propagate the message to others if they are in the same group. This ensures the message will never propagate to an unintended receiver, and only members of each group participate in message dissemination for their own group.

Random-transmission (RTx): In the random transmission protocol, the current message holder sends the message to another node randomly with probability p when they encounter[11] , and never transmits again (i.e., only the node who last received the message will transmit in the future). The message propagates across the network as a random

[11] In the dissertation we only show the results of $p = 1.0$ (always transmit on encounter). We have experimented with other values and discovered that they result in inferior performance.

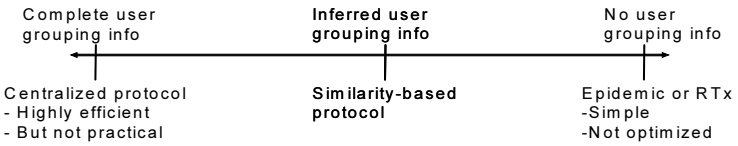


Figure 5-11. The chosen protocols for evaluation span the spectrum of user grouping knowledge used in the forwarding decision process.

walk among the nodes. Loops are avoided by not sending to the nodes who have seen the same message before. This process continues until a pre-set hop limit (i.e., TTL limit) is reached. We also vary the number of copies of active message (i.e., number of threads in the random walk) in the network. When m random walk threads are issued, the message originator is responsible for spreading the copies to m different nodes it encounters with, and each thread carries on independently as described above.

We have chosen the above protocols to span the spectrum of the degree of knowledge about the user grouping in the evaluation, as illustrated in Fig. 5-11. On one extreme of the spectrum we have the *centralized* protocol which has perfect knowledge about user grouping. This information provides an opportunity of highly efficient operation, but it is not realistic to assume its availability, hence the *centralized* protocol serves only as the *theoretical upper bound* of the performance. On the other extreme are the *flooding* and *RTx* protocols, both assuming no knowledge about user grouping at all. They are extremely simple but not optimized for the specific task of *profile-cast*. Our *similarity-based* protocol uses the similarity index defined in Eq. (5–11) to estimate the boundary where the scoped flooding should be stopped. It operates in the middle of the spectrum with *inferred* grouping information.

5.8.3.2 Evaluation results

For the evaluation, we split the WLAN trace into two halves. The first half of the trace is used to determine the grouping of users based on their *mobility profile* and we identify 200 groups with distinct behavioral pattern in terms of *mobility* using the methods detailed in the previous sections. Then we evaluate the group-cast protocol performances

using the second half of the same trace. For each group with more than 5 members, we randomly pick 20% of the members as the source nodes sending out a *one-shot message* to *all other members in the same group* at the beginning of the second half of the trace. We use the same set of senders for all evaluated protocols to ensure a fair comparison. We simulate the protocols for *mobility-profile-cast* and show the results in Fig. 5-12. For all the performance metrics, we choose the results for *flooding (i.e., epidemic routing)* as the baseline and show the normalized performance metrics of the other protocols relative to that of the epidemic routing in the figures.

In the figures we see that *flooding* has the lowest delay and the highest delivery ratio as it utilizes all the available encounters to propagate the message. However, it also incurs significant overhead. The average delay, which is the lowest possible under the given encounter patterns, is in the order of days (3.56 days in this particular case). *Profile-cast* based on *centralized* group membership information, the ideal scenario, shows a great promise for the behavior-aware protocols, as it significantly reduces the overhead (to only 3% of the *flooding*) while maintains almost perfect delivery ratio, with a little extra delay. There is such extra delay in the *centralized* protocol because the messages are carried by nodes in the targeted group only. It is possible to even reduce this delay by obtaining predictions of future encounter events through an oracle, as in [54]. We choose not to address this issue and instead show what can be achieved based on the perfect knowledge of user grouping alone, focusing our analysis on the spectrum of grouping information availability. The *centralized* protocol displays the ceiling performance one can achieve in terms of overhead reduction by incorporating knowledge of user grouping in the *profile-cast* service. However, note that it is not realistic to assume such centralized knowledge.

For our *similarity-based* protocol, its aggressiveness can be tuned with the forwarding threshold of the similarity index. We show the simulation results with various similarity thresholds in the figures. Label *Similarity* x indicates we use x as the threshold for

message forwarding[12] . Experiment results show a significant reduction of the overhead (only 2.5% of the *flooding*) at the cost of the delivery ratio (61% of the *flooding*) if we set a high threshold such as 0.7 (i.e., sending almost exclusively within the same group). Note that the overhead is even less than that of the *centralized* protocol. This setting is perhaps more suitable for applications that one would want to operate with low overhead, and it is sufficient to reach a good part of the group but not essential to reach everyone. The lost-and-found service may fall in this category. On the other hand, setting a low threshold (e.g., 0.5) leads to a better delivery ratio (92% of the *flooding*) but still cuts the overhead to 45% of the *flooding*. This is suitable for messages that are intended to be received by most of the group, but one would not mind some misses in order to cut down unnecessary transmissions to irrelevant users. Tuning the transmission threshold provides a natural mechanism to strike a desired balance between overhead and delivery ratio. The delay incurred in *similarity-based* protocol is also not much different form the optimal case, the *flooding* (up to 14% more, for the case of similarity threshold 0.5).

For the *random transmission* protocol, its aggressiveness is tuned through the setting of number of active copies of the message (m) in the network and the TTL value for each thread. We use different variants of settings and show the results in the figure with labels *RTx*. We first show that *RTx* with infinite TTL does not perform well. Even if there is only one active copy (i.e., $m = 1$) in the network, the overhead is not low (0.367 of the *flooding* protocol). Comparing with the *similarity-based* protocol, when the delivery ratio is similar, the *RTx* protocol incurs much larger overhead (e.g., comparing *similarity* 0.7 with *RTx* $m = 1$ $TTL = inf.$, or *similarity* 0.5 with *RTx* $m = 6$ $TTL = inf.$. In both cases the *RTx* has 30% more overhead than the *similarity-based* protocol.). This is due to the group-membership oblivious behavior of the *RTx* protocol – in many cases the message

[12] x can be in the range of $[0, 1]$. Setting the threshold to 1 would eliminate all transmissions, while setting it to 0 would degenerate the *similarity-based* protocol to *flooding*

is transmitted to some node out of the desired group, as the membership information is not included to guide the forwarding decisions. Hence the *RTx* protocol, without a proper TTL control, makes a lot of unnecessary transmissions and results in high overhead. Using multiple threads with long TTL essentially degenerates the protocol to *flooding*.

On a different note, we try to exercise better control of the *RTx* protocol by using infinite number of threads with small TTL. The extreme example is to use $m = inf.$ $TTL = 1$. This degenerates the protocol to the scenario where the message sender sends directly to all the nodes it encounters with. We observe that the delivery ratio is quite high with this setup. This is mainly due to the choice of our application – when the goal is to send to a group with similar mobility patterns as the sender, intuitively the intended receivers will eventually meet with the sender directly. However, notice that the delay is still much higher than the *centralized* or *similarity-based* protocols, as in this case *RTx* protocol does not take advantage of the intermediate nodes in the network. We further experiment with *RTx* $m = inf.$ $TTL = 5$, and discover it achieves good delivery ratio under moderate overhead, with improved delay. However, picking a suitable TTL is context-dependent, and it is only effective if the goal is to send messages to the nodes that are similar to the sender itself (i.e., close to the sender in the *profile space*).

We further illustrate the tradeoff between delivery ratio and overhead in Fig. 5-13, and mark the "operational region" of the compared protocols. Ideally, one would want the protocol to work at the bottom-right corner, with high delivery rate and low overhead, as close to the *centralized* protocol as possible. The *flooding* protocol also achieves good delivery rate, but the overhead is too much. Our *similarity-based* protocol is shown by the white ellipse. Its operational region stretches from moderate delivery ratio with low overhead to high delivery ratio with moderate overhead. The *RTx* protocol with infinite TTL is represented by the dark grey ellipse, taking the space of moderate delivery ratio with moderate overhead to high delivery ratio with high overhead. As m increases, it degenerates to the *flooding*. However, with a properly chosen stopping threshold, the *RTx*

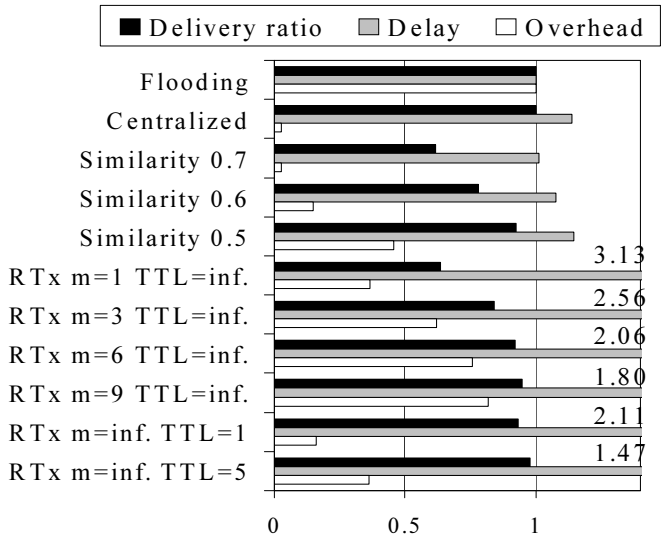


Figure 5-12. Relative performance metrics of the group-cast schemes normalized to the performance of *flooding*.

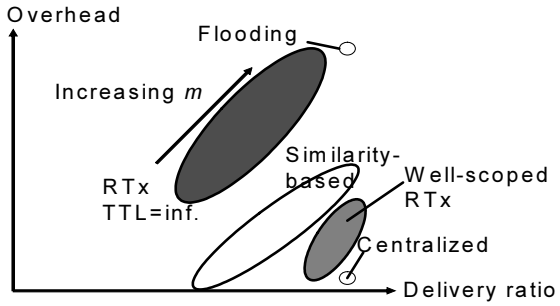


Figure 5-13. The operation regions of the compared protocols in the delivery rate-overhead space.

protocol has the potential to operate in the high delivery ratio, low overhead area, as indicated by the light grey ellipse. However, its average delivery delay is still much higher than that of the *flooding* or *similarity-based* protocols (in the best case, at least 30% more than the similarity-based protocol), as *RTx* does not take full advantage of the available intermediate nodes in the DTN framework.

5.8.4 Extensions of the Profile-Cast Service

In this section we focus on designing the service of *mobility-profile-cast* with target nodes being the ones in the *same* behavioral group as the sender. There are various ways

in which one would like to extend the capability of such a service, in particular, (1) How could a message be delivered to a group with a *specific mobility profile* given by the sender (instead of targeting at similar nodes to the sender)? (2) How could we use different type of *profiles* to describe the target group (instead of the *mobility profile*) in the proposed *profile-cast* service paradigm? In order to do these, we have to first understand a hidden structure of nodal encounters in wireless mobile networks, which is the topic in the next chapter. We will show one promising finding pointing out that the encounter patterns of nodes in realistic mobility traces form SmallWorlds[8], and eventually leverage this finding to design efficient message delivery protocols for more generic cases in chapter 7.

5.9 Conclusion

Our contributions: In this chapter, we classify groups of WLAN users based on the trends in their association patterns in two major university campuses by leveraging clustering techniques and our systematic *TRACE* approach. We design a novel distance metric between users based on the similarity of their *eigen-behavior* vectors, obtained through singular value decomposition (SVD) of the association matrices. SVD is the optimal way to capture underlying trends in the data set, and it also leads to space and time efficient computations. The eigen-behavior distance leads to a meaningful partition of users. We establish that WLAN users on university campuses form a diverse community, which includes hundreds of distinct behavioral groups in terms of association patterns. We further propose *profile-cast* as a new service paradigm, and demonstrate that *mobility-based profile-cast* can be utilized for scoped message dissemination in DTNs and show improved performance over other candidates (i.e., the epidemic routing or random transmission). The proposed *similarity-based* protocol shows significant overhead reduction (less than 45% of overhead compared to the *flooding* with high delivery rate, or as low as 3% of the overhead with a moderate 61% delivery rate). It is also better than the *random-transmission* protocol in terms of the average delay (at least 30% improvement over random-walk protocols). We display that the insight from a detailed study of user

behavior might provide new directions to improve services and protocols, especially as services become highly personalized. In the next two chapters, we will further pursue this line of study, delve into the patterns of user encounter events using a graph analysis approach, and use it to enrich the capability of the *profile-cast* service.

It is surprising to find qualitative commonalities in the user behavior almost across the board considering the differences (e.g., Geographical locations, sizes and structures of the campuses, different student bodies, etc.) among the campuses: (1) More than 60% of the WLAN users display multi-modal behavior (their behavior can be decomposed into multiple modes or types) in the long run. However, for many users the most dominant behavioral mode is much stronger than the rest. This leads to efficient summaries of their behavioral patterns. With SVD, we can capture more than 90% of the power in the association patterns with just five components. (2) Current university WLANs consist of a large number of user groups with distinct association patterns, in the order of hundreds. We find that the distributions of sizes of the major groups, however, are highly skewed and follow a power-law distribution. The top-10 groups contain at least 33% of the users while about a half of the identified groups have less than 10 members.

5.10 Alternative Methods

Besides the *normalized association vectors* and *eigen-behavior distance* obtained through SVD, there are many other potential representations of user association behavior and distance metrics. In this section we briefly discuss these alternatives and some results we obtain with those.

5.10.1 Various Distance Metrics

We establish a meaningful partition of both user populations in Fig. 5-7 with the eigen-behavior distance. However, we have to note that the other summaries presented in section 5.4.2 could also be used to obtain distance metrics. For these single-vector summaries, such as the average of association vectors (X_{onavg}, Eq. (5–3)) or centroid of the first cluster ($X_{centroid1}$, Eq. (5–4)), we define distance metrics between users by simply

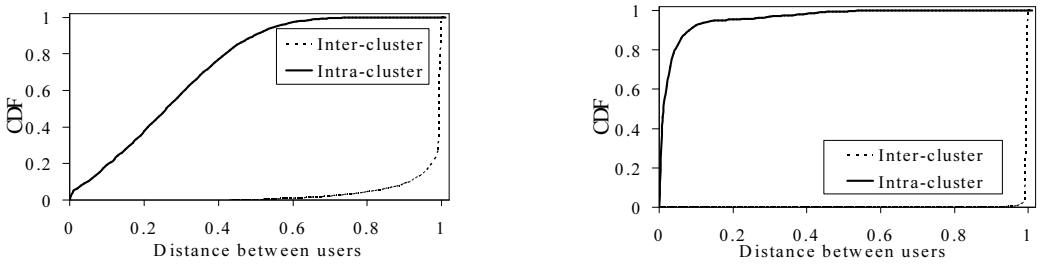


Figure 5-14. Cumulative distribution function of distances for inter-cluster and intra-cluster user pairs (other distance metrics). (a) USC, distance between average of association vectors. (b) Dartmouth, distance between centroid of the first behavioral mode.

calculating the Manhattan distance (Eq. (5–2)) between the corresponding summary vectors. With these distance metrics, we could also arrive at meaningful partitions of user populations and hence those are valid metrics, too. We show two such examples in Fig. 5-14 - the general observation is that while X_{onavg} leads to less well-separated clusters than the eigen-behavior distance, $X_{centroid1}$ leads to even better results.

We need to further compare these different partitions of the user population to understand their properties. We choose to use the Jaccard index [96] to compare the similarity between different partitions of the same population. The Jaccard index between two partitions on the same population is defined as

$$J(P, Q) = r/(r + u + v), \tag{5–12}$$

where r is the number of user pairs who are partitioned in the same cluster in both partition P and Q (i.e., where the two partitions agree on the classification). u (or v) is the number of pairs who are in the same cluster in P (or Q) but in different clusters in Q (or P) (i.e., where the two partitions disagree). We choose the Jaccard index among many other indices for partition similarity due to its low variance on partitions with the same transfer distance [96]. For both traces, we list the Jaccard indices between user partitions from various distance metrics (Average and the Centroid of first behavioral mode) and the

Table 5-2. Jaccard indices between user partitions based on the eigen-behavior distances and various distance metrics.

Distance metric	Average	Centroid w/ threshold = 0.5	Centroid w/ threshold = 0.9
USC	0.757	0.741	0.696
Dartmouth	0.801	0.706	0.710

partition from the eigen-behavior distance in Table 5-2. The Jaccard indices are mostly in the range of 0.7 to 0.8, indicating the partitions are in fact similar. The better separation between intra and inter cluster distance distributions with the $X_{centroid1}$ metric is partly because the distances are calculated based on a subset of association vectors with a coherent trend, discarding other vectors. Nonetheless, different distance metric has its own emphasis. While we argue in section 5.6 with an example that the eigen-behavior distance is useful for classifying users with multiple frequently visited locations with similar preferences, this is not always the only goal. Depending on the application, sometimes one may want to consider only the first behavioral mode and ignore the others.

Instead of applying SVD, one may propose to use the centroids for *all* behavioral modes (i.e., all the clusters formed by the user's association vectors) of a user as a summary. However, the behavioral mode for each user is dependent on the clustering threshold, and it is not simple to choose one that works well for many users, considering the diversity. On the other hand, SVD does not require parameter tuning, and is optimal in the sense of capturing remaining power in the association matrix, so we choose it over the multiple centroids method, if all behavioral modes of a user should be considered.

5.10.2 Various Data Representations

In this section we discuss about alternative representations of user behavior and compare the findings with the *normalized association vector* we choose in the chapter.

In section 5.2.1 we propose to use the normalized association vector in order to mitigate the differences of user activeness across users and across time slots for a given user. This is effective if the *preference* of the user for each time slot is the focus of study.

For example, if a user visits exclusively location A or B on different days, for similar number of days, one way to understand the user behavior is that locations A and B are of the same importance for the given user, since he visits these two places exclusively with similar frequency. However, if the user stays at location A whenever he visits for much longer duration than B, the normalized vector would not reveal such information. On the other hand, if absolute association time is used in the vectors, the large time spent at location A will hide his visits to B when SVD is applied to extract the eigen-behaviors from the user (i.e., vectors with large association time to A dominate the power of the matrix), albeit the user pays a lot of visits to B.

Both representations may be of interest to some applications. So instead of arguing the importance of one over the other, we try to understand its impact on how the users are clustered. Using the USC trace as an example, we compare the results of user partitions using the absolute association time vectors and the normalized association vectors. We observe that the most active users (i.e., The first quarter in terms of the online time) are almost classified the same regardless which representation is used, with Jaccard index 0.9652. This is the case since the most active users are almost always on, and the representation does not make much difference. We see the Jaccard indices drop to 0.7910, 0.7090, and 0.6096 for the second, third, and fourth quarter of users in terms of activeness, respectively, a clear decreasing trend. The least active users are more sensitive to the choice of representation due to their sporadic usage of WLAN.

Time slot sizes to collect the association vectors is another dimension to experiment with. In addition to daily vectors, we consider two other schemes: (1) Generate association vectors for every three-hour time slot. We compare the partitions generated by this fine-grained representation with the partition generated by the daily representation, and get the Jaccard indices of 0.787 and 0.778 for USC and Dartmouth, respectively. This indicates that a finer time scale does not change the user classification much, and our original one-day interval would be sufficient to capture important trends in user behavior.

We also try (2) generate the association vectors only during the time frame between 8AM to 4PM, the busy part of a day, and compare the subsequent partition with the partition generated by the daily representation in which the whole day is included. With this representation, the two traces give very different result – the Jaccard indices are 0.752 and 0.033, respectively. Hence it is not always sufficient to use only the behavior trends during working hours to classify users.

The choice of location granularity in the representation is also important to understand the results. We have also attempted to use access points as locations for the Dartmouth trace as the information is available. For most of the studies, the observations are similar to what we presented so far in the chapter, although one can expect to see more distinct behavior groups from the population if finer location granularity is used. However, it is not easy to interpret these groups meaningfully unless we have the information about detailed AP locations within the buildings and the significance of its covered area in social context (e.g., it does not make much sense in the social context to say a group is featured by visiting the South-East corner of an engineering building often, unless we know a faculty lounge is at that corner.). On the other hand, it is also possible that a group of buildings bear a higher-level meaning in social context (e.g., several close-by dorms form a "residential area", or close-by buildings shared by the students from the same department), and it is also related to understand user visiting preferences from a higher-level behavioral context (e.g., home, at work, at class, etc.). We leave this as future work.

On a different note, it may be possible to use other representations in different type of networks. For example, in encounter-based networks, a representation of encounter probability or duration would be appropriate. We plan to investigate this in our future work.

CHAPTER 6
CASE STUDY III: UNDERSTANDING THE GLOBAL NODAL ENCOUNTER PATTERNS

After we study the behaviors of users in the WLAN traces, as individuals and as members of groups of similar users, in chapter 4 and 5, respectively, we take an even more macroscopic view in this chapter. We consider an important event between mobile nodes in wireless networks – *encounters*. The scope of the analysis is one step wider than what we presented in the last chapter – although we consider encounter events as the enabling events of node-to-node communication in the profile-cast protocol, we utilize these events in a localized fashion. In this chapter, we seek to understand encounters in the mobile network from a different perspective – we take a holistic view on all encounter events happening between all the nodes in the network and study the global encounter patterns in the trace, by observing the encounter patterns with a graph analysis approach. Such an analysis sheds light on the feasibility of forming a infrastructure-less network capable of reaching most of the nodes through time-varying, partial connectivity to some nodes at a given time instant through encounters.

6.1 Introduction

Our work in the previous chapters provides a good understanding of WLAN users. However, most of the research work is focused on individual behavior of mobile nodes (MNs) thus far. The understanding of individual behavior is important in itself, but it does not reveal how MNs interact with one another in the real traces. In this chapter we go beyond the level of individual users, and start to look into a simple yet important interaction event among MNs: *encounters*. Encounters are important events in wireless networks as they provide chances for MNs to directly communicate, even without an infrastructure. We seek to understand encounter patterns in the mobile network from a holistic view by a graph analysis approach. Such an analysis sheds light on the diverse, non-homogenous nature of users in the given environments in terms of their encounter events with other nodes. Furthermore, we evaluate the feasibility of forming an

infrastructure-less network to reach most of the nodes utilizing time-varying inter-node connectivity through encounters, and the robustness of such an ad hoc communication network. We seek to understand encounter patterns of MNs by analyzing month-long WLAN traces from university and corporation campuses in this chapter (i.e., the *MIT-rel*, *Dart-03*, *Dart-04*, *UCSD*, *USC*, *UF* traces from Table 3-1) in addition to a real encounter trace collected at a recent INFOCOM conference[27] (i.e., the *Cambridge-INFOCOM05* trace in Table 3-1). We compare and contrast our observations for the various traces to distill and explain the commonalities and differences observed.

Specifically, we aim to quantify the distribution of encounter events a MN has, and look into the encounter patterns of all MNs to understand the relationship between MNs formed by encounters. This is a research topic that received less attention in the past, but can be useful and sometimes essential for classes of future mobile networking protocols. For example, encounter histories are used to discover routes in ad hoc network routing protocols (e.g. MAID[46], EASE[93]), and encounters are used directly in delay tolerant networks (DTNs) to propagate packets. We define an *encounter* between two users as the event of their association with the same AP for overlapped time intervals. From all the WLAN-based traces we studied, we find that the distribution of encounters is highly asymmetric, indicating a heterogeneous user population. Surprisingly, we find that a user, on average, only encounters between 0.79% and 6.7% of the network user population within a month. We also establish that the total number of encounters for each MN follows BiPareto distribution, the parameters of which are environment specific. We further utilize a graph analysis approach to understand the relationship between MNs formed by encounter events. We utilize the Small World model [8] to understand the characteristics of the *encounter-relationship graphs (ER graphs)*, in which two nodes are connected by a link if they ever encounter. We find that although direct encounters of individual nodes happen only to a small portion of node pairs among the whole population, WLAN users form connected Small World graphs via encounters, and the

metrics of the formed Small Worlds (i.e., disconnected ratio, clustering coefficient, and path length) converge quickly in about one day to its long-term steady values in most cases.

Also, from chapter 4 we know MNs in WLAN traces are in fact not uniformly distributed, and users with similar preferences show up at the same access point (AP) more frequently. We look into this issue and try to identify the closeness (i.e., *friendship*) between node pairs, and understand its influences on network connectivity if we make connections between nodes based on their friendship. Specifically, we give several intuitive definitions of friendship between MNs. These friendship indexes capture the observed closeness between the involved MNs from the trace. Although such closeness may or may not reflect friendship in a social context, it reveals the closeness between wireless devices as displayed in their association patterns. The empirical distributions of these friendship indexes mostly follow the exponential distribution, with few node pairs showing high friendship index. Furthermore, we investigate the issue of how friendship influence the characteristics of the *encounter-relationship (ER) graphs*. We find that if only nodes with high friendship indexes are used in forming the *ER graph*, the resultant graph displays higher clustering coefficient and average path length. In other words, it is more inclined toward a *regular graph*. On the other hand, if we use only nodes with low friendship index in the ER graph, it displays lower clustering coefficient and average path length. This finding points out, similar to social networks, close friends in WLANs often form cliques and random friends are keys to widely-reached connectivity in a network.

Finally, we propose information diffusion experiments to understand how information could be spread among users *without* the help of an infrastructure. We use a simple message spreading strategy to investigate whether it is possible to rely on mutual encounters to spread messages across the network. Surprisingly, given the seemingly very low ratio of the whole population a given node encounters with, the encounter events form a wide-reaching communication network, and the messages spread to most of the whole

population. We further show that even with a relatively high percentage of users being selfish (i.e., not participating in information propagation), the information still spreads and reaches most of the population, indicating the richness of the encounter patterns in current WLAN users. Also, if encounters with short time duration are not exploited, the performances of information diffusion also do not degrade significantly. In addition, the delay of message delivery also does not increase significantly with the addition of selfish users or the removal of short encounter events.

We study the encounters between MNs in section 6.2 and introduce the Small World approach to explain the encounter-relationship graph in section 6.3. We further explain the reason for the Small World to form in section 6.4. Then we discuss the findings in our effort to capture friendship between MNs in section 6.5. Finally, the information diffusion experiment is explained in section 6.6. We provide some discussions and conclude the chapter in section 6.7, together with an outline of our proposed future work on an efficient selective broadcasting protocol.

6.2 Encounters between Nodes

Nodal encounters in mobile networks are important events as they provide opportunities for involved nodes to build up some relationship or to communicate directly. In this chapter we focus on understanding the derived encounter events (refer to chapter 3 for the details) from WLAN traces. Although these derived encounter traces may be not completely accurate, we believe that the encounter events derived from WLAN traces capture a major portion of MNs within direct communication range under current usage pattern.

The distribution of these encounter events is the first step to understand the structure of inter-MN relationship in the traces. The direct questions to ask about the encounter events are: How many other MNs does a user meet? Do nodes meet with each other repeatedly or not? Fig. 6-1 shows the CCDF of fraction of other MNs a given MN has encountered through the whole trace period (i.e., one month). From the figure we

observe that all the nodes in WLAN traces encounter only at most about 50% of the user population within a month, with the UCSD trace being the only exception. This may be partly due to the fact that the 275 PDA users in the UCSD trace were all selected from the freshman class, and they tend to stay in several common dorms as stated in [11] (in other words, the MNs in this trace are selected from a *correlated* sub-group of the whole population on campus). In all other traces, **on average a MN encounters with only** 0.79% **(UF) to** 6.70% **(Dart-04) of the whole user population** within the 30-day trace period. The small average encounter ratio is a combined result of several reasons: (1) most MNs are not always on, and (2) most MNs do not visit many APs[66], hence they can only meet with those who also visit this small set of APs.

Low encounter percentage as shown in the traces is not observed in any of the simulation scenarios used for performance evaluation in the literature. For example, in Fig. 6-2, we show the CCDF of unique encounter fraction obtained from the random direction mobility model, one of the commonly used synthetic mobility model. We observe two obvious differences from the empirical traces: (1) The unique encounter fraction reaches 100% for all nodes within two days. This is because, in typical synthetic mobility scenarios, as those summarized in [49], all nodes follow the same model to make movement decision, albeit with randomness, and eventually encounter with all other nodes [46]. (2) The diversity of the unique encounter fraction, given an observation time period, is not very high (e.g., Within six hours, all nodes encounter between 52% to 75% of the population). The encounter pattern from real wireless network traces, on the other hand, reflects that university campus is a *heterogeneous* environment rather than a homogeneous one constructed by the synthetic models in which all nodes are statistically *i.i.d.*. To better understand how protocols perform in such heterogeneous environment, using homogeneous synthetic models is not sufficient. This finding adds to the motivation of using a flexible mobility model, such as the TVC model we propose in chapter 4, which

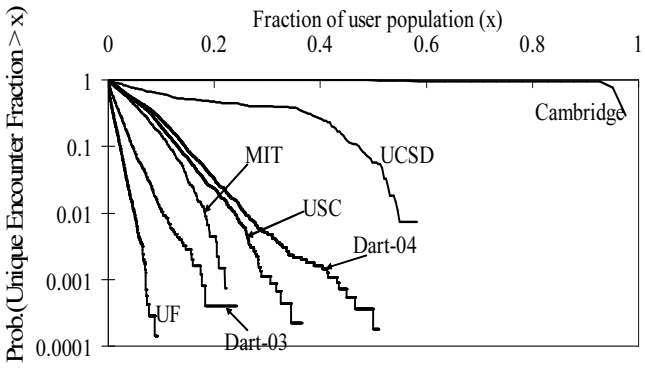


Figure 6-1. CCDF of unique encounter fraction, traces.

is capable of describing nodes with diverse, heterogeneous behavior for future protocol evaluations.

On the other hand, from the Cambridge trace, most of the 41 users meet with the majority of others during the short trace duration (4 days). Specifically, there are 12 MNs who meet with all other 40 MNs, and 39 out of 41 MNs meet at least 38 other nodes. The curve in Fig. 6-1 is mostly a horizontal line at high probability until the unique encounter fraction reaches 0.95. This high unique encounter fraction may be due to the environment setting (a conference, where the premises is considerably smaller than a university campus or corporate buildings, and people are supposed to meet with each other at a conference) or the fact that the selection of participants are related (i.e., people who are interested in the study of mobility patterns and wireless networks in general) rather than randomly picked from the conference attendees.

We also show the CCDF of the total encounter events a MN has throughout the trace period in Fig. 6-3. We observe the **total encounter counts for MNs in each trace span across several orders of magnitude**. There are both MNs with extremely few or many encounters. This is an evidence of **heterogeneous behavior** among MNs. The actual number of total encounters depends on the size of population in the traces. Large traces (i.e., the USC and Dartmouth traces) tend to have more encounters than small traces (i.e., the UCSD and Cambridge traces). However, regardless of the size of

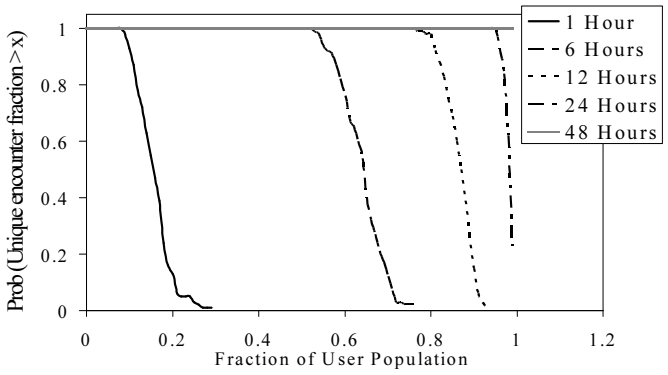


Figure 6-2. CCDF of unique encounter fraction, synthetic model (Random Direction model).

population, **the curves for the total encounter count derived from WLAN traces seem to follow the BiPareto distribution**. We fit the BiPareto distribution curves to the empirical distribution curves, and use the Kolmogorov-Smirnov test [98] to examine the quality of fit. The resulting D-statistics for all traces are between 0.068 and 0.025, which indicates we have a reasonably good fit between the BiPareto distribution curves and the empirical distribution curves. The details about the Kolmogorov-Smirnov test and the parameters of the fitted BiPareto distribution curves are listed in section 6.8. For the Cambridge trace, the total encounter counts for MNs are not as diverse as those in WLAN traces. This may be due to the fact that most nodes participate the conference actively throughout the whole trace period (4 days), but this is unlikely for the longer, one-month WLAN traces. The BiPareto distribution does not show a good fit for the Cambridge trace, as its total encounter distribution drops sharply at a "knee" around 250.

A closer investigation of the relationship between the unique encounter count and the total encounter count of the same MN reveals that **high unique encounter count does not always imply high total encounter count**. The correlation coefficients between the unique encounter count and the total encounter count for various traces range from 0.732 to 0.195. Except for the UCSD trace, all other traces have correlation coefficients below 0.6. As an illustration, we show the scatter plot of the unique encounter count versus the total encounter count for the USC trace in Fig. 6-4. We observe that some

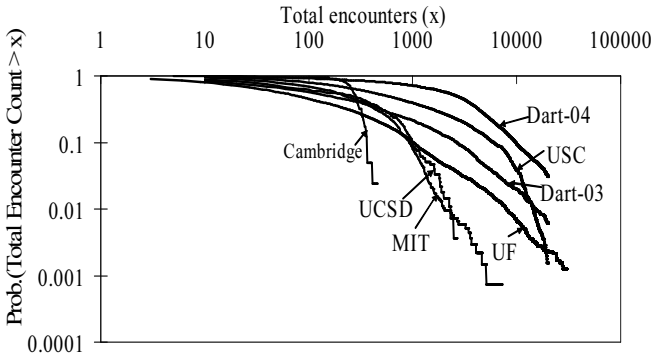


Figure 6-3. CCDF of total encounter count.

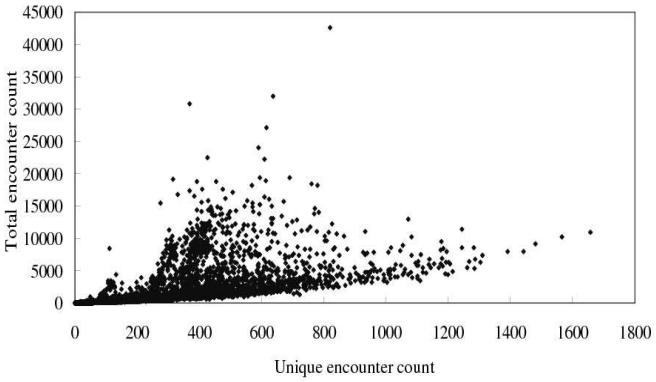


Figure 6-4. Unique encounter count versus total encounter count, USC.

nodes have not many unique encounter counts, but high total encounter counts. This indicates that some node pairs may have a lot of repetitive encounters, suggesting these node pairs have closer relationship than other pairs. This point warrants further study, and we will show some initial attempts on quantifying the *friendship* between MNs in section 6.5.

6.3 Encounter-Relationship Graph

In section 6.2, we see that **MNs have low percentage of unique encounters among the whole population**. Given this fact, We raise a question regarding the possibility of establishing campus-wide relationships among the majority of MNs via encounters alone. That is, do encounters link MNs on the campus into one single community, or just many small cliques?

To investigate this question, we define a *static encounter-relationship graph (ER graph)* as follows: Each MN is represented by a node in the *ER graph*, and an edge is added between two nodes if the two corresponding MNs have encountered at least once during the studied trace period. By the construction of the *ER graph*, we collect all encounter events between MNs within a time period and collapse them on a static graph. The exact timing of encounters are ignored, but we focus on the structure of interconnections built between nodes by available encounter events during that period of time. In other words, the concept of *ER graph* is introduced to capture the potential of establishing a connected network among MNs based on direct encounters alone, and understand the structure of such a network.

We use three important metrics to describe the characteristics of the encounter-relationship graphs, defined as follows:

- **The clustering coefficient (CC)** is used to describe the tendency of nodes to form cliques in a graph. It is formally defined as [44]:

$$CC = \frac{\sum_{n=1}^{M} CC(n)}{M}, \tag{6–1}$$

where

$$CC(n) = \frac{\sum_{a,b \in N(n)} I(a \in N(b))}{|N(n)| \cdot (|N(n)| - 1)}. \tag{6–2}$$

$N(n)$ is the set of neighbors of node n in the ER graph and $|N(n)|$ is its cardinality. $I(\cdot)$ is the indicator function. M is the total number of nodes in the graph. Intuitively, the clustering coefficient is the average ratio of neighbors of a given node that are also neighbors of one another. Higher CC indicates higher tendency that neighbors of a given node are also neighbors to each other, or heavy "cliquishness" in the relationship between MNs formed through encounters.

- **The disconnected ratio (DR)** is used to describe the connectivity of the ER graph. It is defined as:

$$DR = \frac{\sum_{a=1}^{M}(M - |C(a)|)}{M(M-1)}, \tag{6–3}$$

where $C(a)$ is the set of nodes that are in the same connected sub-graph with node a. DR indicates, on average, the percentage of unreachable node starting from a given node in the graph.

- **The average path length (PL)** is used to describe the degree of separation of nodes in the *ER graph*. It is defined as:

$$PL = (1 - DR) \cdot PL_{con} + DR \cdot PL_{disc}, \tag{6–4}$$

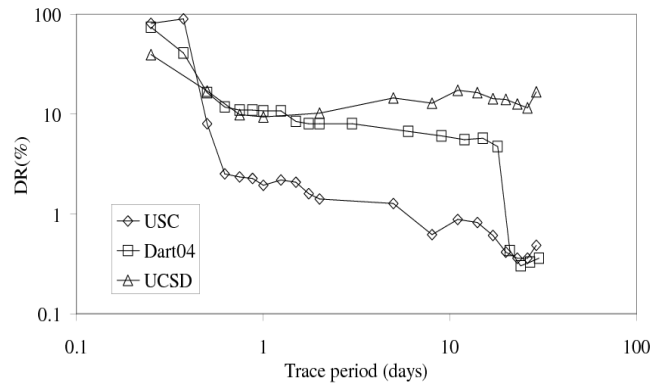
where PL_{con} is the average path length among the connected part of the *ER graph*, defined as:

$$PL_{con} = \frac{\sum_{a=1}^{M}\sum_{b \in C(a)} PL(a,b)}{\sum_{a=1}^{M}|C(a)|}. \tag{6–5}$$

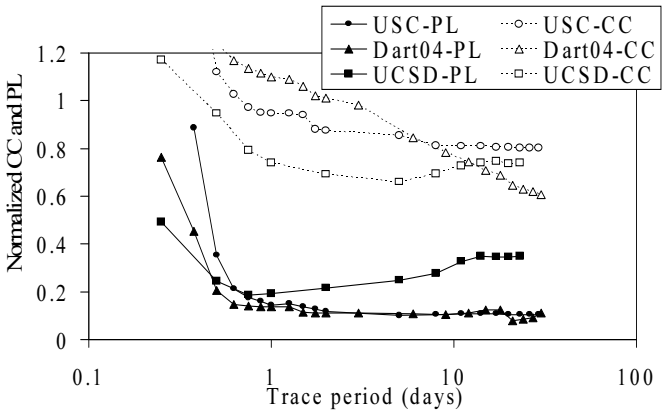
$PL(a,b)$ is the hop count of the shortest path between node pair (a,b) in the *ER graph*[1] . PL_{disc} is the penalty on the average path length for *disconnected* node pairs in the *ER graph*. In the following we use the average path length of the regular graphs (defined later) with the same node number and average node degree for PL_{disc}.

We study how the above metrics evolve for the *ER graphs* derived from various studied period of WLAN traces. Taking the USC trace, the Dartmouth trace (Dart-04), and the UCSD trace as examples, we show the evolution of the three metrics with respect to various studied trace periods in Fig. 6-5. The graphs for other traces show very similar trends, and we leave them in section 6.9 to maintain conciseness here.

[1] Note this path is not the same as the shortest spatial path between node pair (a,b), which may not even exist.



(a)



(b)

Figure 6-5. Change in the *ER graph* metrics with respect to trace period. (a)Disconnected ratio. (b)Normalized clustering coefficient and average path length. The figure is cut from above to show the details between 0 and 1 on Y-axis.

From Fig. 6-5 (a) we note that given sufficient long trace durations, the *ER graphs* **have low** *DR* **(not larger than** 10% **for traces longer than one day in most cases), which implies that nodal encounters are sufficient to provide opportunities to connect almost all nodes in a single community**, even though each node encounters only a small subset of MNs directly. This is an encouraging result that points out the feasibility of building a large, widely-reach network relying only on direct encounters. Although the *DR* starts out very high with very short trace periods (i.e., for trace durations under one day) since MNs have not moved around to create encounters yet, it decreases rather quickly as the trace period increases. Within one day, *DR*'s reduce

Table 6-1. Equations for the CC and PL for the regular and random graphs with M nodes and average node degree d [8, 44].

Graph type	Clustering coefficient	Average path length
Regular graph	$3(d-2)/4(d-1)$	$M/2d$
Random graph	d/M	$log(d)/log(M)$

to around 10%. Although the numbers of MNs in the *ER graph* keep increasing as we look at longer trace periods, in most cases the *DR* does not change significantly after one day.

Another interesting finding is revealed by the other two metrics, the clustering coefficient (CC) and the average path length (PL). To highlight a unique property of these *ER graphs*, we also calculate the CC and the PL for *regular graphs* and *random graphs* with the same corresponding total node number M and average node degree d. These quantities can be calculated according to equations in Table 6-1. In the regular graphs, nodes are first arranged on a circle and each node is connected to d closest neighbors on the circle. In the random graphs, d randomly chosen nodes are assigned as neighbors for each node. Typically, regular graphs have high CC and PL while random graphs have low CC and PL. They are the two extreme cases on the spectrum. In Fig. 6-5 (b), we show the normalized CC's and PL's of the *ER graphs* for various trace periods. These normalized metrics represent, on the scale from 0 (corresponding to the random graph) to 1 (corresponding to the regular graph), where the metrics of the *ER graphs* fall. They are defined as:

$$CC_{norm} = \frac{CC - CC_{rand}}{CC_{reg} - CC_{rand}} \tag{6–6}$$

$$PL_{norm} = \frac{PL - PL_{rand}}{PL_{reg} - PL_{rand}}, \tag{6–7}$$

where CC_{norm} and PL_{norm} represent the normalized CC and PL, respectively. The subscripts reg and $rand$ imply that the corresponding metric is obtained from the regular graph and the random graph, respectively, with the same total node number and average node degree.

We observe that *ER graphs* display **high normalized CC's** which are close to those of the corresponding regular graphs (i.e., normalized *CC*'s being close to 1, and in some cases even higher than 1), and **low normalized PL's** which are close to those of the corresponding random graphs. This highlights that a special pattern of encounters exists in all WLAN traces: Nodes visiting similar sets of APs are highly likely to encounter with all others and introduce highly connected clusters among these nodes, leading to high *CC*. This phenomenon is especially obvious for very short traces, since most MNs do not change its association to the APs to create many encounters. The *ER graphs* for short trace periods feature many small disconnected cliques, each of them being a full-mesh formed by MNs associated with the same AP for that trace period. As we look at longer traces, some of the nodes in one cluster also have random encounters with nodes in other clusters, and these links serve as the "shortcuts" in the *ER graphs* that reduce the *PL*. In previous literature, graphs with high *CC* close to the regular graphs and low *PL* close to the random graphs are referred as Small World graphs [8], [44]. By looking at various traces, we indicate that the *ER graphs* **formed by encounters among nodes using wireless network appear to be Small World graphs**. We also observe that **both** *PL* **and** *CC* **converges to its final values rather quickly in about one day for most traces**, although the size of *ER graphs* keeps increasing as more nodes appear in longer traces.

For the Cambridge trace, we look into similar metrics. We find that for even a small period of time (e.g., 1 day) the 41 MNs encounter most of the whole population. Hence, the *CC* is very high (above 0.91 even if we take only the first day into consideration), and the *PL* is low (less than 1.1). Actually, the 41 MNs presented in the trace almost form a fully-connected mesh, and the *DR* is 0. This may be partly due the nature of the conference setting from which the trace was collected. People move around to meet more often than in their regular daily life at universities or corporations, hence the encounter pattern at a conference seems to be richer than in regular environments. The

well-connected *ER graph* may also come from the fact that the conference was held in a place much smaller than a university campus or a corporate building. Conceptually, the single clique in the Cambridge-INFOCOM trace may in fact correspond to one of the cliques observed in the WLAN traces (i.e., the MNs visiting similar sets of APs). However, the above arguments need further validation, by more thorough study of encounters in different settings.

6.4 The Reasons underneath the Small World Encounter Pattern

In this section, we follow up on the intuition briefly introduced in the last section to further understand the reasons for the Small World encounter pattern to emerge. One theory we suggest in the last section is that nodes visiting similar sets of APs are highly likely to encounter others with similar mobility preference and introduce highly connected cliques among these nodes, leading to high *CC*. And then, some of the nodes in one clique also have random encounters with nodes in other cliques, and these links serve as the "shortcuts" in the *ER graphs* that reduce the *PL*. We will correlate the notion of similarity metric of nodal association pattern introduced in section 5.5.1 (see Eq. (5–11)) and the Small World graphs to validate this intuition.

We devise the following experiment to understand the effect of mutual similarities between users' association patterns on the global encounter patterns. Using USC trace as an example, we categorize all user pairs into four zones, as illustrated in Fig. 6-6. Zone A consists of user pairs who are highly similar (with the similarity metric above 0.8), and zone B, C, and D consist of user pairs with less similarity in each zone. The boundaries between the zones are so chosen that, when we consider an average user, it has roughly similar number of encountered users falling in each zone.

After designating user pairs into zones, we redraw the *ER graphs* to include only links between two nodes in the graph if the node pair belongs to a certain zone. This is an effort to evaluate how links among similar or dissimilar users play its roles in the resulting

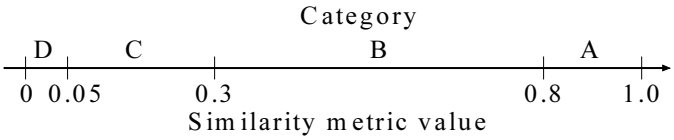


Figure 6-6. Classification of node pairs into different categories based on their similarity metric range.

Table 6-2. The graph properties of the ER graphs with selected links (only links falling into certain similarity categories (see Fig. 6-6 for the bins) are included).

Links included from zone	A	B	C	D	AB	BC	CD	ABC	BCD	ABCD (all)
Average node degree	72.48	72.16	62.27	62.73	144.62	134.43	125.00	206.89	197.16	269.62
Disconnected Ratio (%)	96.85	8.98	11.35	7.25	6.36	4.22	4.26	2.40	1.49	0.53
Clustering Coefficient	0.7814	0.4568	0.1737	0.2968	0.6973	0.4896	0.3578	0.6339	0.5003	0.6117
Average Path Length	1.537	3.102	2.638	2.563	3.092	2.397	2.359	2.385	2.236	2.200

ER graphs. For *ER graphs* including links from various zones, we again obtain the three graph properties introduced in the last section, and summarize them in Table 6-2.

We see from Table 6-2 that when the *ER graphs* include only edges from one zone, under similar average node degree in the *ER graph* (we have chosen the categorization bins carefully to ensure this), if the edges are formed between nodes with high similarity, it results in high disconnected ratio and clustering coefficient in general. This trend is especially pronounced for the *ER graph* including only edges in zone A, validating our intuition that extremely similar nodes (in terms of their *mobility preferences*) form disjoint clusters. The node pairs that are dissimilar to each other (e.g., node pairs in zone D) lead to an *ER graph* with low disconnected ratio, low clustering coefficient and low average path length[2] . Similar trend is also observed when we include edges from two or three zones – indeed taking edges from only similar nodes increase the CC and PL, and the inclusion of edges between dissimilar nodes decrease DR, CC, and PL.

The above observations reveal that the heavy cliqueness in the ER graphs stems from groups of nodes visiting similar locations. Notice although it is not guaranteed that all

[2] Notice that the average path length for the *ER graph* with only edges in zone A is the lowest. However, this is an artifact due to the extremely high disconnected ratio – the paths between nodes, if exist, are all short paths between nodes in the same clique.

of them end up encountering each other[3] , in general users do meet with other similar users with higher probability. On the other hand, we observe that as encounter events between dissimilar nodes are added into the *ER graph*, the DR, CC, and PL begin to fall, indicating the special role of "short-cuts between the cliques" played by these random links.

In the next section, we further investigate the interplay of inter-node relationship and the *ER graph* structure, from a slightly different perspective. We consider the notion of *friends* as people who I encounter repeatedly and frequently, and see how friendship changes the structure of the *ER graphs*.

6.5 Capturing User Friendship in WLAN Traces

In this section we further try to quantify the *friendship* between MNs based on information available from the traces, and its influences on the *ER graphs* when we include only friends in the graph.

In our daily lives, we are bound to meet with colleagues and friends much more often than others. In this section we try to investigate using the wireless LAN traces whether such an uneven distribution of closeness among MN pairs exists, and try to measure it using the concept of *friendship dimensions*. The likelihood or duration of encounters between two MNs captures the *friendship* between them. This "friendship" in WLAN trace may or may not reflect social friendship, which is impossible to validate from anonymized traces. We propose to identify friendship between MN pairs based on three different dimensions – Encounter duration, encounter count, and encounter AP count, with the following definitions:

[3] One can construct a synthetic trace where a group of people visit several locations in a perfectly staggered cycle. Now while all these users are exactly the same in terms of the location visiting preferences, they never encounter with one another.

- **Friendship based on encounter time** is defined as $Frd_t(a,b) = E_t(a,b)/OT(a)$, which is the ratio of the sum of encounter durations between node a and b, $E_t(a,b)$, to the total online time of node a, $OT(a)$. This is an index for how close node b is to node a based on the duration of encounters. Note that in general $Frd_t(a,b) \neq Frd_t(b,a)$ and $0.0 \leq Frd_t(a,b) \leq 1.0$ for any node pair a and b.
- **Friendship based on encounter count** is defined as $Frd_c(a,b) = E_c(a,b)/S(a)$, which is the ratio between the count of association sessions of node a that contain encounter events with node b, $E_c(a,b)$, to the total association session count of node a, $S(a)$.
- **Friendship based on encounter AP count** is defined as $Frd_{AP}(a,b) = E_{AP}(a,b)/AP(a)$, which is the ratio between the number of APs at which node a has encounter events with b, $E_{AP}(a,b)$, to the total APs node a visits, $AP(a)$.

We first observe how friendship indexes distribute among all node pairs in the traces. As shown in Fig. 6-7, the CCDF curves of friendship indexes based on encounter time follow exponential distributions for all campuses. We again use the Kolmogorov-Smirnov test [98] to examine the quality of fit. The resulting D-statistics for all traces are between 0.0356 and 0.0052, which indicates we have a reasonably good fit between the exponential distribution curves and the empirical distribution curves. The actual parameters we use for the fitting are listed in section 6.8.

The exponential distribution of the friendship indexes is an indication that the majority of nodes do not have tight relationship with one another. In all the traces, only less than 5% of ordered node pairs (a,b) have friendship index $Frd_t(a,b)$ larger than 0.01. This reveals the fact that for node pairs that do encounter with each other, most of them do not show strong relationship. Among all node pairs with non-zero friendship index, only 4.47% of them have friendship index larger than 0.7, and another 11.85% of them with friendship index between 0.4 to 0.7. In other words, we can say that the friendship between the MNs is very "sparse" (i.e., only few pairs of nodes can be called "friends"

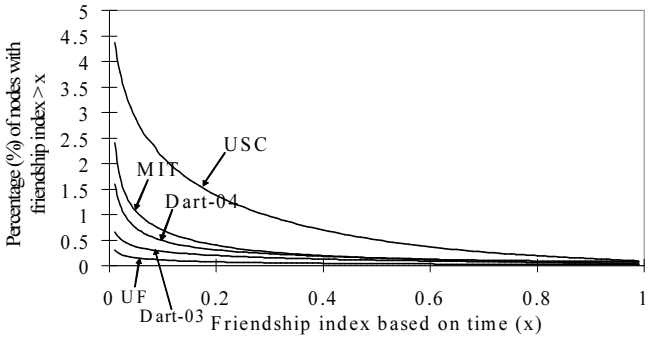


Figure 6-7. CCDF of friendship index based on time.

Table 6-3. Correlation coefficient for friendship indexes for all traces.

Trace name	Friendship index based on		
	encounter time	encounter count	AP count
MIT-rel	0.415	0.327	0.186
UCSD	-0.024	-0.004	-0.003
USC	0.158	0.205	0.130
Dart-03	0.351	0.278	0.043
Dart-04	0.629	0.201	0.068
UF	0.190	0.091	0.036

based on the above definitions). Friendship indexes based on encounter frequency or encounter AP count also show similar exponential distributions.

We also look into the issue of whether the friendship index for an ordered node pair $Frd_t(a, b)$ and the reversed tuple $Frd_t(b, a)$ are symmetric. We calculated the correlation coefficients for all the traces for three definitions of friendship indexes, as shown in Table 6-3. The resulting correlation coefficients between ordered node pair (a, b) and (b, a) are low in most cases (ranging from 0.415 to -0.024, the only exception being 0.629 for friendship index based on encounter time for Dartmouth 2004 trace), implying high asymmetry in friendship indexes.

After seeing the sparseness and high asymmetry of the friendship relationship between the MNs, we ask the following question: if we consider friendship in establishing relationships between nodes, how would that influence the structure of the encounter-relationship graphs? Typically, a MN may not maintain relationships with all other MNs it encounters

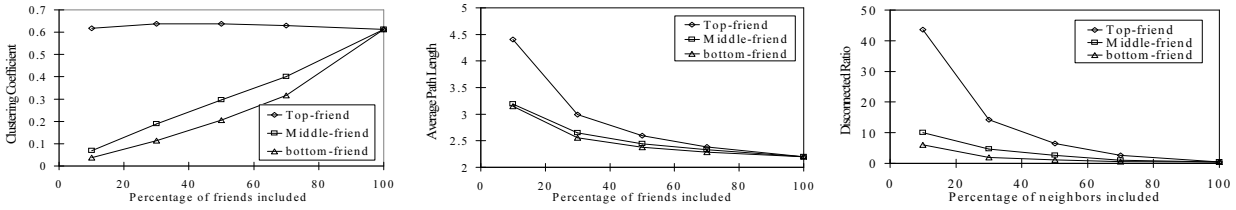
with, but are more likely to maintain connections selectively only with those MNs that are considered "trust-worthy". For example, a MN may choose to trust those MNs with which it has high friendship indexes. The criteria of choosing the nodes to keep a relationship with may influence the structure of the *ER graphs*. To better understand the interplay between the nodal friendship and the resulting ER graph structure, we try to include friends with various degrees of closeness in the *ER graph*, and see how it influences the structure of the graph. We use the friendship index based on time as an example to show how different friendship levels of included links can change the structure of the *ER graph* significantly.

We sort the list of nodes that node a has encountered according to friendship index, $Frd_t(a,b), \forall b \ni Frd_t(a,b) \neq 0$. After sorting, each node picks a certain percentage of nodes from the list with which to establish a link on the *ER graph*. We choose nodes from the top, middle, or bottom of the list and with various percentages, and obtain the corresponding metrics for the new *ER graphs* that include only the links to the chosen nodes. Note that the links in these *ER graphs* are directed links when we consider friendship, as friendship is asymmetric between a given node pair. Therefore, we replace the definition of the clustering coefficient of a node in Eq. (6–2) by the following

$$CC(n) = \frac{\sum_{a \in F(n)} \sum_{b \in F(n)} I(a \in F(b))}{|F(n)| \cdot (|F(n)| - 1)}, \tag{6–8}$$

where $F(n)$ is the set of chosen friends of node n to maintain links with. Note that friendship is an asymmetric relationship, so $b \in F(a)$ does not imply $a \in F(b)$, and vice versa. Intuitively, here the clustering coefficient is the average ratio of the included friends of a node that also include each other as a friend. When calculating the average path length and the disconnection ratio, we follow the same definitions as introduced in section 6.3, but the paths must follow the direction of edges on the ER graph.

Following the above definitions, we obtain the metrics when including given percentages of all encountered nodes from the top, middle, or bottom of the sorted



(a) (b) (c)

Figure 6-8. Metrics of encounter-relationship graph by taking various percentage of friends. (a) Clustering coefficient. (b) Average path length. (c) Disconnected ratio.

node list according to the friendship index based on time. The figures are shown in Fig. 6-8. We use the USC trace as an example, and similar results are also observed in other traces. The figures show a clear trend that if neighbors ranked high in the friendship index are included, the resultant ER graph shows stronger clustering, and the average path length is much higher. The result stems from the fact that top friends of a given node are also likely to be top friend between one another, forming small cliques in the graph. The clustering coefficient remains high due to these cliques. The disconnected ratio and the average path lengths are high due to the lack of links between different cliques. On the other hand, when low-ranked friends are included in the graph, the links included are distributed in a more random fashion, reflected by the low clustering coefficient and low average path length. Similar results are also observed in a social science study of friendship between pupils [99]. As a larger portion of friends are included in the graph, all three metrics converge to the values when all encounters are included[4] .

Therefore, although it is possible to create a campus-wide community based solely on nodal encounters, it is not sufficient to trust and utilize only top-ranked friends (or the MNs one encounters frequently), as this results in an *ER graph* with high clustering

[4] Note that including 100% of friends means to include every MN encountered in the ER graph, hence the resulting ER graph is the same as the one defined earlier in section 6.3).

coefficient and average path length, and may lead to a disconnected network. In order to remain connected to a larger community, one should also use some randomly-chosen users (or middle-ranked friends) as they are the key to reduce the degree of separation in the underlying *ER graph*.

6.6 Information Diffusion using Encounters

In addition to establishing relationship between nodes, encounters can also be utilized to diffuse information throughout the network. In this model, information is spread with nodal mobility and encounters, where nodes exchange information when they encounter each other directly. The speed and reachability of information diffusion among the nodes are determined by the actual pattern and sequences of encounters. In this section we seek to answer the question of whether the *current* encounter patterns between MNs in wireless networks are rich enough to be utilized for information diffusion. If the answer is yes, what is the delay incurred in such a information diffusion scheme, and how robust is it?

In this section, we first understand the optimistic expectation of the potential performance of information diffusion under idealistic assumptions in subsection 6.6.1. We then remove some of the assumptions and evaluate the performance in more realistic settings in subsequent subsections.

6.6.1 Ideal Scenarios

As the first step to understand the potential of information diffusion under realistic encounter patterns, we make the following idealistic assumptions: (1) There are sufficient bandwidth and reliable communication between MNs, and sufficient storage space on all MNs. (2) MNs discover the communication opportunities immediately when they encounter other MNs, and (3) every MN in the network is willing to participate in forwarding information for others. In this experiment, we focus more on analyzing how the encounter pattern itself influences the performance of information diffusion. The experiments in subsection 6.6.2 and 6.6.3 deal with more realistic scenarios when some

of the above assumptions are removed. However, we do not address the technology limitations on the devices itself (i.e., storage capacity, power constraint, etc.).

The diffusion mechanism we use is the following: When a source node has some information to send, it simply transmits it to all nodes it encounters with if they have not received the information yet. All intermediate nodes cooperate in the information diffusion process, keeping a copy of received information and forwarding it the same way as the source node does. This simple approach is known as the epidemic routing in the literature [71]. Under perfect environment with sufficient resources, it achieves the lowest delay and the highest delivery rate possible.

In all the simulations (in this and the subsequent subsections), we use a traffic pattern in which the source node has some information to send to all other nodes. The source starts to "diffuse" the information when it is first online. As time evolves, nodes encounter with each other and an increasing portion of the whole population receive the information. We study the percentage of nodes that have received the information within various trace periods (i.e., the number MNs that have received the message over the total MNs that have appeared during the trace period under discussion) and show the results in Fig. 6-9, using the USC, Dart-04, Dart-03, and MIT traces as examples. Each point in the figures of this section is an average value of multiple experiments. In each experiment we start the information diffusion from a different source node. We choose to use 30% of the nodes that appear the earliest in the corresponding trace period as the sources.

From Fig. 6-9 we observe that even within a short trace period (e.g., two days) the information can reach a moderate portion of the population as the unreachable ratio is less than 25% in all traces. As the trace period increases, reachability also improves. In all except the Dart-03 trace, **the unreachable ratios are less than** 2% **if we allow one month for the information diffusion**. Given that most nodes encounter with only a very small portion of the whole population (Fig. 6-1), this result is perhaps beyond our original expectation. **It gives a positive confirmation that it is potentially possible**

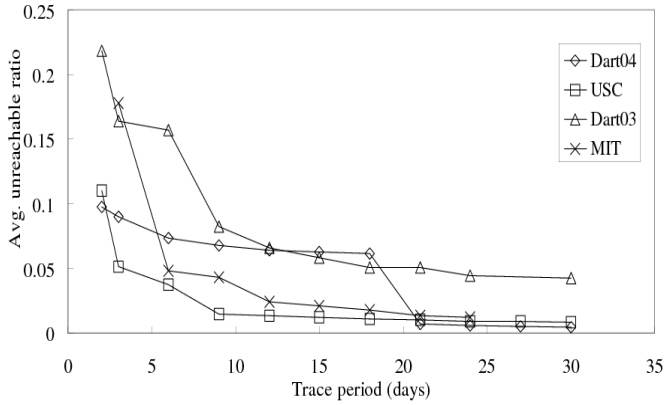


Figure 6-9. Unreachable ratio of information diffusion using the epidemic routing.

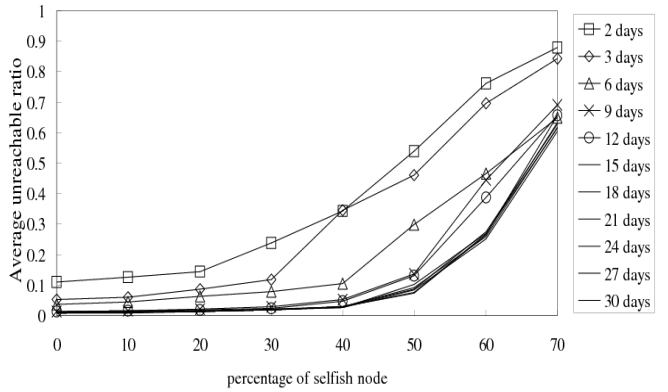


Figure 6-10. USC trace: Unreachable ratio with various selfish node percentage and trace period.

to deliver information relying only on encounters, in a campus environment with high success rate, under *current* user behavioral pattern.

6.6.2 Selfish Users

After studying the ideal case, we consider a more realistic setup. We first relax the ideal assumption (3) above. In some cases, some nodes may not be cooperative to propagate the information. To understand how uncooperative users potentially influence the feasibility of information diffusion, we carry out the following experiment – we make a portion of users *selfish* such that they never forward information for other sources, and we study the performance degradation under this setup. For each of the trace periods used, we increasingly make a certain percentage of nodes selfish, starting from those with the

highest unique encounter counts. By making nodes with high unique encounter counts selfish first, we eliminate more transmission opportunities than if we pick selfish nodes randomly, hence we expect to observe a greater impact on the performance.

The relationship between the percentage of selfish node and the unreachable ratio for the USC trace is shown in Fig. 6-10. For the sake of conciseness, we only show figures for the USC trace here. The figures for other traces display similar trends and they are shown in section 6.9. The result is very surprising – for all trace period tested, the unreachable ratio does not increase significantly before at least 20% of nodes are selfish. The performance is even more robust if we take longer period of trace. This implies that **even a significant portion of users are not willing to propagate information for others, the underlying nodal encounter pattern is rich enough for the information to find an alternative way through**. Hence the delivery rate is quite robust for up to an intermediate percentage of selfish nodes. Note that we make the MNs with most unique encounters selfish first, hence the performance of information diffusion is robust even if the nodes with the *most* chances to propagate the information are not cooperative. We further show how the average delay of information diffusion changes with the increasing selfish node percentage in Fig. 6-11 for the USC trace. In the figure, the average delay increases for longer trace duration because information that is not deliverable in shorter trace periods becomes deliverable. More interestingly, for all tested trace durations, the average delay does not increase significantly before more than 40% of the nodes are selfish. This implies **the average delay is also robust against selfish user behavior up to an intermediate percentage**.

6.6.3 Removal of Short Encounters

Another idealistic assumption we made is that the MNs can communicate with each other successfully regardless of the durations of encounter events. This may not be true in realistic scenario due to wireless bandwidth limitations or delay in discovering encounter events. To address this issue, we remove short-lived encounter events that do

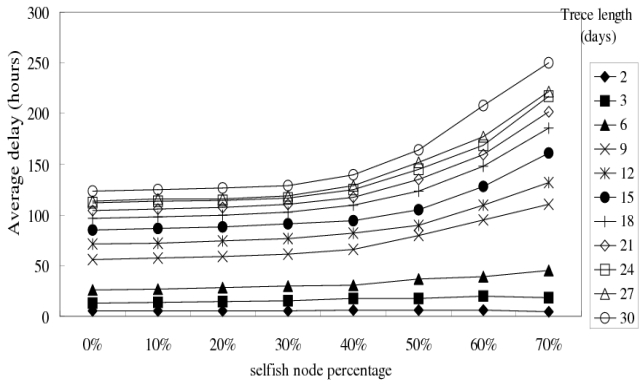


Figure 6-11. USC trace: Average message delay with various selfish node percentage and trace period.

not permit prompt discovery and useful information exchange in the following experiment, and re-evaluate the performance of information diffusion with different minimum duration thresholds for an encounter event to be considered useable.

In Fig. 6-12, we show the relationship between the unreachable ratio versus the lower limit of encounter duration (i.e., we remove all encounter events that have shorter durations than the value), using the first 15-day traces from USC and Dartmouth as examples. From the graph we observe that, the unreachable ratio increases almost linearly as we increase the lower limit of usable encounter duration. There is no obvious point at which the performance suddenly degrades severely. We carry out the experiments up to the shortest usable encounter threshold set at *one hour*, a rather demanding scenario. Even in such cases, besides the UF trace which has a very low encounter ratio (see Fig. 6-1), the unreachable ratio is below 30%. This implies **removing encounters with short durations does not cause abrupt degradation in the performance of information diffusion**, in terms of both the reachability and the average delay (see Fig. 6-13). In other words, short encounters are not the key reason for the success of information diffusion. The encounter events with long durations are also rich enough to be utilized for message propagation in most cases.

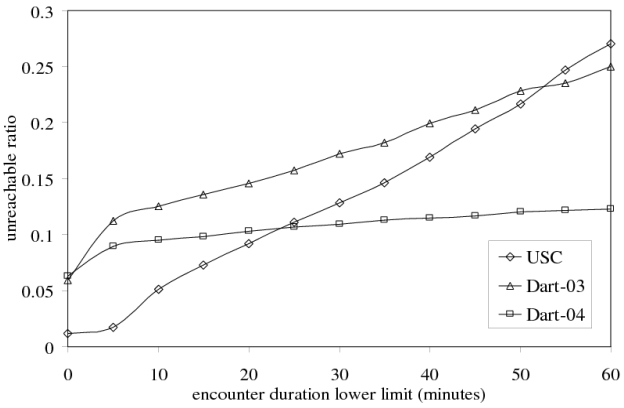


Figure 6-12. The unreachable ratio after removing short encounters under the duration lower limit.

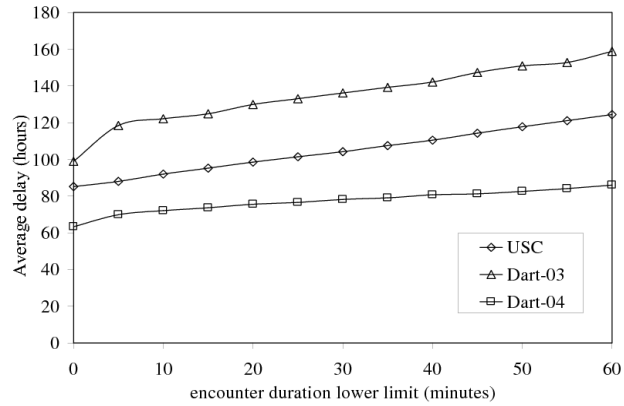


Figure 6-13. The delay after removing short encounters under the duration lower limit.

6.7 Conclusions and Future Work

Our contributions: The contributions of this work are two-folds: First, by investigating the inter-node encounters and utilizing the concept of Small World, we provide new methodologies to understand underlying user behaviors in wireless networks. The understanding gained by studying distributions of encounter events and the *encounter-relationship graphs* reveals how a network can be formed between MNs given their usage pattern in the studied environments. It could be utilized to design better protocols or applications in the future, as we detail in the next chapter. Second, by experimenting information diffusion with current WLAN traces, we display the potential for the

success of information diffusion by the participation of only wireless users (i.e. without infrastructure). We consider these as important findings and they warrant further study.

In this chapter we investigate the encounters between MNs in WLAN traces from four sources. We find that MNs encounter with only a small subset of other nodes (on average between 0.79% to 6.70%), and the total encounter counts follow the BiPareto distribution. In spite of low percentage of unique encounters, the relationship graph constructed using encounters alone connects most of the MNs. Furthermore, such encounter-relationship graphs display Small World graph characteristics, and its graph properties converge to its long-term value within only short time periods. The relationship between different pairs of MNs, however, is very skewed and can be modeled by the exponential distribution. Establishing relationships only with those considered as high-ranked friends leads to a network with high clustering and disconnections, and using low-ranked friends is the key for good reachability in the encounter-relationship graphs. Finally, using simulation study with a simple protocol, we also display the potential for information diffusion without relying on the infrastructure, utilizing encounters and mobility of MNs alone.

The Small World approach to understand the *ER graphs* and the result of information diffusion experiments both highlight positive potential of building a campus-wide network without infrastructures. The robustness of information diffusion brings up two interesting points: (1) For message delivery, the delivery ratio and delay are not affected significantly, even if we can not choose the shortest paths due to non-cooperative users or unutilized short encounters. (2) On the other hand, it would be difficult to prevent diffusion of harmful or malicious messages, such as computer worms or viruses, from propagating through encounters [104]. Both observations are due to the richness in the underlying encounter pattern providing abundant chances for message delivery. The performance of information diffusion under various information delivery schemes and potential methods to prevent malicious information from spreading are both directions for future work.

More specifically, the Small World encounter relationship patterns can be considered as an ambient structure in human networks, and be used to design more efficient message forwarding protocols than the epidemic routing [71] based on which we show the potential of encounter-based information diffusion in this chapter. This task would be our main focus in the next chapter.

6.8 BiPareto Distribution and Kolmogorov-Smirnov Test

In this section we first briefly introduce the Kolmogorov-Smirnov test and the BiPareto distribution, and then list the detail numerical results of fitting BiPareto and exponential distribution curves to total encounter (section 6.2) and friendship index (section 6.5) distributions, respectively.

The BiPareto distribution is used in [101] to fit the number of connections per user TCP session and mean connection inter-arrival time in a TCP session. Later, BiPareto distribution is again used in [14] to fit the distribution of association session length in wireless LAN. The CCDF of BiPareto distribution is as follows:

$$Prob(X > x) = (\frac{x}{k})^{-\alpha}(\frac{x+c}{k+c})^{\alpha-\beta},\ x > k \tag{6–9}$$

$$Prob(X > x) = 1,\ x \leq k \tag{6–10}$$

The left part of the CCDF curve of the BiPareto distribution on log-log scale is a straight line with slope $-\alpha$. As the x variable comes close to the turning point, c, the slope of the CCDF curve gradually changes from $-\alpha$ to $-\beta$. In our study of total encounter distributions, we choose $k = 1$ for all curves.

The Kolmogorov-Smirnov test is used to determine whether the hypothesized distribution (in our case, the BiPareto distribution) adequately fits the empirical distribution. The K-S test is not sensitive to the binning of data set, unlike the Chi-square test[98]. Therefore we choose the K-S test in our study.

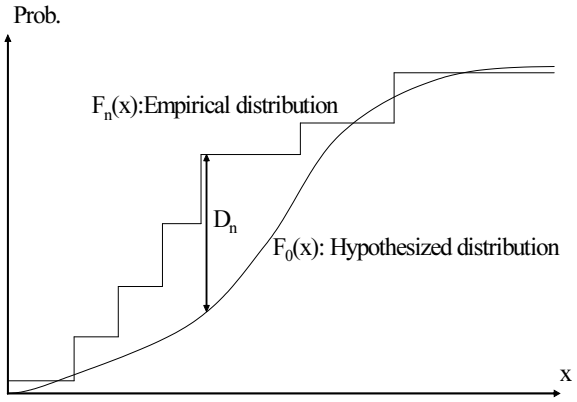


Figure 6-14. Illustration of the D-statistics and the K-S test.

Referring to Fig. 6-14, in the K-S test the distances between the hypothesized distribution and the empirical distribution are measured throughout the range of random variable x, and the maximum of the measured distances is called the D-statistics. More formally, the D-statistics is defined as:

$$D_n = sup_x[|\ F_n(x) - F_0(x)\ |], \tag{6–11}$$

where $F_n(x)$ and $F_0(x)$ are the empirical and hypothesized distributions, respectively. Intuitively, the D-statistics measure the maximum difference between the two distribution curve. A smaller D-statistic indicates a better fit of the hypothesized distribution to the empirical distribution.

We use the minimum squared error method to find the best fit of BiPareto distribution curves to the empirical total encounter distributions for various traces. The parameters are listed in Table 6-4. From the table we observe that the D-statistics are no larger than 0.05 except for UCSD trace (0.07), indicating a reasonable fit of the BiPareto distribution.

We also list the λ parameters we obtained using the minimum squared error method to fit exponential distributions to the empirical distribution of friendship indexes based on encounter time in Table 6-5. The corresponding D-statistics are also listed.

Table 6-4. BiPareto distribution fitting to the total encounter curves and the D-statistics for the K-S test.

Trace name	BiPareto parameters			D-statistics
	α	β	c	
MIT	0.027	9.8	4000	0.036
UCSD	0.062	16.3	9900	0.068
USC	0.019	0.83	550	0.049
Dart-03	0.0723	0.81	290	0.049
Dart-04	0.0285	4.43	11850	0.025
UF	0.1071	1.324	392	0.0066

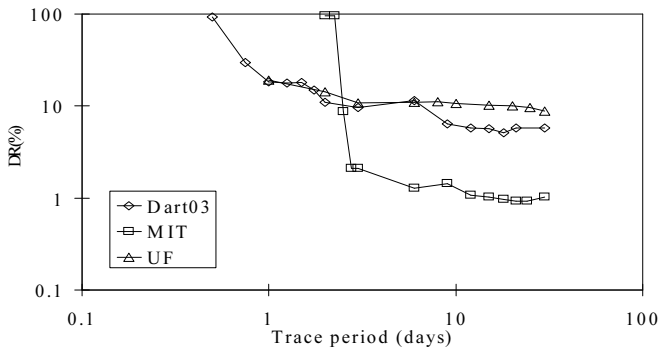
Table 6-5. Exponential distribution fitting to the friendship index based on encounter time curves and the D-statistics for the K-S test

Trace name	λ	D-statistics
MIT-rel	369.19	0.0167
USC	305.3	0.0356
Dart-03	500.4	0.0052
Dart-04	411.81	0.0116
Dart-rel	409.91	0.0120
Dart-cons	412.35	0.0119
UF	579.06	0.0023

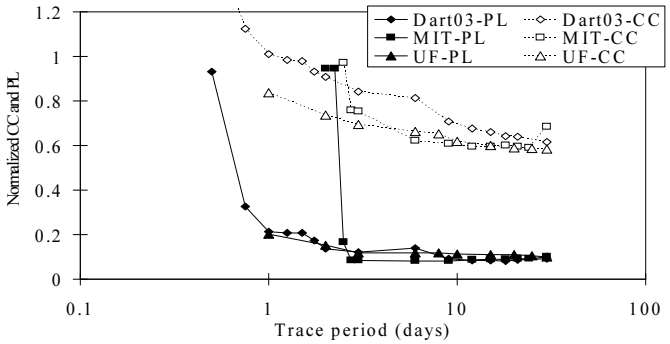
6.9 Additional Experiment Results

In addition to the figures shown in section 6.3, we also obtain the same metrics for MIT, Dart-03, and UF[5] traces. The figures (Fig. 6-15) have similar trends as discussed in section 6.3. One interesting observation here is that for the MIT trace, the disconnected ratio is very high until day 3 in the trace. A further investigation reveals that the MIT trace collection was started on a Saturday, and for a pure working environment (i.e., corporate buildings) Saturdays and Sundays are the least active days. The disconnected ratio is almost 100% until day 3 because the MNs that were on during the weekend are mostly stationary ones. We observe a jump of number of node in the trace, a sudden decrease in *DR*, and an abrupt change in both *CC* and *PL* on day three. For the UF trace,

[5] Due the the large size of the UF data set, the results shown in this section are based on a random sampling of about 30% of users in the trace (We select $10,000$ at random out of $32,695$). Please refer to [112] for more detailed and updated results.



(a)



(b)

Figure 6-15. Change in the *ER graph* metrics with respect to trace period. (a) Disconnected ratio. (b) Normalized clustering coefficient and average path length.

based on the $10,000$ sampled user, the DR for 30-day trace is 8.85%, the normalized CC is 0.584 and the normalized PL is 0.099. We perform full analysis (based on all $32,695$ users that appeared in the 30-day trace) in order to understand the effect of random sampling on the above metrics. The results are as follows: DR 1.94%, CC 0.566, and PL 0.039. It appears the additional users in the full trace lead to a significant decrease in the DR and PL, due to added connectivity, but the CC remains similar. More detailed analysis can be found at [112].

In addition to the USC trace, we further perform similar information diffusion experiments on adding selfish user behavior to the Dartmouth, MIT, and UF traces. The experiment setup is the same as described in subsection 6.6.2. The results for the average unreachable ratio are shown in Fig. 6-16, 6-18, 6-20, and 6-22 for the Dart-04,

MIT, Dart-03, and UF traces, respectively. The trends for the Dart-04 and MIT traces are similar to those shown in subsection 6.6.2. For longer trace periods (above 9 days), the unreachable ratio does not change significantly for up to 20% of selfish nodes, and the robustness of performance increases if longer trace periods are used. This confirms that the robustness of information diffusion under *current* encounter patterns is not an artifact of coarse location granularity in the USC trace. In the Dart-03 and UF traces, the performance of information diffusion is less robust than other traces, since they have the smaller encounter ratio (cf. Fig. 6-1) among all the traces[6] . The unreachable ratio for the Dart-03 and UF traces increases faster as compared to other traces when we make users selfish. The results for the average delay are shown in Fig. 6-17, 6-19, 6-21, and 6-23 for the Dart-04, MIT, Dart-03, and UF traces, respectively. The results are similar to Fig. 6-11 in subsection 6.6.2. One noticeable difference is that, in some cases the average delay first increases as the selfish node percentage increases, but later it decreases. This is due to the low reachability (i.e., high unreachable ratio) – in this situation, only MNs that are easy to reach will be able to receive the message, leading to a decrease in the average delay (calculated from the small subgroup of still reachable MNs).

[6] The resutls for UF trace shown here are based on the $10,000$ sampled users. Analysis for the full trace is in process and will be available at [112].

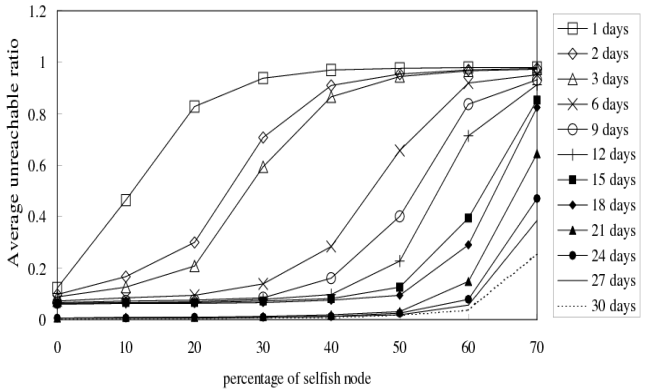


Figure 6-16. Dart-04 trace: Unreachable ratio with various selfish node percentage and trace period.

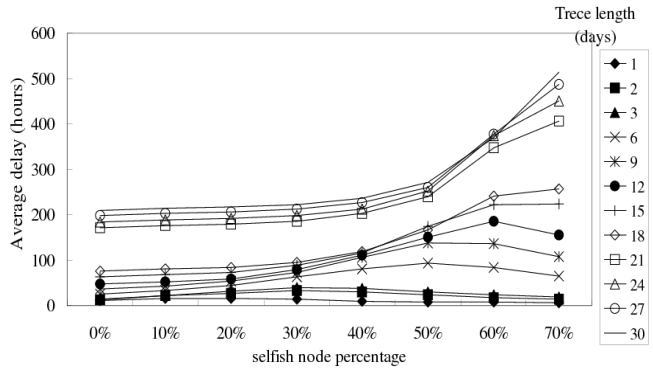


Figure 6-17. Dart-04 trace: Average message delay with various selfish node percentage and trace period.

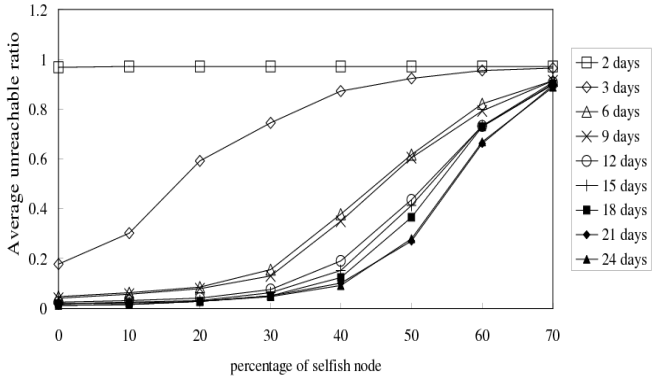


Figure 6-18. MIT trace: Unreachable ratio with various selfish node percentage and trace period.

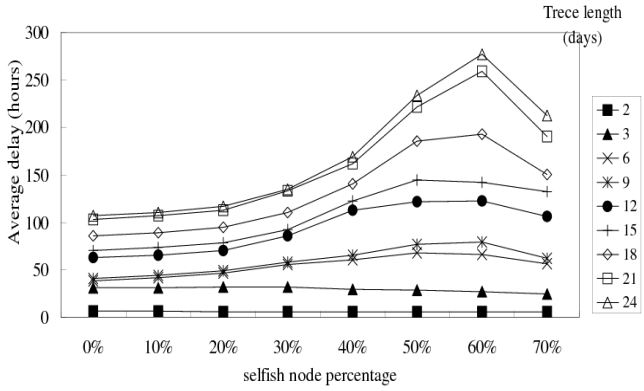


Figure 6-19. MIT trace: Average message delay with various selfish node percentage and trace period.

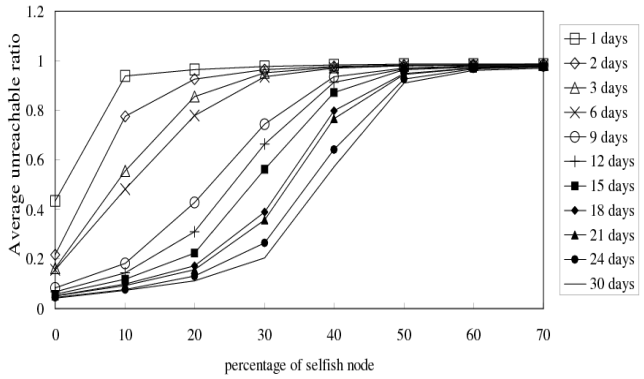


Figure 6-20. Dart-03 trace: Unreachable ratio with various selfish node percentage and trace period.

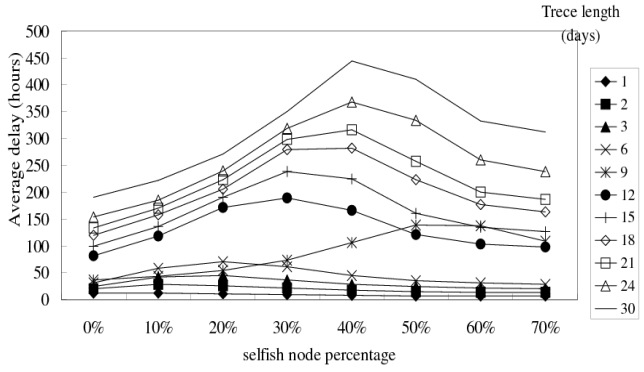


Figure 6-21. Dart-03 trace: Average message delay with various selfish node percentage and trace period.

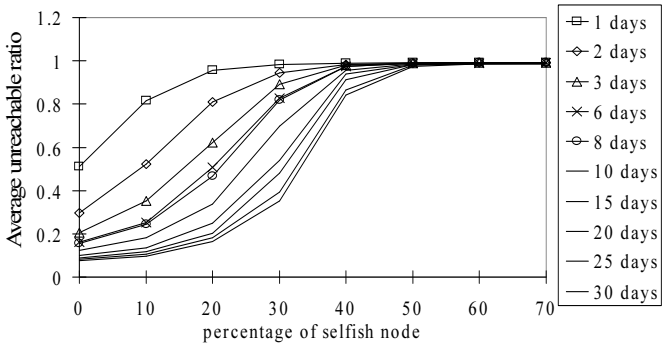


Figure 6-22. UF trace: Unreachable ratio with various selfish node percentage and trace period.

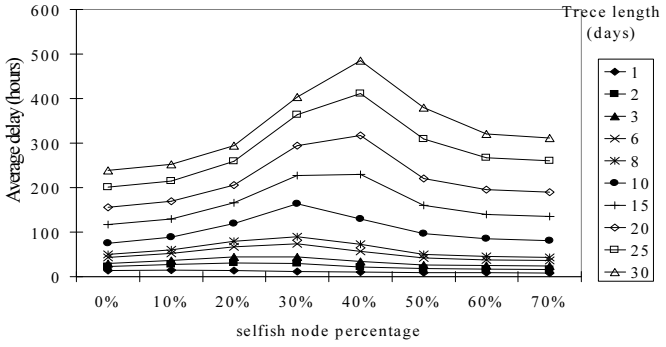


Figure 6-23. UF trace: Average message delay with various selfish node percentage and trace period.

CHAPTER 7
CASE STUDY III: CSI: A PARADIGM FOR BEHAVIOR-ORIENTED DELIVERY SERVICES IN MOBILE HUMAN NETWORKS

In this chapter, we further develop the *profile-cast* paradigm in mobile human networks we first proposed in section 5.8. In such a paradigm, messages are sent to inferred behavioral profiles, instead of explicit IDs. Using *behavioral profile space* gradients and small world structures, we provide fully distributed and more generic message dissemination protocols, named *CSI*, relying on the *I*mplicit yet *S*table relationship discovered between mobile users. The choice of message target in *CSI* is more generic. One can choose a target behavioral profile either with the same representation as the user *eigen-behavior* (i.e., the mobility preferences) or in a totally orthogonal context.

7.1 Introduction

We envision future networks that consist of numerous ultra portable devices delivering highly personalized, context-aware services to mobile users and societies. Such scenarios elicit strong, tight-coupling between user behavior and the network. Users' mobility and on-line activities significantly impact wireless link characteristics and network performance, and at the same time, the network performance can potentially influence user activities and behavior. Such a tight user-network coupling provides a rich set of opportunities and poses several challenges. On one hand, fundamental understanding of the mobile user behavior becomes crucial to the design and analysis of future mobile networks. On the other hand, novel services can now be introduced and utilize such a coupling to effectively navigate mobile societies, providing efficient information dissemination, search and resource discovery.

In this chapter, building on top of the findings in the previous chapter, we propose a novel behavior-driven communication paradigm to enable a new class of services in mobile societies. Current communication paradigms, including unicast and multicast, require explicit identification of destination nodes (through node IDs or group membership protocols), while directory services *map* logical, interest-specific queries into destination

IDs where parties are then connected using interest-oblivious protocols. The power and scalability of such conventional paradigms might be quite limited in the context of future, highly dynamic mobile human networks, where it is desirable in many scenarios to support implicit membership based on interest. In such scenarios, membership in interest-groups is not explicitly expressed by users, it is rather implicitly and autonomously inferred by network protocols based on behavioral profiles. This removes the dependence on third parties (e.g. directory lookup), maintenance of group membership (e.g., in multicast) or the need to flood user interests to the whole network, and minimizes delivery overhead to uninterested users.

Applying such a behavior-driven paradigm in mobile networks poses several research challenges. First, how can user behavior be captured and represented adequately? Second, is user behavior stable enough to enable meaningful prediction of future behavior with a short history? How can such services be provided when the interest or behavior cannot be centrally monitored and processed? And finally, can we design privacy-preserving services in this context?

To address these questions we propose a systematic framework with two phases 1) behavioral profile extraction by analyzing large-scale empirical data sets, investigating the stability of users in the behavioral space, and 2) leverage the behavioral profiles for service design – We use the implicit structure in the human networks to guide message and query dissemination given a target profile.

Specifically, we first analyze network activity traces and design a summary of user *behavioral profiles* based on the *mobility preferences*. The similarity of the *behavioral profile* for a given user to its future profile is high, above 0.75 for eight days and remains above 0.6 for five weeks. The surprising observation is that, the similarity metric between a pair of users predicts their future similarity reasonably well. The correlation coefficient between their current and future similarity metrics is above 0.7 for four days, and remains above 0.5 for fifteen days.

This phenomenon demonstrates that the *behavioral profile* we design is an intrinsic property of a given user and a valid representation of the user for a good period of time into the future. We refer to this phenomenon as the *stability* of user *behavioral profiles*, which can be used to map the users into a high dimensional *behavioral space*. The *behavioral space* is defined as a space where each dimension reflects a particular interest. For example, when we consider mobility preferences, each dimension represents the fraction of time spent at a given location. The position of users in the behavioral space reflects how similar they are with respect to the behavioral profile we construct. We propose a new communication paradigm, in which a *target profile* is used to replace network IDs to indicate the intended receiver(s) of a message (i.e., those with *matching* behavioral profile to the target profile chosen by the sender are the intended receivers.). It is a *C*ommunication paradigm in human networks based on the *S*tability of the user behavioral profile to discover the receivers *I*mplicitly, abbreviated as *CSI*. We present two modes of operation under the over-arching paradigm: the *target mode (CSI:T)* and the *dissemination mode (CSI:D)*. The *target mode* is used when the *target profile* is specified in the same context as the *behavioral profile* (i.e., the *target profile* is in terms of *mobility preferences*). The *dissemination mode*, on the other hand, is used when the *target profile* is de-coupled from mobility preferences.

We show that our CSI schemes perform very close to the delay-optimal schemes assuming global knowledge and improve significantly over the baseline dissemination schemes. For the *CSI:T mode*, comparing with the delay-optimal protocol, our protocol is close in terms of success rate (more than 94%) and has less overhead (less than 84% to the optimal), and the delay is about 40% more. For the *CSI:D mode*, our protocol features lower storage overhead than the delay-optimal protocol with more than 98% success rate – *CSI:D* uses a storage overhead less than 60% of the delay-optimal protocol, while the delay of *CSI:D* is about 32% more than the optimal.

Our contributions:

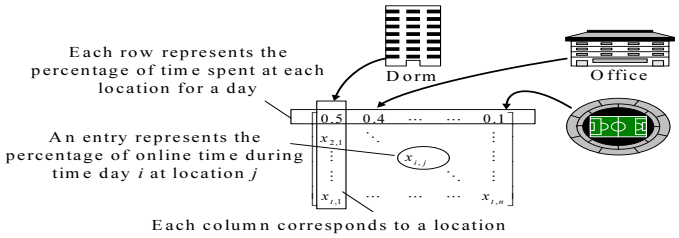


Figure 7-1. Illustration of the association matrix to describe a given user's location visiting preference.

(1) We introduce the notion of multi-dimensional *behavioral space*, and devise a representation of user *behavioral profiles* to map users into the behavioral space. Our study is the first to establish conditions for stability of the relationship between campus users in this space.
(2) We propose *CSI*, a new communication paradigm delivering message based on user profiles. The target profile in CSI can even be independent of the context of behavioral profile we use to construct the *behavioral space*.
(3) We design an efficient dissemination protocol utilizing the stability of behavioral profiles and SmallWorld in mobile societies, then empirically evaluate and validate the efficacy of our proposal using large-scale traces from university campuses.

The outline of the chapter is as follows. We summarize the important background from previous chapters in section 7.2. This is followed by an analysis to understand the user behavioral pattern in section 7.3. We further discuss the potential usages of this understanding in section 7.4 and design our *CSI* schemes in section 7.5 as an example. We use simulations to evaluate the performance of CSI schemes in section 7.6. Finally, we discuss some finer points in section 7.7 and conclude in section 7.8.

7.2 Background

7.2.1 Mobility-Based User Behavior Representation

We represent mobile user behavior of a given user using the *association matrix* as defined in chapter 5 and illustrated in Fig. 7-1. In the matrix, each row vector describes the percentage of time the user spends at each location on a day, reflecting the importance

of the locations to the user[1] . In chapter 5 it has been shown that the *location visiting preferences* can be leveraged to classify users of wireless networks on university campuses. For a given user, the singular value decomposition (SVD) [41] is applied to its *association matrix* M, such that

$$M = U \cdot \Sigma \cdot V^T, \tag{7–1}$$

where a set of *eigen-behavior* vectors, $v_1, v_2, ..., v_{rank(V)}$ that summarize the important trends in the original matrix M can be obtained from matrix V, with corresponding weights $w_{v_1}, w_{v_2}, ..., w_{v_{rank(V)}}$ calculated from the eigen-values in matrix Σ. This set of vectors are referred to as the *behavioral profile* of the particular user, denoted as $BP(M)$, as they summarize the important trends in user M's behavioral pattern. The *behavioral similarity* metric between two users A and B is then defined based on their behavioral profiles (this is the same definition as in Eq. (5–11) but reproduced here for clarity), vectors a_i's and b_j's and the corresponding weights, as

$$Sim(BP(A), BP(B)) = \sum_{i=1}^{rank(A)} \sum_{j=1}^{rank(B)} w_{a_i} w_{b_j} |a_i \cdot b_j|, \tag{7–2}$$

which is essentially the weighted cosine similarity between the two sets of *eigen-behavior* vectors.

7.2.2 Traces

For the study in this chapter, we present results based on two data sets from the University of Southern California (*USC-06spring*) and the Dartmouth College (*Dart-04spring*). The details of the data sets are listed in Table 3-1.

The information available from these anonymized traces contains many aspects of the network usage (e.g., time-location information of the users by tracking the association and disassociation events with the access points, amount of traffic sent/received, etc.).

[1] While there may be numerous other representations of user behavior, we shall show that this representation possesses desirable characteristics for the purposes of this study.

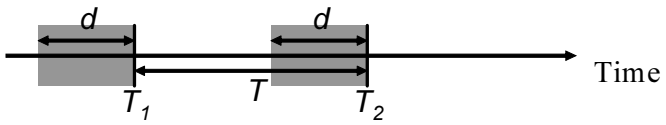


Figure 7-2. Illustration: consider the trailing d days of behavioral profile at time points that are T days apart.

The richness in user behavioral data poses a challenge in *representing* the user behavior in a meaningful way, such that the representation not only reveals an intrinsic, stable behavioral profile of a user, but the identified behavioral profile also leads to practical applications. We show here that the *location visiting preferences* (which is only a subset of the user behavioral data) is a stable attribute for both individual users and the relationship between users. This property will prove quite valuable to the design of efficient message dissemination schemes, which we empirically validate using the above traces.

7.3 Understanding Spatio-Temporal Characteristics of User Behavioral Patterns

In this section we introduce our analysis of user behavioral patterns and its significance on the service design. While previous works on user classification based on long-term behavioral trend [38, 73, 111] are useful and in line with our goal, the stability of such classification over time has not been studied systematically. In particular, the short-term behavior of a user may deviate significantly from the *norm*, and the *stability* of user behavioral profiles is a decisive factor for whether it can be leveraged to represent the user's future behavior. In this section we investigate the following questions: (1) How long of behavioral history do we need to classify a user? and (2) How much does the behavior of a given user and its relationship with other users change with respect to time?

We consider the effect of the amount of past history (of user behavior) on its *behavioral profiles*. Each user uses the location visiting preference vectors in the past d days to summarize the behavior in the most recent history – the user retains d location visiting preference vectors for these days, organize them in a matrix, and use singular

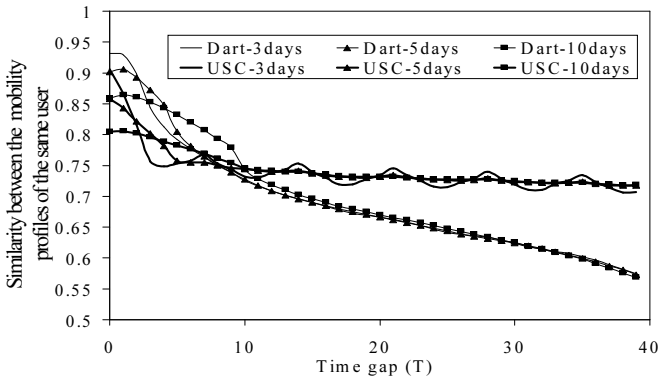


Figure 7-3. Similarity metrics for the same user at time gap T apart.

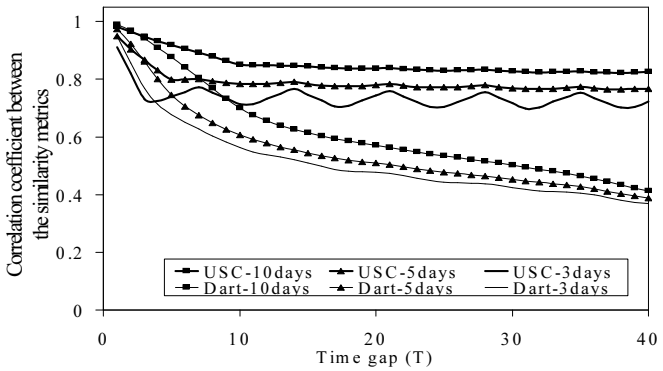


Figure 7-4. Correlation coefficient of the similarity metrics between the same user pair at time gap T apart.

value decomposition to obtain the *behavioral profile*, as described in section 7.2.1. We seek to understand how d influences the representation and similarity calculations. More specifically, we look into two important aspects: (1) Whether the representation of a given user is stable across time, and (2) whether the relationships between user pairs remain stable as time evolves.

We first consider the stability of the representation of a given user. Considering two points in time that are T days apart, we obtain the *behavioral profiles* for the same user at both end points, using the logs of the trailing d days ending at those end points, as illustrated in Fig. 7-2. Then we use the similarity metric defined in Eq. (7–2) to compare how stable a user's behavioral profile is to one's former self after T days has elapsed. The average results with various values of the time gap, T, and considered behavioral history

d are shown in Fig. 7-3. We notice that, even if we collect a short history of user behavior (say $d = 3$), the representation is similar to the behavior of the user for a long time into the future. When we consider $T = 35$ days apart, the behavioral profiles from the same user still show high similarity, at about 0.6. The amount of history used does not influence the result too much when the considered T is large enough to avoid overlaps in the used behavioral history (i.e., when $T > d$). We conclude that on university campuses, the *behavioral profile* for a given user is stable, i.e., it remains highly similar for the same user across time. One interesting note is that, when the behavioral profile includes only part of a week ($d < 7$), the similarity of the user to its former self shows a weekly pattern (i.e., when T is an integer multiple of seven, the similarity peaks), especially in USC.

Second, we try to quantify how the behavioral similarity between the same pair of users varies with time. For this part, we use Eq. (7–2) to calculate the similarity between two users, A and B, at two points in time, $Sim_{T_1}(A, B)$ and $Sim_{T_2}(A, B)$, where T_1 and T_2 are T days apart. We perform this calculation to all user pairs, and then calculate the correlation coefficient of the similarity metrics obtained after a T-day interval, as

$$r = \frac{\sum_{\forall A,B}(X - \overline{X})(Y - \overline{Y})}{NS_X S_Y}, \tag{7–3}$$

where $X = Sim_{T_1}(A, B)$ and $Y = Sim_{T_2}(A, B)$, and the notations $\overline{X}$ and S_X denote the average and standard deviation of X, respectively. N is the total number of user pairs. The correlation coefficient quantifies how stable the relationship between user pairs is. We repeat the calculation for all pairs of users with various d and T values to arrive at Fig. 7-4. We observe that the similarity metrics between user pairs correlate reasonably well if the considered time periods are not far apart. For T smaller than one week, the correlation coefficient is above 0.62. This indicates, once the similarity between a pair of user is obtained, it remains a reasonable predictor for their mutual relationship for some time period into the future. Although the reliability of the stale similarity data decreases with respect to time, the current similarity of a user pair remains moderately correlated to

their future similarity, in the time range up to several weeks. The correlation is above 0.4 for up to five weeks.

This investigation establishes that the user behavioral profile is a stable feature to represent the users – the representation of an individual user and the relationship between users are well correlated with the past history for the near future. Thus we map the behavioral profile to a virtual *behavioral space* [61], in which each user's behavior is quantified as a high dimensional point[2] . The mutual similarity metric between users is a function of their respective positions in this space. In the following sections, when we say two users are *similar*, it means they are *close* in the behavioral space (i.e., the *distance* between the two users is small). We also use the term *neighborhood of a node* to refer to the other nodes that are *similar* to this particular node in the behavioral space.

7.4 The Behavior-Driven Communication Paradigm

Profiling users based on stable behaviors is a fundamental step to understand human behavior. Motivated by the stability of user behavioral profiles, we introduce a *behavior-driven communication paradigm* where we use *user behavioral profiles*, instead of network IDs, to represent users. We envision that such a radical approach has several benefits.

First, it enables behavior-aware message delivery in the network without mapping attributes to network IDs. As each user maintains its behavioral profile, it is now possible to deliver announcements about a sports event on campus towards sports enthusiasts (e.g., people who visit the gym often) or advertise a performance at the school auditorium to the regular attendees of such events.

Second, it facilitates the discovery of nodes with certain behavior patterns. Consider, for example, in the message ferry [108] architecture where nodes with high mobility

[2] The dimension of the behavioral space is the same as the *mobility preference vector* representation, typically in the order of a hundred for these two campuses.

move messages across the network to facilitate the communication between otherwise disconnected nodes. One can choose a target profile that reflects a mobility profile and thus eliminate the need of knowing the identity of the ferry beforehand or enforcing this mobility pattern on a controlled node – a typical user who happens to have the desired mobility pattern can be discovered and serves as a ferry.

Our *behavior-driven communication paradigm* is applicable to several architectures. In the *centralized server-based architecture*, user profiles could be collected and stored at a data repository, and mined for user classification, abnormality detection, or targeted advertisements. In the *cellular networks*, the low-bandwidth channel between the users and the infrastructure can be leveraged to exchange behavioral profiles and match users. In this dissertation, however, we mainly consider *decentralized infrastructure-less networks*, and focus on how stable behavioral profiles are used for better message dissemination. We name this scheme as *CSI*, since it is a *C*ommunication scheme based on the *S*table, *I*mplicit structure in human networks.

7.5 Protocol Design

In this section, we first present our premises and design requirements for the CSI schemes. We then discuss the design of the CSI schemes based on in-depth understanding of the relationship between similar behavioral profiles and encounter events.

7.5.1 Assumptions and Design Requirements

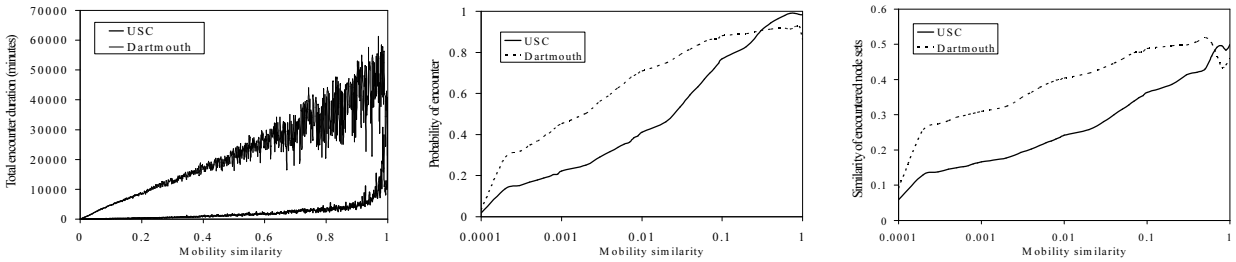
We assume that each node profiles *its own behavioral pattern* by keeping track of the visiting durations of different locations and summarizing the behavioral profile using the technique discussed in 7.2.1. This is an individual effort by each node involving no inter-node interactions. This can be done by the nodes over-hearing the beacon signals from the fixed access points in the environment to find out its current location. Note that, the use of these beacon signals is only for the node to profile its own behavior – they are not used to help the communication in our protocols (we will re-visit detailed points of this assumption in section 7.7). Also, for the ease of understanding, we assume in this

section that nodes are willing to send its behavioral profiles to other nodes when needed. A privacy-preserving option that eliminates this operation is also discussed in section 7.7.

The goal of our *CSI* scheme is to reach a group of nodes matching with the target profile specified by the sender, under the following performance requirements: (1) The protocol should be scalable, in particular not being dependent on a centralized directory to map target profiles to user identities. (2) It should work in an efficient manner and avoid transmission and storage overhead when possible. Also, it should avoid control message exchanges in the absence of data traffic. (3) The syntax of the target profile should be flexible, allowing the target profile to be not in the same context as the behavioral profiles we use to represent the users. Also the operation of the protocol should be flexible to allow tradeoff between various performance metrics. And finally, (4) the design should be robust and help in protecting user privacy.

We design two modes of operation for the *CSI* scheme under the above requirements. When the target profile is in the same context as the behavioral profile (in our example, since the behavioral profile is a summary of user mobility, this corresponds to the scenario when the target profile describes users that *move* in a particular way), the *CSI:Target mode (CSI:T)* should be used. When the target profile is irrelevant to the behavioral profile (e.g., when I want to send to everyone interested in movies on campus), the *CSI:D mode* should be used instead. Although it seems that the applicability of *CSI:T* is limited, we note that the behavioral profile (in terms mobility) can sometimes be used to infer other social aspects of the users, such as affiliations or even interests (e.g., people who visit the gym often should like sports in general). Such inferences expand the scenarios in which *CSI:T* can be used. When this is not possible, *CSI:Dissemination mode (CSI:D)* provides a more generic option.

The major challenge involved in the design process is that each node is only aware of the behavioral profile of itself. Furthermore, we require no persistent control message exchanges for the nodes to "learn" the structure of the network proactively when they



(a) (b) (c)

Figure 7-5. Relationship between the similarity in behavioral pattern and other quantities. (a) Total encounter duration. (b) Encounter probability. (c) Similarity of encountered node sets.

have no message to send. Nodes only compare their behavioral profiles *when they are involved in message dissemination*. Based on this very limited knowledge about the behavioral space, a node must predict how useful a given encounter opportunity is in terms of achieving the fore-mentioned requirements. Since encounter events may occur sporadically in sparse, opportunistic networks, the nodes must make this decision for each encounter event independent of other encounter events (that may occur long before or after the current one under consideration). Such a heuristic must rely on the understanding of the relationship between nodal behavioral profiles and encounters, which we discuss the next.

7.5.2 Relationship between Behavioral Profiles and Encounters

We now analyze the relationship between user behavioral profiles and a key event for user-to-user communication in an infrastructure-less network – *encounters*. While it seems intuitive that users visiting similar locations should encounter with each other with higher probability, this is *not obvious* on university campuses. Students and faculty have their own schedules, and they may rarely encounter due to the difference in their schedules although they might be in the same building at different times. Hence we investigate the relationship between behavioral profiles and encounter events, first as a sanity check of our intuition, and more importantly, to understand the relationship between the behavioral profiles and various aspects of the encounter events (e.g., the encounter probabilities,

encounter durations, etc.). This helps reveal the *implicit structure* existing in mobile human networks, which is the key to the design of the *CSI* schemes in the following sections.

We classify all node pairs into different bins of behavioral similarity metric (as defined in Eq. (7–2)), and obtain various characteristics of encounter events as a function of the pair-wise behavioral similarity. In Fig. 7-5 (a), we show the aggregate encounter time duration between an average pair of nodes given the behavioral similarity. In Fig. 7-5 (b), we show the probability for a given node pair to encounter with each other, given their similarity. Combining these two graphs, we see that **if two users are similar in behavioral profiles, they are much more likely to encounter, and the total time they encounter with each other is much longer – an indication that nodes with** *similar behavioral profiles* **indeed are more likely to have better opportunities to communicate directly.** When two users are similar enough (with behavioral similarity larger than 0.3), they are almost guaranteed to encounter at some point (with probability above 0.9). However, we note that some "random" encounter events happen between dissimilar users. For users with very low (almost zero) similarity, the probability for them to encounter is not zero, although such encounter events are much less reliable (i.e., they occur with much shorter durations, see Fig. 7-5 (a)).

In Fig. 7-5 (c) we further compare the behavioral similarity of node A and B versus the sets of nodes A and B encounter. We denote the set of nodes A encounters with as $E(A)$. The similarity of the two sets of nodes is quantified by $|E(A) \cap E(B)|/|E(A) \cup E(B)|$, where $|\cdot|$ is the cardinality of the set. This graph shows, **as two nodes are increasingly similar, there is larger intersection of nodes they encounter. When an unlikely encounter event between** *dissimilar nodes* **occurs, it helps both nodes to gain access to a very different set of nodes, which they are unlikely to encounter directly.**

The above findings relate to the SmallWorld encounter patterns between mobile users [60] we discuss in the previous chapter. The key features of SmallWorld networks [8] are high clustering coefficient and low average path length. In the human networks we analyze in this section, people with similar behavior form "cliques". The "random" encounter events between dissimilar nodes build *short-cuts* between these cliques to shorten the distances between any two nodes. We leverage these properties in the protocol design.

7.5.3 CSI: Target Mode

In the *CSI:target mode (CSI:T)*, the sender specifies the *target profile (TP)* for the recipients which must have the same format and semantics as that of the user behavioral profile, i.e., in our case the TP is a summarized *mobility preference* vector (i.e., the percentage of times the target node(s) visit various locations). For example, we could reach people who like sports by sending messages to those who visit the gym regularly. This criteria could be set up by specifying the TP as a vector with only one 1 corresponding to the gym location (hence only time spent at this location is considered). If a given user A has $Sim(BP(A), TP) > th_{sim}$, i.e., its behavioral profile, $BP(A)$, is more similar to TP than a sender specified threshold, we say node A belongs to the group of *intended receivers*. This threshold is set by the sender according to the desired degree of similarity to the TP. The TP and the threshold, th_{sim}, are included in the message header to describe the intended receivers of the message.

We first discuss the intuition behind the design of the *CSI:T mode* using Fig. 7-6 as an illustration. As per section 7.5.2, to deliver messages to receivers defined by a given TP, one way is to gradually move the message towards nodes with increasing *similarity* to the TP via encounters, in the hope that such transmissions will improve the probability of encountering the intended receivers. Finally, when the message reaches a node *close* to the TP (in the behavioral space), most nodes encounter frequently with this node are also

similar to *TP*. Hence, the message should be spread to other nodes in the *neighborhood* (in the behavioral space) of the node.

Consider the pseudo-code in Algorithm 1. There are two phases in the operation, the *gradient ascend phase* and the *group spread phase*. (1) Starting from the sender, if node A currently holding the message is not an intended receiver (i.e., $Sim(BP(A), TP) < th_{sim}$), it works in the *gradient ascend phase*, otherwise it works in the *group spread phase*. (2) In the *gradient ascend phase*, for each encountered node, the current message holder asks the behavioral profile of the other node, and if the other node is more similar to the TP in the behavioral space, the responsibility of forwarding the message is passed to this node. One can imagine that these similarities form an inherent *gradient* for the message to follow and reach the close neighborhood of the TP in the behavioral space, hence the name *gradient ascend phase*. Note that, up to this point, there is only one copy of the message in the network – these intermediate nodes who are not similar to the TP only forward the message once. (3) When the message reaches a node with similarity larger than th_{sim} to the *TP*, the *group spread phase* starts. This intended receiver holds on to the message, and requests the behavioral profiles from nodes it encounters. If they are also intended receivers, copies of the messages will be delivered to them. All intended receivers, after getting the message, continue to work in the *group spread phase*. Although multiple copies of the message are generated in the *group spread phase*, it is triggered only when the message is close to the *TP*, thus most of the encounter events and inquiries will occur among the *intended receivers*, reducing unnecessary overhead.

7.5.4 CSI: Dissemination Mode

In the *CSI:Dissemination mode (CSI:D)*, there does not exist a direct relationship between the target profiles of the recipients and their measured behavioral profiles. One particular example is to reach people who like movies on campus. If there is no movie theaters on campus, the measured behavioral profiles (i.e., mobility preference) cannot be used to infer such an interest. This situation is illustrated in Fig. 7-7. It appears

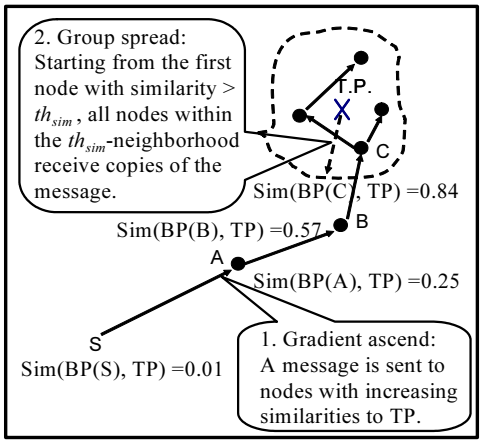


Figure 7-6. Illustration of the CSI:T scheme in the *high dimension behavioral space*. One copy of the message follows increasing similarity gradient to reach the neighborhood of the target profile, then triggers group spread.

```
/* BP(A): Behavioral profile of node A                                    */
if node A has the message then
    if Sim(BP(A), TP) > th_sim then
        Initiate Group_spread();
    else
        Initiate Gradient_ascend();
Gradient_ascend(){
while the message is not sent do
    foreach node E encountered do
        Get BP(E) from E;
        if Sim(BP(E), TP) > Sim(BP(A), TP) then
            Send message to E;
}
Group_spread(){
foreach node E encountered do
    Get BP(E) from E;
    if Sim(BP(E), TP) > th_sim then
        Send message to E;
}
```

Algorithm 1: Algorithm for the CSI:T mode

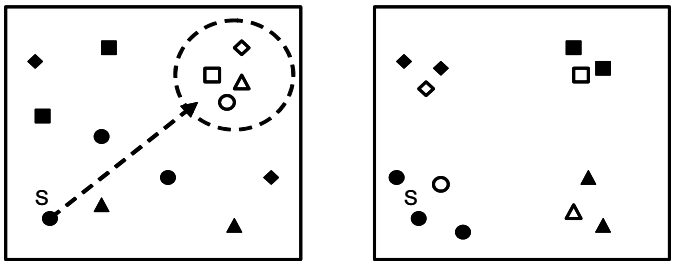


Figure 7-7. Design philosophy of the *CSI:D* scheme. Left chart: The goal is to send a message to a group of nodes with a similar characteristic in the *interest space* (white nodes in the circle). Right chart: However, they may not be similar to each other in the behavioral space (nodes with the same legend represent similar nodes in the behavioral space).

there is little insight provided by the similarities between the nodal behavioral profiles to guide message propagation, as the intended receivers in this case may be scattered in the behavioral space, and the relationship between the target profile and the behavioral profile cannot be quantified. Although it is always possible to reach most users through epidemic routing, as we have shown its robustness in the previous chapter (see section 6.6), this leads to high overhead, and requires all nodes in the network to keep a copy of the message. The objective of *CSI:D mode* is to reduce the numbers of message copies transmitted and stored in the network, yet make it possible for most nodes to get a copy quickly, if they belong to the intended receivers.

We again first discuss the intuition behind the design of the *CSI:D mode* in this paragraph, using Fig. 7-8 as an illustration. From section 7.5.2, **since the nodes with high similarity in their behavioral profiles are almost guaranteed to encounter, there is really no need for each of them to keep a copy and disseminate the message. Electing a few** *message holders* **within a single group of similar nodes would suffice.** This intuition leads to the construction of our message dissemination strategy for the *CSI:D*. We aim to have only one *message holder* among the nodes who are similar in their behavioral profiles (or equivalently, pick only one *message holder* within a *neighborhood* in the behavioral space. In Fig. 7-7, this corresponds to having only one

message holder among each group of nodes with the same legend). We add the messages holders carefully to avoid overlaps in the encountered nodes among message holders. As suggested by Fig. 7-5 (c), we should **select nodes that are very** *dissimilar* **in their behavioral profiles to achieve low overlaps.** Recall that dissimilar node pairs still encounter with non-zero probability, our design philosophy is to leverage these "random" encounter events as *short-cuts* to navigate through the behavioral space efficiently, hopping across the space to reach dissimilar nodes with relatively few message transmissions. Such a design philosophy is also related to the SmallWorld human network structure – a message will be received by an intended receiver shortly once it has reached someone in the receiver's "clique".

Consider the pseudo-code in Algorithm 1. (1) The sender itself starts as the first message holder in the network. (2) Each message holder tries to strategically add additional message holders in the network. When it encounters with other nodes, it asks for the behavioral profile of the other node to be considered as a potential additional message holder. Each message holder keeps a list of the behavioral profiles of all known message holders[3] , and the new node has to be dissimilar (with the similarity metric lower than a threshold, th_{fwd}) to all known holders to be added as a new message holder and keep another full copy of the message. (3) If, on the other hand, this node is similar to the message holder (i.e., within similarity threshold th_{nbr}), it uses a single bit to remember that there is a message holder in its neighborhood and propagates this information to similar nodes. This bit is used to prevent excessive message holders in the same neighborhood, even if some nodes have not encountered with the message holders directly. (4) When holders encounter, they update each other with the behavioral profiles of the

[3] Note this list does not necessarily contain all holders in the network. Message holders that are added by a particular message holder are not known to other holders until they meet and sync the lists.

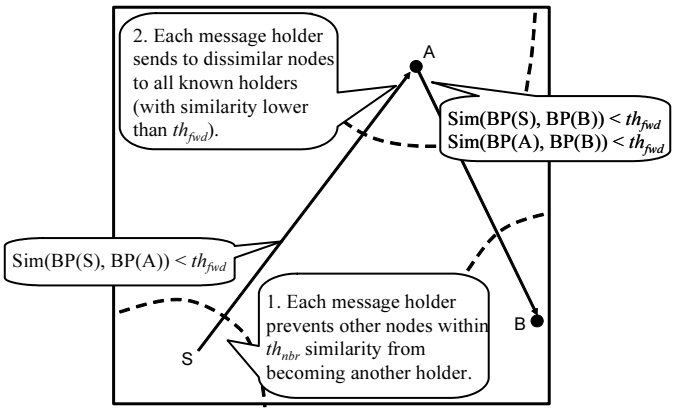


Figure 7-8. Illustration of the CSI:D scheme. The idea is to select the message holders in a non-overlapping fashion to cover the entire behavioral space.

known holders list, to gain a better view of the situation of message spreading. (5) If two similar holders (i.e., when their similarity metric is above the threshold th_{nbr}) encounter, one of them should cease to be a holder to reduce duplicated efforts.

Each message holder is responsible for disseminating the actual message to the intended receivers. The message holders sends the *TP* specified by the sender in the message to the encountered nodes. If the encountered node is an intended receiver, the full message will be transferred.

7.6 Simulation Results

In this section, we perform extensive simulations with the CSI schemes, based on the derived encounters between users from the *USC-06spring* and *Dart-04spring* traces. We compare the performances of our proposal to oracle-based forwarding decisions to show that our performance is close to the optimum (in terms of the delivery success rate and the overhead), and does not fall much behind in delay. We also compare CSI to epidemic routing [71] and variants of random walk[4] . In all the simulation cases, we split the traces

[4] The CSI could not be directly compared with existing routing schemes (e.g., [61, 74, 76, 107]) in DTN as most of them have a different routing objective: reaching a particular network ID.

```
/* BP(A): Behavioral profile of node A                                    */
/* H_i(A): The i-th known holder of node A                                */
/* holder_in_group(A): If A knows there is a message holder in its
   neighborhood                                                           */
if node A is a message holder then
    foreach node E encountered do
        Get BP(E);
        if E is not a holder then
            if Sim(BP(E), BP(H_i(A))) < th_fwd ∀i and holder_in_group(E) = false
            then
                Elect E as an holder;
                Add BP(E) to holder list;
                Send the message;
                Send BP(H_i(A)), ∀i;
            else if Sim(BP(E), BP(H_i(A))) > th_nbr for any i then
                Let E set holder_in_group(E) = true;
        else
            if Sim(BP(E), BP(A)) > th_nbr then
                A ceases to be a holder;
            else
                Sync holder lists between node A and E;
else if holder_in_group(A) = true then
    foreach node E encountered do
        Get BP(E);
        if Sim(BP(A), BP(E)) > th_nbr then
            Let E set holder_in_group(E) = true;
```

Algorithm 2: Algorithm for CSI:D mode.

into two halves, use the first half to obtain the behavioral profiles for all users, and then use the second half of the trace to evaluate the success of our proposed schemes.

7.6.1 CSI: Target Mode

7.6.1.1 Simulation setup

In the scenario of CSI:T mode, the sender specifies the TP and a threshold of similarity th_{sim}. If a node shows a similarity metric higher than th_{sim} to the TP, it is an

intended receiver. In our evaluation, we use the top-10 dominant behavioral profile[5] (i.e., the behavioral profiles with the most number of people following it, typically in the order of hundreds) in our traces as the *TP*, and for each *TP* we randomly pick 100 users as the senders generating messages targeting at the *TP*. We use the threshold $th_{sim} = 0.8$ as the transition point between the *gradient ascend phase* and the *group spread phase*.

We compare our *CSI:T* scheme with several other protocols discussed below. The *epidemic routing* [71] is a message dissemination scheme with simplistic decision rules: all nodes in the network send copies of messages to all the encountered nodes who have not received the message yet. The *random walk (RW)* protocol generates several copies of the message from the sender, and each copy is transferred among the nodes in a random fashion, until the hop count reaches a pre-set TTL value. *Group spread only* is a simplified version of our protocol. It uses only the *group spread phase*, i.e., the original sender holds on to the message until it encounters with someone who is more similar than th_{sim} to the *TP* and starts the *group spread phase* directly from there.

We also consider three protocols that require global knowledge of the future. The *delay-optimal* protocol sends copies of the message only to the nodes which lead to the fastest delivery to the targeted receivers, and no one else. This is the oracle-based optimal protocol achievable if one has perfect knowledge of the future, and serves as the upper bound for performance. The *overhead-optimal* protocol, on the other hand, optimizes (i.e., minimizes) the number of transmission counts using the knowledge of future encounter events. This protocol delivers messages to all reachable receivers under the minimum possible transmission count. The pseudo-code we use for these two optimal protocols based on complete knowledge of all encounter events is summarized in Algorithm 3. Notice

[5] We have also experimented with other target profiles, such as rarely visited locations on campuses or profiles that contain a combination of several locations, and the results are similar to those presented in this section.

this is basically a generalized version of the Dijkstra algorithm, with different metrics used in either protocol. More specifically, for the *delay-optimal* protocol, the metric to be considered is the delay (i.e., the reach time at each node subtracts message send time); for the *overhead-optimal* protocol, the metric to be considered is the hop count to reach the node.

The *optimal single-forwarding-path* is the oracle-based protocol to find the fastest path to deliver the message to the neighborhood of the TP – Using the knowledge of the future, it identifies the path that leads to the earliest message delivery to any of the intended receivers. That is, we use the results from the *delay-optimal* protocol, identify the node that receives the message the earliest among all intended receivers, and find the path taken from the sender to reach this particular node. The *optimal single-forwarding-path* then uses this path to deliver one copy of the message to the neighborhood of the intended receiver group. Once a copy of the message is delivered to the th_{sim}-neighborhood to the TP, it follows the same *group spread phase* as in CSI:T. This is the optimal performance (upper bound) for the family of protocols delivering one copy of message to the neighborhood of the target profile, if one chooses a good (shortest delay) path – note that this shortest-delay path may not always follow an increasing gradient of similarities to the TP.

We compare these message dissemination schemes with respect to three important performance metrics: *delivery ratio*, *average delay*, and *transmission overhead*. The *delivery ratio* is defined as the percentage of the intended receivers (those with similarity greater than th_{sim} to the TP) actually received the message. We account for the transmission overhead as *the total number of messages sent* in the process of delivery. See more discussions on the additional overhead of exchanging the behavioral profiles later in section 7.7.1.

```
/* done[i]: if the metric for node i is finalized                    */
/* metric[i]: The current best metric to reach node i                */
/* from[i]: the previous hop of node i                               */
/* reach_time[i]: the time node i receives the message               */
/* s: the source node                                                */
/* candidate: current node under consideration, from which all other
     "unfinished" nodes could potentially improve the metric         */
forall Node i do
    set done[i] = false ;
    set metric[i] = inf. ;
    set from[i] = null ;
    set reach_time[i] = inf. ;
set done[s] = true ;
set metric[s] = 0 ;
set reach_time[i] = sendtime ;
set candidate = s ;
while candidate ≠ null do
    foreach node k that done[k] = false do
        foreach Encounter event after reach_time[candidate] between candidate and
        k do
            if Message delivery from candidate to k improves (reduces) metric[k]
            then
                Modify metric[k] ;
                set reach_time[k] = Encounter_event_time ;
                set from[k] = candidate ;
    forall Node k such that done[k] = false and metric[k] ≠ inf. do
        Find node m with minimum metric[m] ;
        if m ≠ null then
            set candidate = m ;
            set done[m] = true ;
        else
            set candidate = null ;
```

Algorithm 3: Algorithm for the oracle-based optimal protocols.

7.6.1.2 Simulation results

We show the normalized performance metrics with respect to that of *epidemic routing* (the relative performance for each protocol assuming *epidemic routing* is 1.0) and its 95% confidence intervals in Fig. 7-9. We observe that *epidemic routing* leads to the highest overhead while its aggressiveness also results in the highest possible delivery ratio and the lowest possible delay. The *random walks* do not work well regardless the number of copies and the value of TTL, as they use no information to guide the propagation of the message towards the right direction. Our *CSI:T* protocol leads to a success rate close to the *epidemic routing* (0.96 for USC, 0.94 for Dartmouth) with very small overhead (0.02 for USC, 0.018 for Dartmouth). For the simplified version, *group spread only*, the delay is longer and the success rate is lower than our protocol. We will further investigate this phenomenon later.

When comparing *CSI:T* with the optimal protocols with future knowledge, we see that there is really not much room for improvement in terms of the success rate and the overhead. Our gradient ascend approach in *CSI:T* is similar to what is achievable even one has the knowledge of the future in these two aspects. Specifically, *CSI:T* has more than 94% of delivery rate and uses *less than* 84% overhead of the *delay-optimal* strategy. When comparing with the *overhead-optimal* protocol, we observe that the overhead *CSI:T* incurs is about the same (with less than 5% difference) to the *overhead-optimal* protocol, and the delay is less in the USC case (by 20%) but slightly more in the Dartmouth case (by 11%). We can therefore conclude that our *CSI:T* protocol does well in terms of overhead and delivery rate, even compared to the optimal protocols with perfect information of the intended receivers and future encounter events. The delay, on the other hand, has some room for improvement. The key reason of this difference (in terms of delay) is that our gradient ascend phase generates only one copy of message from the sender and it moves towards the TP following strictly ascending similarity. Comparing with the best (fastest) path to the TP used in the *optimal single-forwarding-path*, our *CSI:T* has 1.40 and 1.47

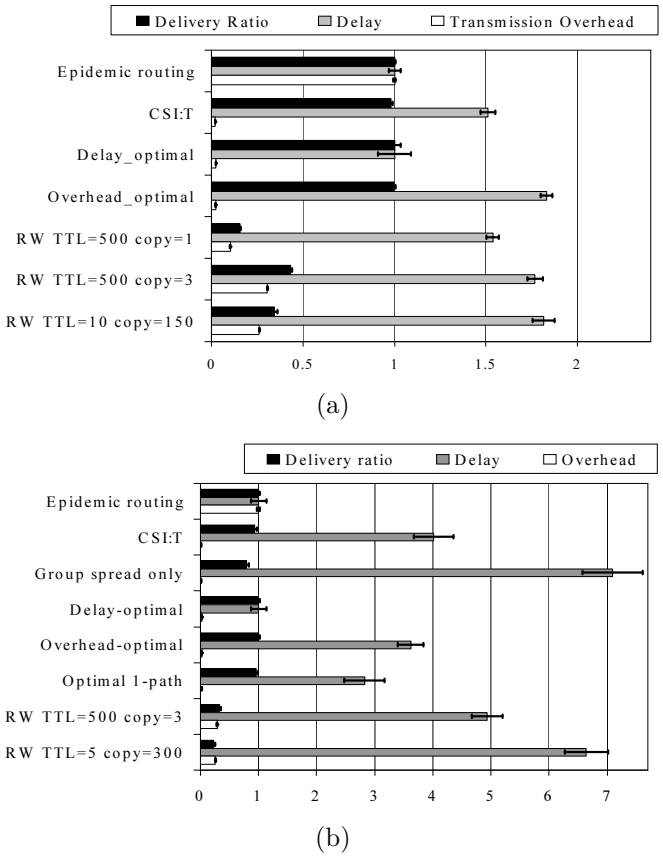
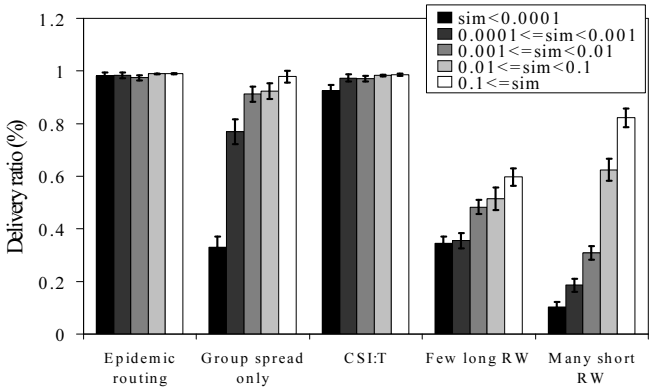


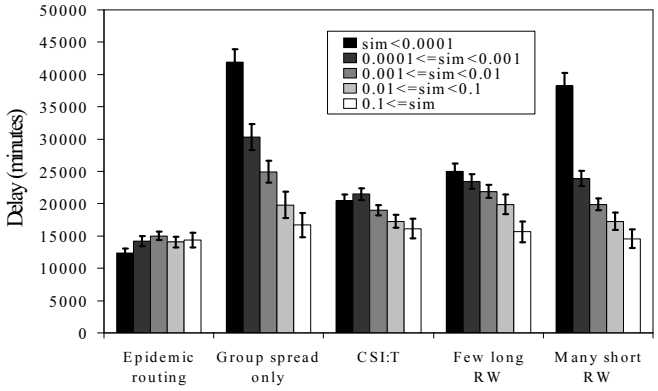
Figure 7-9. Performance comparison of CSI:T to other protocols. (a) USC. (b) Dartmouth.

times more delay, for USC and Dartmouth, respectively. If we compare with the *delay-optimal* strategy, where multiple copies are generated whenever it helps to improve the delay, the difference is even larger. This calls for a further investigation of selecting good path(s) from the sender to the TP, which we leave out for future work.

We take a closer look at the performance metrics by splitting the simulation cases into categories, depending on the original similarity metric between the sender's behavioral profile and the TP, $Sim(BP(S), TP)$. By the split statistics shown in Fig. 7-10, we see why the *gradient ascend phase* is needed to improve the success rate and reduce the delay. When we use only the *group spread phase*, and the sender is dissimilar from the TP, it takes a longer time before any encounter event happens directly between the sender and



(a)



(b)

Figure 7-10. Split performance metrics by the similarity between the sender and the target profile (USC). (a) Delivery ratio. (b) Average delay.

anyone in the neighborhood of the *TP*, if it happens at all – hence the delay is longer, and the success rate is lower.

Comparing the differences between two versions of random walks, few long threads and many short threads, reveals an interesting difference. The concept that leads to the difference is illustrated in Fig. 7-11. Many short threads are better if the sender is close to the *TP*, in terms of both delivery ratio and delay, as the sender generates a lot of threads to "occupy" the neighborhood – since the threads are short, and similar users encounter more frequently, they are likely to stay in the neighborhood. Contrarily, if the sender is far away from the *TP*, long random walk threads provide a legitimate chance of moving close to the *TP*, while short threads provide less hope.

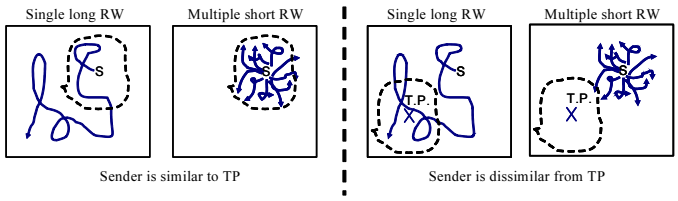


Figure 7-11. Illustrations for the comparison between one long random walk and many short random walks.

7.6.2 CSI: Dissemination Mode

7.6.2.1 Simulation setup

In the scenario of *CSI:D mode*, the target profile specified by the sender cannot help to determine to where the message should be sent in the behavioral space. Hence, the strategy seeks to keep one copy in every neighborhood in the behavioral space. In our evaluation, we start from 1000 randomly selected users as the senders. Since the target profile of the intended receivers can be orthogonal to the behavioral profile, we create the scenario for evaluation by randomly selecting 500 nodes as the intended receivers for each sender, and consider the average performances. We vary the two thresholds, th_{fwd} and th_{nbr} in our *CSI:D mode* scheme proposed in 7.5.4, to adjust the aggressiveness of the forwarding scheme. Setting low values for both thresholds leads to less aggressive operations and inferior performances. At the same time is also leads to lower overheads, as the messages are copied to fewer message holders, and the existence of a message holder prevents nodes in a larger neighborhood from becoming another message holder.

We compare various parameter settings of our *CSI:D* mode with two baseline protocols, the *epidemic routing* and the *random walk*. The epidemic routing works the same way as before, serving as the baseline for comparison. In the random walks, the visited nodes along the walks become message holders and they will later disseminate the messages further when encountering with the intended receivers. The *delay-optimal* protocol again assumes global view of the network and the knowledge of the future. Every node in the network knows who the intended receivers are, and sends the messages to

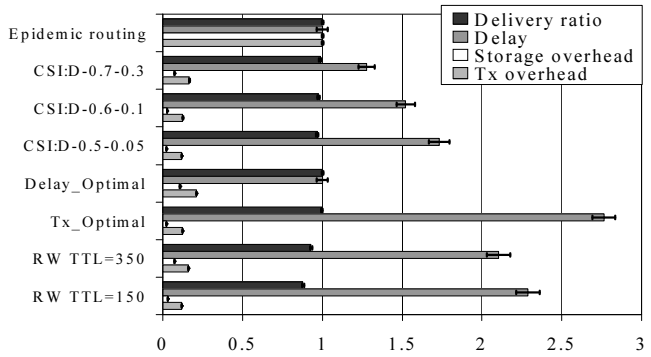
other nodes only if they lead to the fastest delivery of the message to one of the receivers. The *Tx-optimal* (transmission optimal) protocol sends the message to other nodes only if they lead to the delivery of the message to one of the receivers with minimum number of transmissions, considering future encounter events. In both optimal protocols, the intermediate nodes (i.e., non-receivers) keep a copy of the message in the optimal protocols as they have to store this for future transmission(s).

The performance metrics we consider are *delivery ratio*, *average delay*, *transmission overhead*, and, in addition, *storage overhead*. Here the *transmission overhead* refers to the total number of transmissions to reach the message holders and the intended receivers. The *storage overhead* is the number of eventual message holders that remains in the network after our scheme is stabilized (recall that some message holders may decide to cease performing the task if another message holder is found with similar behavioral pattern in *CSI:D*). This is the overall amount of storage space invested by the nodes collectively to deliver the message[6] . In the *epidemic routing* and the *optimal* protocol, all nodes that receive the message hold on to the message for future transmissions (there is no distinction between the message holder and a regular node), hence the transmission overhead and the storage overhead are the same.

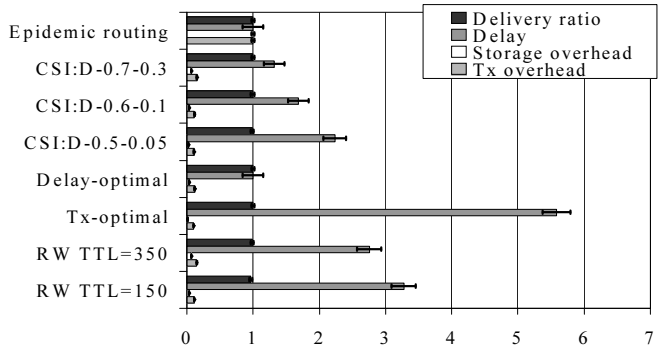
7.6.2.2 Simulation results

In Fig. 7-12 we show the average result of the 1000 simulation cases with the 95% confidence interval. We use the legend CSI:D-th_{fwd}-th_{nbr} for our *CSI:D* scheme. Comparing with the *epidemic routing*, our protocol saves a lot of transmission and storage overhead. It is possible to use only about 7.2% strategically chosen nodes as the message holder and reach the intended receivers with little extra delay (about 32% more), when

[6] Typically, only about a couple dozens of message holders drop the message in the simulation cases. Even if we have accounted for the temporarily invested storage, it adds less than 1% additional storage overhead.



(a)



(b)

Figure 7-12. Performance comparison of CSI:D to other protocols. (a) USC. (b) Dartmouth.

$th_{fwd} = 0.3$ and $th_{nbr} = 0.7$. Notice that the storage overhead of the *CSI:D* scheme is even lower than the *delay-optimal* protocol (less than 60%) with the objective of minimizing the delay. The delay of the *CSI:D* is not much more than the *delay-optimal* protocol, at around 27% to 32% more when $th_{fwd} = 0.3$ and $th_{nbr} = 0.7$. On the other hand, if one desires further reduction in the overhead, setting lower threshold values provide a way to trade performance for overhead, e.g., setting $th_{fwd} = 0.05$ and $th_{nbr} = 0.5$ cuts the transmission overhead to about the same as the *Tx-optimal* protocol (less than 7% more). Performance-wise, the *delivery ratio* is still more than 96.7% with this less aggressive parameter setting, and the *delay* is better than the *Tx-optimal* protocol by 60% and 150% for USC and Dartmouth, respectively.

For the *random walks*, we have configured the TTL values for them to have similar overhead with the *CSI:D* (i.e., compare RW TTL=350 with CSI:D-0.7-0.3 and RW TTL=150 with CSI:D-0.6-0.1). We notice that although the delivery rate of the *random walk* is also pretty good (1.5% to 10% inferior to the corresponding *CSI:D*), thanks to the non-zero encounter probability between dissimilar nodes, its delay is much longer than the corresponding *CSI:D* (between 50% to 108% more). This is because the *random walk* does not leverage the implicit structure of the human network to select the message holders wisely, as the *CSI:D* does. The *random walk* leaves copies within the same neighborhood of the original sender with higher probability, as similar nodes are more likely to encounter (i.e., the *random walk* will not "leave the neighborhood" in a small number of hops). Hence, there exists significant overlap between the nodes encountered by the selected message holders, and the other nodes that are dissimilar to these holders have to wait for a long time before some "random" encounter events occur to receive the message, resulting in the longer delay.

7.7 Discussions

In this section we discuss about some more finer details of the CSI schemes, regarding its overhead and privacy preserving feature.

7.7.1 Additional Overhead

In addition to the message transmission and storage, in our proposed CSI schemes, due to the need for exchanging and maintaining the behavioral profiles, there are some additional overhead. We discuss them in details in this section.

Overhead for exchanging the behavioral profiles: We identify some additional components to the actual message transmissions when the encounter events between mobile nodes are leveraged for message dissemination. Some of the components are common to *any* message dissemination schemes, and the others are unique to our CSI schemes.

- The common overhead for all the DTN message dissemination schemes considered include the beacon signals for nodes to discover each other when they encounter, and the exchange of a list of "messages I have seen" to avoid a given node receiving duplicated messages from different nodes. This type of overhead is a function of the encounter patterns itself and is independent of the actual protocol used. We ignore these common factors in our analysis.
- Exchanging the behavioral profiles for the evaluation of mutual similarity is an additional component that exists only in our behavior-aware CSI schemes. These profiles are a handful of vectors associated with its weights. For most of the users, empirically, five to seven eigen-behavior vectors capture more than 90% of the power in their *association matrices* [73]. This is a small constant overhead we pay for each encounter when one of the nodes has some message to send. If the message size is much larger than the overhead, which is usually the case as messages are transferred in a bigger unit (i.e., a "bundle") in DTNs, it is worthwhile to pay this overhead to gain the reduction of transmission counts as we see in section 7.6. Furthermore, with CSI, if there is no message to send, there is no need to exchange the behavioral profile. Thus, comparing with the protocols that require proactive, persistent exchanges of control messages when nodes encounter (e.g., ProPHET [74] requires the exchange of encounter probability vectors), qualitatively, the CSI schemes have lower overhead, especially when the volume of traffic is low in the network.
- The actual message size has to be augmented with the TP as well. This is a constant overhead, and it can be reduced if the target vector is "sparse" (e.g., if the TP considers only the visits to the gym exclusively, there is only one 1 in the vector. Instead of adding a vector $(0, ..., 0, 1, 0, ...)$ in the header, the vector can be encoded (i.e., by specifying (gym, 1)) to save space.).
- In the CSI:D mode, the message holders have to exchange the list of behavioral profiles of known holders. This happens only between a small subset (less than 8%)

of the nodes, and the exchange is necessary only when there is a difference in the lists. To further alleviate this, the two nodes can compare their known holder lists using a hash value, and exchange only the difference.

Overhead for maintaining the behavioral profiles: In order to maintain the behavioral profile, the nodes have to keep track of its visiting time to various locations. Note this does not require a node be aware of all possible locations in the environment – it has to keep track of only the ones it has been to. When two nodes exchange the behavioral profiles, each entry in the behavioral profile contains only a subset of locations with annotations for these locations (e.g., Node A specifies (library, gym) = (0.8, 0.2) while node B specifies (library, computer lab) = (0.4, 0.6)). The nodes will take a union of the location sets when comparing their similarities (e.g., in the previous example, when node A sends the behavioral profile to B, B will convert the profiles to $BP(A)$: (library, gym, *computer lab*) = (0.8, 0.2, 0) and $BP(B)$: (library, *gym*, computer lab) = (0.4, 0, 0.6) before comparing). The required storage on each node is minimal, as we show about three to five days of summarized *mobility preference* is sufficient to establish a stable behavioral profile for the user in section 7.3.

In addition, if the beacon signals from locations are not available, it is possible to use the mutual encounter vectors as the behavioral descriptors for the nodes – nodes who move similarly should have similar encounter sets. In this sense, we could replace the representation to be totally independent of the infrastructure.

7.7.2 Privacy Issues

While the behavior-aware message dissemination schemes achieve good performance with significant overhead reduction, it also raises user privacy concerns. In some cases, individuals may not want to reveal their own behavior. We discuss privacy-preserving options with our CSI scheme below.

First we emphasize that the original design of CSI presented in section 7.5 inherently possesses a privacy-preserving feature: we only use a small subset of user behavior

(specifically, the mobility preference) in the behavioral profile, and with the singular value decomposition, we reveal only the summarized trend, not detailed location visiting events for the user. In addition, the behavioral profiles are exchanged only between nodes, not stored in any public directory, and it limits only to when a given node is involved in message dissemination.

We can further reduce the behavioral profile exchanges in the CSI scheme, and hence help to preserve privacy as follows. For the CSI:T mode, when nodes encounter, instead of exchanging their behavioral profile, the node with a message to send would first send to the other node the *TP* of the message and its similarity score to the *TP*. The other node silently calculates its similarity to the *TP* and decides whether to request for the actual message. This completely removes the need for behavioral profile exchanges in CSI:T mode.

For the CSI:D mode, when a message holder looks for potential new holders, instead of asking other nodes to send the behavioral profile, the message holder sends the list of known holder's behavioral profiles to the other node. Since this list contains only the *behavioral profiles* of the known holders, not their *identities*, dissemination of such lists in the network does not pose a threat to the privacy of the existing message holders. Furthermore, when there are multiple holders in the list, the other node is not able to tell which behavioral profile corresponds to the holder to whom it is currently corresponding to. If the other node decides to become a message holder, its behavioral profile has to be added to the list of known holders. Instead of immediately sending the behavioral profile of the new holder to the old holder, which poses an opportunity for the old holder to link the identity and the behavioral profile of the new holder, the new holder only adds its behavioral profile to its own known holder list, and delays the dissemination for a later holder profile list exchange.

Finally, as a last resort, privacy-minded individuals can always opt-out of the service, and we expect this would not impact the performance severely, as it has been shown that

the encounter pattern between nodes in mobile networks is rich enough to sustain up to 40% of nodes opting out before observing a performance degradation, in chapter 6.

7.8 Conclusions and Future Work

In this chapter, we propose a paradigm to represent, summarize and manipulate behavioral profiles and use such profiles as targets for the communication. We have presented a novel service of message dissemination in infrastructure-less mobile human networks based on the behavioral profiles of the users. The CSI schemes meet the design goals outlined in section 7.5.1 with respect to efficiency, flexibility and privacy preserving properties. The CSI schemes perform closely to the delay-optimal protocols (with 94% or more success rate, less than 83% of overhead, and the delay is inferior by 40% or less). In addition, we also observe that human behavior as observed in the large-scale empirical traces is quite robust and only a few days' worth of data is adequate to summarize and leverage for message dissemination, which is quite surprising.

We are working toward an implementation of the CSI schemes based on mobile devices and consider a real-world evaluation. One key issue is to adapt our algorithm in a more privacy-preserving fashion which is also resistant to spam (e.g., include a reputation system). We are also considering different applications of behavioral profiles, including targeted advertising via our CSI schemes.

CHAPTER 8
CONCLUSIONS AND FUTURE WORK

In this dissertation, we have performed realistic investigation of user-behaviors based on the *empirical* data sets collected from actual users. The findings at different levels, ranging from micro-scopic individual user mobility characteristics to macro-scopic network-wide encounter patterns, show a significant deviation from the scenarios provided by simplistic synthetic models. Furthermore, we have identified common characteristics in mobile network user behaviors from multiple quantitatively and qualitatively different environments – (a) For *individual user mobility*, we find *infrequent online time*, *skewed location visiting preferences* and *repetitive associations*. (b) For *relationship between user pairs*, we quantify their similarity based on *mobility preferences*, and observe different node pairs have very different degree of similarity, and further cluster users based on the similarity. (c) For the *global encounter pattern*, we see Small Worlds emerge from the graph representing nodal encounter events. The first contribution of my dissertation, hence, is to **instantiate the significant deviations from the synthetic scenarios in realistic mobile network environment**, and such a discrepancy calls for more investigation.

The second contribution of the dissertation is to **leverage the above findings and show-case its impact on various tasks in wireless mobile network design**. I have covered several topics, including (a) Design of a realistic *time-variant community mobility model*, (b) Identification of the *groups of similar users* from the general population, and (c) Proposal of the *profile-cast*, a new service paradigm to deliver messages to groups with a certain *property* without the nodal IDs, based on the understanding of human network structure. The success of the above case studies and its improvement over environment-oblivious design approach highlights **the need of understanding the environments for the design of wireless mobile networks** in the future. As the

mobile devices become ubiquitous and strengthen their couplings with individual users, we believe such environment-aware approach will become a necessity for network design.

Finally, we propose the *TRACE framework* as the over-arching guideline of the dissertation. This is a methodical, step-by-step procedure we follow in each of the case studies of the dissertation. While the details in each step has to be modified according to the specific task at hands, we argue that **the TRACE framework serves as a generic guideline for environment-aware network design**.

Future work:

There are multiple directions derived from this study.

Fundamental understanding of human behavior: As the mobile network devices become ubiquitous and tightly coupled with individuals, monitoring users through collected traces provides a powerful platform to observe and understand fundamental human behaviors. Although the central focus in the dissertation is about its application in wireless network design, the characteristics and patterns identified from the traces really have generic utilizations beyond the scope of the dissertation. We would like to follow up in this direction in the future.

User privacy preservation: As users spend increasing amount of time and perform more tasks online, nowadays online lifestyle is really more exposed to the danger of privacy leak through monitoring. While the trace-based studies provide great promises, it is also of utmost importance to defend user privacy. Issues related to privacy emerge from all steps through such a project, including the collection and post-processing of traces (better anonymization techniques should be devised) and the design of behavior-based message dissemination protocols (users should be able to decide when and how much to reveal their behavior profiles, or opt out altogether). The mobile network paradigm provides various new challenges in user privacy, which is related but out of the scope of this dissertation.

Testbed implementation: The evaluations in this dissertation are based on the assumption that users would behave the same way as reflected by the collected traces

while new services are added. In realistic scenarios, however, the users and the services sometimes *interact* and modify the user behaviors. Hence, it would be of interest to deploy some ideas on small handheld devices and deploy them, say, on university campuses. One prominent example is the profile-cast service. It could be deployed and used for disseminating announcements on university campuses. However, issues such as cost and scale of experiment (ideally, one has to deploy users randomly, in the order of hundreds, to have an unbiased sample) have prevented such a trial. We leave this as a potential future work.

APPENDIX: OBTAINING MOBILITY INFORMATION THROUGH SURVEYS

In this appendix we briefly discuss the effort we made to collect mobility information from students on a university campus. The survey we handed out is shown in section A.1. We build a mobility model, called the Weighted Waypoint model [21], base on the survey results. Descriptions about the WWP model can be found in section A.2. Note that this work is a primitive, bare-bone version of the TVC model we present in chapter 4 – many important ideas can also be found here. Using this model, we show that there can be high user concentration at popular spots on campus due to the high probabilities of visiting the popular locations and the high stay durations at these locations, leading to potential congestions at the APs in these locations. We propose a congestion alleviation protocol by routing some traffic to near-by APs with low load through multi-hop routing across the mobile users.

A.1 Mobility Survey

The survey we handed out includes two parts, for the mobility pattern and the on-campus network usage of the participants, respectively. We show the survey form in Fig. A-1.

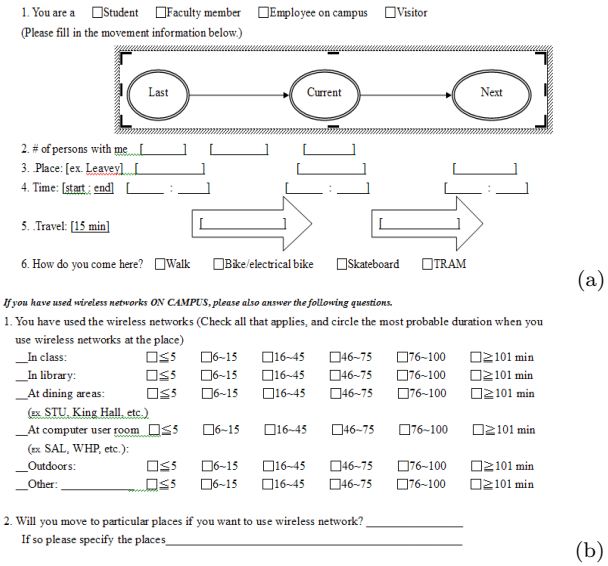


Figure A-1. The survey form. (a). Mobility pattern. (b). Network usage.

A.2 Weighted Waypoint Mobility Model and Its Impact on Ad Hoc Networks

In this section we describe a generic mobility model named weighted waypoint (WWP) model. The WWP model captures the influences of mobile node's preferences in choosing destinations. It also incorporates location-dependent pause duration and weights for choosing next destination. We built one example of the WWP model based on a mobility survey carried out on the campus of University of Southern California (USC). We further show that preferences in destination selection lead to significant difference in network performance.

It is note-worthy that although the survey approach is labor intensive and hence does not scale well, it has some strengths that complement the traces collected from the existing network infrastructure. In particular, the network traces capture the on-line users only, not potential users; hence it observes only part of the total population, and the relationship between on-line and potential users (captured in the survey by questions in Fig. A-1 (b), the probability of using wireless networks) cannot be directly deduced from the network traces.

A.2.1 General Description of the Weighted Waypoint Model

For realistic mobility modeling, it is important to address the following issue: The destination is *not* randomly chosen for pedestrians on a university campus. Given the environment setting of a campus, there are usually popular locations where people tend to visit more often than others. We investigate this issue in this work and propose a new model called the Weighted Way Point (WWP) model. The major differences between the WWP model and the popular Random Waypoint (RWP) model are: (a) *mobile nodes (MN) no longer randomly choose their respective destinations.* We model such a behavior by identifying popular locations in the environment and assigning different "weights" to them according to the probability of choosing those locations as the destinations of nodal movements. We refer to such identified areas as *locations* henceforth. (b) *The "weights" of choosing the next destination location depends on both the current*

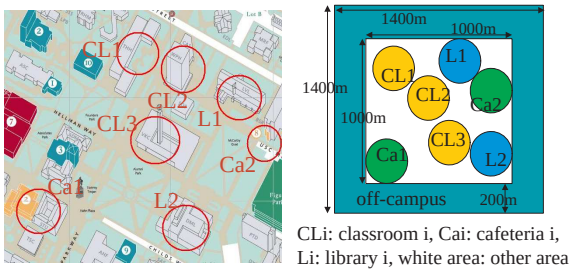


Figure A-2. The virtual campus.

location and time. We use a time-variant Markov model to capture this location and time dependent weight assignment. (c) *The pause time distribution at each location is different* and is a property of that location. In sum, in the WWP model, the simulation area is no longer a homogeneous area without any special point of interest.

A.2.2 Establishing an Example WWP Model based on USC Campus

We apply the above general framework to model a small part of the USC campus, covering several major intersections and buildings. The modeled area is shown in Fig. A-2. We refer this topology as the "virtual campus" henceforth. In this scenario we identify 7 noncontiguous locations: 3 classrooms (CL), 2 libraries (L), and 2 cafeterias (Ca).

In order to find adequate parameters for our WWP model example for the USC campus, we conducted a mobility survey targeted at randomly selected students on campus. During the period between March 22nd 2004 and April 16th 2004, we collected 268 survey responses on the USC campus. The detailed questions we ask in the surveys can be found in section A.1. The location granularity of our mobility survey is per-building. In each survey, the student is asked to fill in his/her current location (building), the previous building visited, the next building to visit, and the pause duration at each of these 3 buildings. To set up the WWP model for a campus environment, we categorize buildings on campus into three different location types: *classrooms*, *libraries*, and *cafeterias*. The buildings and area that does not belong to these 3 categories are collectively referred to as *other area*. We also model the mobile nodes moving to *off-campus area* with certain probabilities. MN chooses its next destination from one

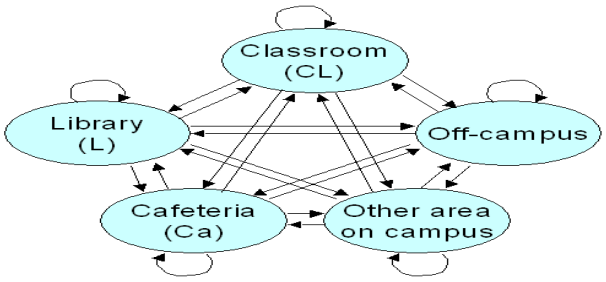


Figure A-3. Markov model of location transition of mobile nodes.

of these 5 *location* types according to a Markov model, as shown in Fig. A-3. We set up the transition probabilities to different location types according to its "weights" or popularity. From the survey we capture statistics about the following parameters: (a) The pause time distributions at classrooms, libraries, cafeterias, and other area. (b) The time-varying transition probability given the current location type and time section (*morning*:9AM-1PM or *afternoon*:1PM-5PM) of the day. (c) In addition to these mobility-related parameters, we also survey for the wireless network usage – the probability and duration a respondent uses wireless networks at different types of locations.

We discuss the main findings of our mobility survey below.

Pause Time Duration The pause time duration is as shown in Fig. A-4. (a) The distribution of pause time at classroom is like a bell-shaped normal distribution with the peak around the 60-120 minutes interval, which is the regular class duration (about 90 minutes) at USC. (b) Also we can see that people are more likely to stay in the library for intervals greater than 240 minutes than in any other locations. For *other area* on campus, the duration tends to be exponentially distributed.

Transition Probability The "transition probability matrix" from the survey data is shown in Table A-1. (a) People tend to go to a cafeteria more in the morning interval (lunchtime) than in the afternoon. Instead of visiting the *other* category, most transitions (more than 50%) are between classrooms and libraries. (b) Also most transitions involving off-campus

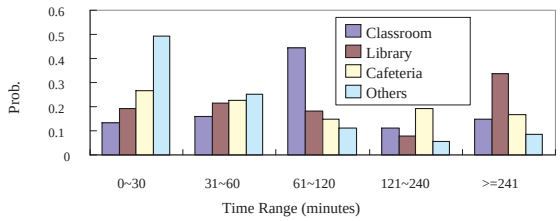


Figure A-4. Pause time distribution for locations.

Current location/time	Destination	Classroom	Library	Cafe	Others	Off Campus
Classroom	9-13	**0.26**	**0.31**	**0.23**	0.14	**0.06**
	13-17	**0.17**	**0.30**	**0.00**	0.19	**0.34**
Library	9-12	**0.14**	**0.14**	**0.26**	0.03	**0.43**
	13-17	**0.36**	**0.23**	**0.04**	0.13	**0.24**
Cafe	9-13	0.15	0.44	0.00	0.22	0.19
	13-17	0.20	0.50	0.00	0.30	0.00
Others	9-13	0.09	0.12	0.25	0.30	0.24
	13-17	0.20	0.43	0.09	0.14	0.14
Off Campus	9-13	**0.69**	**0.21**	0.05	0.05	0.00
	13-17	**0.64**	**0.24**	0.02	0.04	0.06

Table A-1. Transition probability matrix.

location are of the type "offcampus-class-offcampus" or "offcampus-library-offcampus" which we believe reflects the general student behavior. This implies the fact that off-campus students come to campus mostly to attend classes or to use libraries.

We also try to obtain the transition probability matrix from the USC wireless network traces [80], with building-level granularity. There are three initial findings on this: (a) Starting from a given building, the transition probabilities toward the others are not equally distributed. This supports our assumption that some locations are more popular than others in a campus environment. (b) From the trace we observe similar trends to the survey – Cafeterias are more popular in the morning interval, and there are a lot transitions between libraries and classrooms. (c) From a given building, the transition probabilities toward close-by buildings are higher than buildings that are far away. This may suggest that pedestrian mobility on campus exhibits *locality*.

Wireless Network flow duration The histogram of flow duration distributions at different types of locations is shown in Fig. A-5. The flow duration distribution shows a heavier tail for the library, probably due to people working in the libraries with their laptop connected to the wireless network. We further compare the findings of this part with

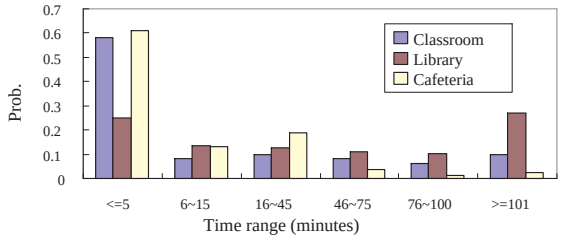


Figure A-5. Flow duration distribution for locations.

the distributions of user online time in the Dartmouth WLAN traces [13]. From the Dartmouth trace [81] we find that for most buildings the online time distribution is highly skewed toward short durations, regardless of the building type. The observation based on our surveys and traces are similar except for the libraries.

A.2.3 Simulation Results

A.2.3.1 Properties of WWP model

We use simulations to show the characteristics of the WWP model, in comparison with the RWP model. First, WWP model shows *uneven spatial distribution* of MNs. The MNs tend to cluster within the popular locations, as shown in Fig. A-6. However the node density is quite low for other area and off campus locations. This is a combined effect of popular locations being chosen as destinations with higher probabilities and pause times at those locations being long with higher probability. Second, although for a given fixed transition probability matrix there should be some theoretical steady state of the MN distribution, the transition probability matrix is time-dependent and changes from time to time throughout the day, hence the MN distribution in the simulation area never reaches a steady state in Fig. A-6. This suggests *converging to a steady-state distribution is not necessarily a requirement of realistic mobility models*. Third, we use the *move-stop ratio* (the total move time divided by the total stationary time) as one metric of a mobility model and find that the WWP model based on our mobility survey data has a lower move-stop ratio 0.12 as compare to 0.99 from the RWP model with common parameter

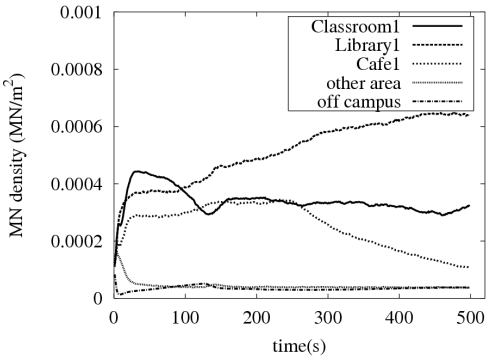


Figure A-6. Mobile node density versus time.

Model and parameters	Move-stop ratio
WWP with both transition matrix	0.12
RWP with pause time=[0,100](s), speed=[2,50](m/s) — typical parameter setting	0.99

Table A-2. Move-stop ratio.

settings, as shown in Table A-2. This indicates, *for a university campus scenario, people are less mobile than typical scenarios generated by the RWP model.*

A.2.3.2 Impact of the WWP model on network performance

We further show the impact of the WWP model on the network performance. We consider both last-hop wireless networks (802.11 WLANs) and ad hoc networks. Assuming MN only uses wireless networks with some probability (the probability of using the WLAN at a given location type is obtained from the survey data) when it stops within classrooms, libraries, and cafeterias, we find that as the number of MNs increases in the system, the WWP model has about twice the number of flows as compared to the RWP model. Also the congestion ratio (the ratio of flows connected to an AP with 7 or more simultaneous connected flows) of the WWP model is doubled comparing with the congestion ratio of the RWP model. Another interesting result reveals that even when both models have the similar number of flows, the WWP model always has a higher congestion ratio than the RWP model. This is because in the WWP model locations are chosen as MN destinations with non-uniform weights. If a location is more popular than others, it attracts more MNs

hence a greater proportion of the flows are initiated at the location. Thus some locations have seen more flows and these flows are likely to be congested. Where as in the RWP model the flows are more evenly distributed among the locations hence the congestion ratio is not as high given the same number of total flows.

For ad hoc networks, we compare the success rate of route discovery using DSR [103] as the routing protocol under two different MN location relationships, MN pairs in the same location and MN pairs in different locations. If the WWP model is used, we show that the route discovery success rates are 88.61% and 28.53% for MNs in the same location and in different locations, respectively. The reason for the low route discovery success rate for MNs in different locations is that the number of nodes present between these locations is very small due to the preference of choosing popular locations as destinations. Hence few nodes are able to serve as the intermediate nodes to establish a route between MNs in different locations. Therefore it is likely that the network will be partitioned into small subsets clustered at the popular locations, and it is difficult to find a route between these subsets.

A.3 A Congestion Alleviation Mechanism for WLANs

As the MNs cluster at the popular locations, more flows are generated toward local APs. However, since the distribution of MNs is uneven across locations, the distribution of flows is also uneven across APs. We show the number of simultaneous flows at three APs located in the upper-right corner of the virtual campus (Fig. A-2) as a function of time in Fig. A-7 when there are 200 MNs in the simulation. While the AP at library1 has a large number of flows, APs at classroom2 and cafeteria2 are quite underutilized. This uneven distribution of flows suggests the possibility of using ad hoc network techniques to re-route some flows to the underutilized neighboring APs in order to alleviate local congestion. It is feasible to improve the QoS of the flows at the congested AP, if we can find a multi-hop ad hoc route to redirect it to underutilized neighboring APs (NAPs). We propose the following MN-initiated flow-switching mechanism to achieve this goal.

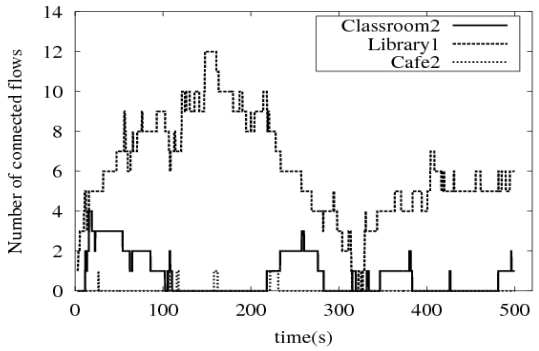


Figure A-7. Uneven flow distribution across APs.

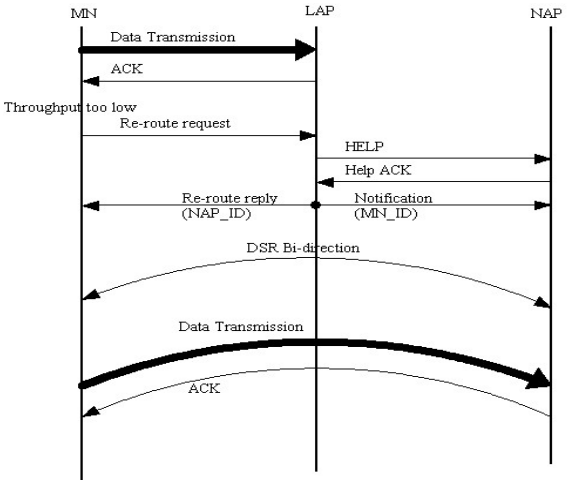


Figure A-8. The control flow chart of the proposed flow-switching mechanism.

A.3.1 Flow-Switching Mechanism

The nodes with on-going flows keep monitoring the average end-to-end throughput to the local access point (LAP). If the average throughput is lower than an application-defined threshold, the MN notifies the LAP that it would like to be re-routed to a NAP using an ad hoc multi-hop route, in the hope of getting better average throughput. The following operations of our proposed mechanism is also summarized in Fig A-8.

The MN notifies the LAP of its request to be switched to other APs by sending a "re-route request" to the LAP. Upon receiving this message, the LAP requests help from its neighbors by sending a "help" message to one of them. The choice of the neighbor is based on the APs geographical knowledge of the AP-deployment topology. The LAP

will make a random choice from its close by neighbors. It is possible to make a better choice by looking at the current loads of NAPs, but we do not incorporate this option currently. The NAP replies with a "help ACK" message. The local AP then notifies the two parties (MN and NAP) about the ID of each other. Based on this information, the neighbor AP and the MN can send out a route request packet (We adopt DSR [103] as the ad hoc routing protocol.) for each other simultaneously. This is achievable because the wired network between the access points provides a "tunnel" to exchange information between the MN and the NAP before they actually establish an ad hoc route to each other. The intermediate nodes at which the bi-directional route request messages meet will concatenate the partial routes from both ends and send back route reply messages to the MN and the NAP. Such "meet in the halfway" behavior is possible because DSR caches the partial route a route-request packet traversed before reaching the node, therefore an intermediate node is able to establish the end-to-end path if it is visited by route-request packets from both ends one after the other. The bi-directional search for the ad hoc route can potentially reduce the route discovery time.

In our work we assume that MNs use a dedicated wireless channel to communicate with other MNs, so that the ad hoc network does not interfere with congested local wireless channel used by the LAP and other MNs. This can be achieved by reserving a dedicated channel for the ad hoc network communication. All APs and MNs in the system must agree on using this reserved channel only for the ad hoc network communication. The channel is not used locally by any AP.

If the LAP assigns a MN to be switched to one of its neighbor, but there is no available multi-hop route from the MN to the NAP, the switching is considered a failure and the MN will reestablish its connection to the LAP after a fixed period of time. If the MN is able to establish a route to the designated NAP, but the route breaks later due to movement of intermediate nodes, the MN will also reestablish the connection to the LAP. Such fall-back-to-LAP behavior is necessary to avoid a MN waiting indefinitely for

an ad hoc route to the designated NAP, which may not appear for a long time. If the LAP is still congested, the MN may start another switch trial later, possibly to another NAP. Note that for the duration of the flow to the LAP, the MN stays stationary and within the coverage of the LAP, so the route to the LAP is always available. The MN switches the flow to a NAP only for better throughput, not because the route to the LAP is unavailable.

In order to avoid the situation that all MNs sense the congestion at the LAP at the same time and try to switch, potentially leaving the LAP underutilized and the NAPs congested, we add a randomization factor in making switching decisions. When a MN sense local congestion, it does not always try to switch immediately. Instead, it sends the re-route request with a switching-initiation probability p. By adjusting the switching-initiation probability, we can reduce the effect of shifting overload APs at the cost of slower responses to local congestion.

A.3.2 Simulation Results

We use ns-2 [88] network simulator to simulate our proposed flow switching mechanism. We vary the total number of MNs in the simulation area from 100 to 200 to create different degree of congestion. The mobility model used by the MNs is the proposed WWP model introduced in section A.2. In the simulation, we assume that each AP operates at 2Mb/s bit rate. Each MN flow requires 200Kb/s throughput. To simplify the simulation, the MNs identify the LAP congestion by counting current number of flows connected to the LAP. The local AP becomes congested and the throughput for local flows start to drop if 7 or more simultaneous flows are connected to the LAP (This number was obtained via detailed simulations. The wireless channel cannot reach 100% utilization because of contentions in the wireless channel.) We simulate both the scenarios with and without the flow switching mechanism.

The effect of the flow-switching algorithm is primarily to re-distribute the load of traffic across the APs. If some AP becomes congested, the MNs sense the congestion by

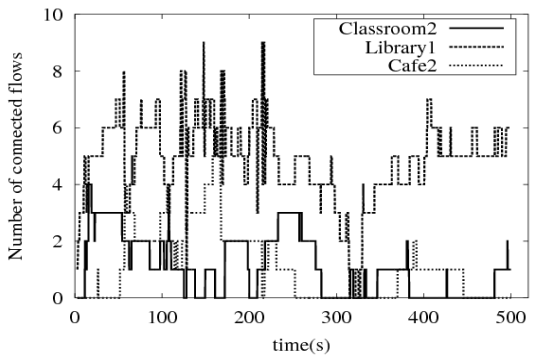


Figure A-9. Flows re-distributed across APs, relieving congestion at library1.

observing degradation in the throughput of the on-going flow and try to switch the flow to one of the NAPs. If some of the MNs succeed in flow switching, the excessive flows at the LAP will shift to its neighbors, and both the flows that are switched and the flows that stay at the LAP can enjoy uncongested wireless channel and better throughput. The consequence of the flow-switching is illustrated by comparing Fig. A-7 to Fig. A-9, where we illustrate the number of flows at the same 3 APs located in the upper-right corner of the virtual campus (Fig. A-2), with the flow-switching mechanism. We see that some flows at library1 are switched to classroom2 and cafeteria2, so the congestion at library1 is not as bad as in the case without flow-switching shown in Fig. A-7.

To better understand the effect of the flow-switching mechanism on the overall improvements of the system, we propose to use the metrics "AP congested time ratio" and "flow quality time ratio". The former is defined as the time ratio an AP has at least 7 flows connected to it. This is the time ratio that the AP cannot provide adequate QoS to the connected flows. The latter is defined as the time ratio of a flow connected to any AP with less than 7 flows connected simultaneously. This is the proportion of time the flow receives adequate throughput. Note that between the time a MN decides to switch a flow to NAP until the time it finds a route to the designated NAP, the flow is not connected to any AP hence this time period will not be included the quality time

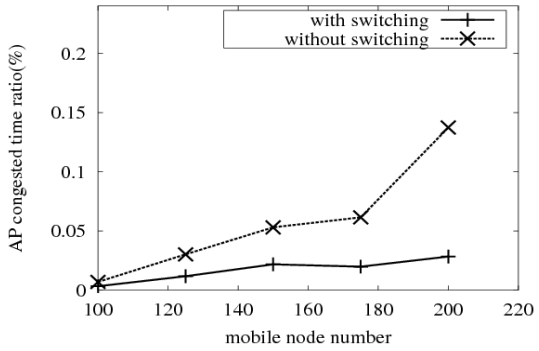


Figure A-10. All APs: Average AP congested time ratio.

ratio. The results shown below are the averages of 6 independent simulation runs, using a different, randomly generated mobility scenario for each run.

Fig. A-10 shows the average of AP congested time ratio of all APs. Fig. A-11 shows the average of AP congested time ratio of the most congested AP in each simulation run. We can see that due to the uneven MN distribution resulting from the WWP model, the overall congested time ratio is low for the whole system. However, the most congested AP is quite overloaded. This is exactly the situation when switching some flows to the NAPs should be helpful. From the figures we see that the congested time ratio of the most congested AP is reduced by more than 50% in all except for the 100 MN case. This implies flow-switching helps to reduce the local congestion of wireless LANs more than half of the time when congestion exists. The flow quality time ratio is the metric to observe for the improvement we get by employing flow-switching from user's perspective. In Fig. A-12 we show the flow-switching mechanism improves the quality time ratio for all cases.

We observe in the case of smaller MN numbers (100 or 125 MNs) the effect of flow-switching is not so pronounced. This is because when the network is sparse, there is less chance to find a route to the designated NAP for the switching flows. Hence the effectiveness of flow-switching is limited. The success rate for a switching flow to find a route to the chosen NAP is about 0.27 when there are 100 MNs, and the success rate increases to 0.43 when there are 200 MNs.

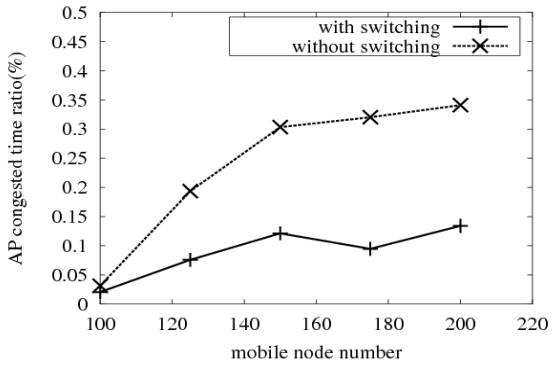


Figure A-11. The most congested AP: Average AP congested time ratio.

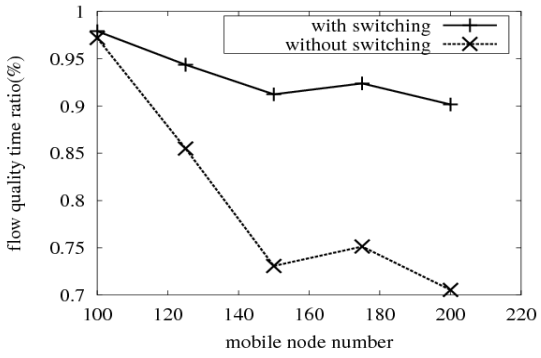


Figure A-12. Average quality time ratio of all flows.

REFERENCES

[1] MobiLib: Community-wide Library of Mobility and Wireless Networks Measurements. http://nile.usc.edu/MobiLib.

[2] CRAWDAD: A Community Resource for Archiving Wireless Data At Dartmouth. http://crawdad.cs.dartmouth.edu/index.php.

[3] C. Perkins, "Ad Hoc Networking," Addison-Wesley, published Dec. 2000.

[4] Delay tolerant networking research group. http://www.dtnrg.org.

[5] R. Aldunate, S. Ochoa, F. Pena-Mora, and M. Nuaabaum, "Robust Mobile Ad Hoc Space for Collaboration to Support Disaster Relief Efforts Involving Critical Physical Infrastructure," In Journal of Comp. in Civil Engineering, vol. 20, issue 1, pp. 13-27, 2006.

[6] S. Jain, K. Fall, and R. Patra, "Routing in a delay tolerant network," In Proceedings of ACM SIGCOMM, Aug. 2004.

[7] F. Bai, T. Elbatt, G. Hollan, H. Krishnan, V. Sadekar, "Towards Characterizing and Classifying Communication-based Automotive Applications from a Wireless Networking Perspective," in Proceedings of the 1st IEEE Workshop on Automotive Networking and Applications (AutoNet 2006), Nov. 2006.

[8] D. J. Watts and S. H. Strogatz. "Collective Dynamics of 'Small-World' Networks," Nature, vol. 393, pp. 440-442, 1998.

[9] Simulation codes used in this work and its detailed description are available at http://nile.cise.ufl.edu/~weijenhs/TVC_model

[10] M. Balazinska and P. Castro, "Characterizing Mobility and Network Usage in a Corporate Wireless Local-Area Network," In Proceedings of MobiSys 2003, pp. 303-316, May 2003.

[11] M. McNett and G. Voelker, "Access and mobility of wireless PDA users," ACM SIGMOBILE Mobile Computing and Communications Review, v.7 n.4, October 2003.

[12] D. Kotz and K. Essien, "Analysis of a Campus-wide Wireless Network," In Proceedings of ACM MobiCom, September, 2002.

[13] T. Henderson, D. Kotz and I. Abyzov, "The Changing Usage of a Mature Campus-wide Wireless Network," in Proceedings of ACM MobiCom 2004, September 2004.

[14] M. Papadopouli, H. Shen, and M. Spanakis, "Characterizing the Duration and Association Patterns of Wireless Access in a Campus," 11th European Wireless Conference 2005, Nicosia, Cyprus, April 10-13, 2005.

[15] W. Hsu and A. Helmy, "On Important Aspects of Modeling User Associations in Wireless LAN Traces," the Second International Workshop On Wireless Network Measurement (WiNMee 2006), April 2006.

[16] N. Eagle and A. Pentland, "Reality mining: sensing complex social systems," in Journal of Personal and Ubiquitous Computing, vol.10, no. 4, May 2006.

[17] Cab Spotting, a project that tracks taxi mobility in the San Francisco Bay Area. Trace available at http://cabspotting.org/api.

[18] W. Leland, M. Taqqu, W. Willinger, and D. Wilson, "On the Self-similar Nature of Ethernet Traffic (extended version)," in IEEE/ACM Transactions on Networking, Vol. 2, Issue 1, pp. 1 – 15, Feb. 1994.

[19] C. Faloutsos, P. Faloutsos, and M. Faloutsos, "On Power-Law Relationships of the Internet Topology," In Proceedings of the ACM SIGCOMM Conference on Network Architectures and Protocols, Sep. 1999.

[20] F. Bai, N. Sadagopan, and A. Helmy, "The IMPORTANT Framework for Analyzing the Impact of Mobility on Performance of Routing for Ad Hoc Networks", AdHoc Networks Journal - Elsevier, Vol. 1, Issue 4, pp. 383 – 403, Nov. 2003.

[21] W. Hsu, K. Merchant, H. Shu, C. Hsu, and A. Helmy, "Weighted Waypoint Mobility Model and its Impact on Ad Hoc Networks - Mobicom 2004 Poster Abstract," Mobile Computing and Communication Review, Jan 2005.

[22] D. Batacharjee, A. Rao, C. Shah, M. Shah, and A.Helmy "Empirical Modeling of Campus-wide Pedestrian Mobility: Observation on the USC Campus", in Proceedings of IEEE Vehicular Technology Conference (VTC), Sep. 2004.

[23] D. Tang and M. Baker, "Analysis of a Local-area Wireless Network," In Proceedings of the 6th annual international conference on Mobile computing and networking (MobiCom 2000), Aug. 2000.

[24] D. Tang and M. Baker, "Analysis of a Metropolitan-Area Wireless Network," Wireless Networks, vol. 8, no. 2-3, pp. 107 – 120, Nov. 2004

[25] H. Zang and J. Bolot, "Mining Call Data to Increase the Robustness of Cellular Networks to Signaling DoS Attacks," in Proceedings of ACM MobiCom 2007, September 2007.

[26] The Haggle Project. http://www.cl.cam.ac.uk/research/srg/netos/haggle/index.html

[27] A. Chaintreau,P. Hui, J. Crowcroft, C. Diot, R. Gass, and J. Scott, "Impact of Human Mobility on the Design of Opportunistic Forwarding Algorithms," in Proceedings of INFOCOM 2006, Apr. 2006.

[28] J. Leguay, A. Lindgren, J. Scott, T. Friedman, and J. Crowcroft, "Opportunistic Content Distribution in an Urban Setting," in Proceedings of Workshop on Challenged Networks (CHANTS-06), co-located with ACM SIGCOMM 2006, Sep. 2006.

[29] J. Su, A. Chin, A. Popivanova, A. Goel, and E. de Lara, User mobility for opportunistic ad-hoc networking, in Proceedings of the 6th IEEE Workshop on Mobile Computing Systems and Applications (WMCSA04), 2004.

[30] J. Burgess, B. Gallagher, D. Jensen, and B. Levine, "MaxProp: Routing for Vehicle-Based Disruption-Tolerant Networks," In Proceedings of IEEE INFOCOM, Apr. 2006.

[31] N. Banerjee, M. Corner, and B. Levine, "An Energy-Efficient Architecture for DTN Throwboxes," in Proceedings of INFOCOM 2007, May 2007.

[32] P. Juang, H. Oki, Y. Wang, M. Martonosi, L-S. Peh, and D. Rubenstein, "Energy-Efficient Computing for Wildlife Tracking: Design Tradeoffs and Early Experience with ZebraNet," In ASPLOS-X conference, Oct. 2002.

[33] T. Small and Z. Haas, "The Shared Wireless Infostation Model: a New Ad Hoc Networking Paradigm (or where there is a whale, there is a way)," In Proceedings of the 4th ACM MOBIHOC, June 2003.

[34] M. Kim and D. Kotz, "Modeling Users' Mobility among WiFi Access Points," in Proceedings of the International Workshop on Wireless Traffic Measurements and Modeling (WiTMeMo '05), Seattle, Washington, June 2005.

[35] M. Papadopouli, H. Shen, M. Spanakis, "Modeling Client Arrivals at Access Points in Wireless Campus-wide Networks," 14th IEEE Workshop on Local and Metropolitan Area Networks, Chania, Crete, Greece, September 18-21, 2005.

[36] G. Chen, H. Huang, and M. Kim, "Mining Frequent and Periodic Association Patterns," Dartmouth College Computer Science Technical Report TR2005-550, July 2005.

[37] X. Meng, S. Wong, Y. Yuan, and S. Lu, "Characterizing Flows in Large Wireless Data Networks," in Proceedings of ACM MobiCom, September 2004.

[38] M. Kim and D. Kotz, "Periodic properties of user mobility and access-point popularity," Journal of Personal and Ubiquitous Computing, 11(6), August, 2007.

[39] J. Kang, W. Welbourne, B. Stewart, and G. Borriello, "Extracting places from traces of locations," in SIGMOBILE Mobile Computing and Communication Review, vol. 9, no. 3, July 2005.

[40] I.T. Jolliffe, Principal Component Analysis, second ed., Springer series in statistics, published 2002.

[41] R. Horn and C. Johnson, Matrix Analysis, Cambridge University Press, published 1990.

[42] A. Lakhina, K. Papagiannaki, M. Crovella, C. Diot, E. D. Kolaczyk, and N. Taft, "Structural Analysis of Network Traffic Flows," ACM SIGMETRICS, New York, June 2004.

[43] C. J. F. ter Braak, "Principal Components Biplots and Alpha and Beta Diversity," Ecology, vol. 64, pp. 454-462, 1983.

[44] R. Albert and A. Barabasi, "Statistical mechanics of complex networks," Review of modern physics, vol. 74, no. 1, pp. 47-97 , Jan. 2002.

[45] A. Helmy, "Small Worlds in Wireless Networks," IEEE Communications Letters, pp. 490-492, Vol. 7, No. 10, October 2003.

[46] F. Bai and A. Helmy, "Impact of Mobility on Last Encounter Routing Protocols," in Proceedings of the Fourth Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks (SECON 2007), June 2007.

[47] V. Srinivasan, M.Motani, and W. T. Ooi, "Analysis and Implications of Student Contact Patterns Derived from Campus Schedules," in Proceedings of MOBICOM 2006, Sep. 2006.

[48] M. Grossglauser and D. Tse, "Mobility increases the capacity of ad hoc wireless networks," IEEE/ACM Transactions on Networking, 10(4), 2002.

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

[50] F. Bai, A. Helmy, "A Survey of Mobility Modeling and Analysis in Wireless Adhoc Networks", Book Chapter in "Wireless Ad Hoc and Sensor Networks", Springer, October 2006, ISBN: 978-0-387-25483-8.

[51] C. Bettstetter, G. Resta, P. Santi, "The Node Distribution of the random waypoint mobility model for wireless ad hoc networks," in IEEE Trans. on Mobile Computing, vol. 2, issue 3, pp. 257-269, Jul. 2003.

[52] T. Spyropoulos and K. Psounis, "Performance Analysis of Mobility-assisted Routing," In Proceedings of ACM MOBIHOC, May 2006.

[53] A. Jindal and K. Psounis, "Fundamental Mobility Properties for Realistic Performance Analysis of Intermittently Connected Mobile Networks," In Proceedings of IEEE PerCom Workshop on Intermittently Connected Mobile Ad Hoc Networks (ICMAN), 2007.

[54] S. Jain, K. Fall, and R. Patra, "Routing in a delay tolerant network," In Proceedings of ACM SIGCOMM, Aug. 2004.

[55] R. C. Shah, S. Roy, S. Jain, and W. Brunette, "Data mules: Modeling and analysis of a three-tier architecture for sparse sensor networks," Elsevier Ad Hoc Networks Journal, 2003.

[56] T. Spyropoulos, K. Psounis, and C. S. Raghavendra, "Single-copy routing in intermittently connected mobile networks," In Proceedings of IEEE SECON, Oct. 2004.

[57] T. Spyropoulos, K. Psounis, and C. S. Raghavendra, "Spray and wait: Efficient routing in intermittently connected mobile networks," In Proceedings of ACM SIGCOMM workshop on Delay Tolerant Networking (WDTN), Aug. 2005.

[58] X. Hong, M. Gerla, G. Pei, C. and Chiang, "A group mobility model for ad hoc wireless networks," in Proceedings of the 2nd ACM international workshop on modeling, analysis and simulation of wireless and mobile systems, August, 1999.

[59] A. Jardosh, E. Belfding-Royer, K. Almeroth, S. Suri. "Towards Realistic Mobility Models for Mobile Ad hoc Networks." in Proceedings of ACM MobiCom, Sep. 2003.

[60] W. Hsu and A. Helmy, "On Nodal Encounter Patterns in Wireless LAN Traces," the Second International Workshop On Wireless Network Measurement (WiNMee 2006), April 2006.

[61] J. Leguay, T. Friedman, and V. Conan, "Evaluating Mobility Pattern Space Routing for DTNs," in Proceedings of IEEE INFOCOM, April, 2006.

[62] C. Tuduce and T. Gross, "A Mobility Model Based on WLAN Traces and its Validation," In Proceedings of IEEE INFOCOM, Mar. 2005.

[63] R. Jain, D. Lelescu, M. Balakrishnan, "Model T: An Empirical Model for User Registration Patterns in a Campus Wireless LAN," In Proceedings of ACM MOBICOM, Aug. 2005.

[64] D. Lelescu, U. C. Kozat, R. Jain, and M. Balakrishnan, "Model T++: An Empirical Joint Space-Time Registration Model," In Proceedings of ACM MOBIHOC, May 2006.

[65] M. Kim, D. Kotz, and S. Kim, "Extracting a mobility model from real user traces," In Proceedings of IEEE INFOCOM, Apr. 2006.

[66] W. Hsu and A. Helmy, "On Important Aspects of Modeling User Associations in Wireless LAN Traces," In Proceedings of the Second International Workshop On Wireless Network Measurement, Apr. 2006.

[67] W. Hsu, T. Spyropoulos, K. Psounis, A. Helmy, "Modeling Time-variant User Mobility in Wireless Mobile Networks," In Proceedings of IEEE INFOCOM, May 2007.

[68] M. Musolesi and C. Mascolo, "A Community Based Mobility Model for Ad Hoc Network Research," In Proceedings of the Second International Workshop on Multi-hop Ad Hoc Networks (REALMAN), May 2006.

[69] S. Merugu, M. Ammar, and E. Zegura, "Routing in Space and Time in Networks with Predictable Mobility," Georgia Institute of Technology CC Technical Report, GIT-CC-04-07, 2004.

[70] W. Zhao, M. Ammar, and E. Zegura, "Multicasting in Delay Tolerant Networks: Semantic Models and Routing Algorithms," In Proceeding of the 2005 ACM SIGCOMM workshop on Delay-tolerant networking, Aug. 2005.

[71] A. Vahdat and D. Becker, "Epidemic Routing for Partially Connected Ad Hoc Networks," Technical Report CS-200006, Duke University, April 2000.

[72] Y. Ko and N. Vaidya, "Flooding-Based Geocasting Protocols for Mobile Ad Hoc Networks," Mobile Networks and Applications, vol. 7, Issue 6, pp. 471-480, Dec. 2002.

[73] W. Hsu, D. Dutta, and A. Helmy, "Extended Abstract: Mining Behavioral Groups in Large Wireless LANs" In Proceedings of ACM MOBICOM, Sep. 2007. Longer technical report available at http://arxiv.org/abs/cs/0606002

[74] A. Lindgren, A. Doria, and O. Scheln, "Probabilistic routing in intermittently connected networks," Lecture Notes in Computer Science, vol. 3126, pp. 239 -V 254, Sep. 2004.

[75] A. Helmy, S. Garg, P. Pamu, N. Nahata, "Contact Based Architecture for Resource Discovery (CARD) in Large Scale MANets", Third IEEE/ACM International Workshop on Wireless, Mobile and Ad Hoc Networks (WMAN), Apr. 2003.

[76] E. Daly and M. Haahr, "Social Network Analysis for Routing in Disconnected Delay-Tolerant MANETs," In Proceedings of ACM MOBIHOC, Sep. 2007.

[77] W. Hsu, K. Merchant, H. Shu, C. Hsu, and A. Helmy, "Preference-based Mobility Modeling and the Case for Congestion Relief in WLANs using Ad hoc Networks," IEEE Vehicular Technology Conference (VTC), Los Angeles CA, Sep. 2004.

[78] P. Hui, E. Yoneki, S. Chan, and J. Crowcroft, "Distributed Community Detection in Delay Tolerant Networks," In Proceedings of Sigcomm Workshop MobiArch '07, August, Kyoto, Japan.

[79] A.B. McDonald and T.F. Znati, "A Mobility-based Framework for Adaptive Clustering in Wireless Ad Hoc Networks," In IEEE Journal on Selected Areas in Communications, vol. 17, issue 8, pp. 1466-1487, Aug. 1999.

[80] W. Hsu and A. Helmy, MobiLib USC WLAN trace data set. Downloaded from http://nile.cise.ufl.edu/MobiLib/USC_trace/

[81] D. Kotz, T. Henderson and I. Abyzov, CRAWDAD data set dartmouth/campus/ movement/01_04 (v. 2005-03-08). Downloaded from http://crawdad.cs.dartmouth.edu/dartmouth/ campus/movement/01_04

[82] M. Balazinska and P. Castro, CRAWDAD data set ibm/watson (v. 2003-02-19). Download from http://crawdad.cs.dartmouth.edu/ibm/watson

[83] M. McNett and G. M. Voelker, Wireless Topology Discovery project data set. Download from http://sysnet.ucsd.edu/wtd/

[84] J. Scott, R. Gass, J. Crowcroft, P. Hui, C. Diot, and A. Chaintreau, CRAWDAD trace cambridge/haggle/imote/infocom (v. 2006-01-31). Download from http://crawdad.cs.dartmouth.edu/cambridge/haggle/imote/infocom

[85] D. Kotz, T. Henderson, and I. Abyzov, CRAWDAD trace set dartmouth/campus/tcpdump (v. 2004-11-09). We use the list of device types in the fall03 tcpdump data. Download from http://crawdad.cs.dartmouth.edu/dartmouth/campus/tcpdump

[86] P. Krishna, N.H. Vaidya, M. Chatterjee, and D.K. Pradhan, "A Cluster-based Approach for Routing in Dynamic Networks," In ACM SIGCOMM Computer Communication Review, vol. 27, issue 2, pp. 49-64, Apr. 1997.

[87] K. Seada, M. Zuniga, A. Helmy, B. Krishnamachari, "Energy-Efficient Forwarding Strategies for Geographic Routing in Lossy Wireless Sensor Networks," In Proceedings of ACM Sensys, Nov, 2004.

[88] The Network Simulator - NS-2. http://www.isi.edu/nsnam/ns/

[89] B. Karp and H. Kung, "GPSR: greedy perimeter stateless routing for wireless networks," In Proceedings of ACM MobiCom, Aug. 2000.

[90] S.Tanachaiwiwat and A. Helmy, "Encounter-based Worms: Analysis and Defense", IEEE Conference on Sensor and Ad Hoc Communications and Networks (SECON) 2006 Poster/Demo Session, September 2006.

[91] X. Zhang, G. Neglia, J. Kurose, and D. Towsley, "Performance Modeling of Epidemic Routing," in Proceedings of IFIP Networking 2006.

[92] R. Groenevelt, P. Nain, and G. Koole, "The Message Delay in Mobile Ad Hoc Networks," In Proceedings of PERFORMANCE, Oct. 2005.

[93] M. Grossglauser and M. Vetterli, "Locating Nodes with EASE: Mobility Diffusion of Last Encounters in Ad Hoc Networks," In Proceedings of IEEE INFOCOM, April 2003.

[94] H. Dubois-Ferriere, M. Grossglauser, and M. Vetterli. "Age Matters: Efficient Route Discovery in Mobile Ad Hoc Networks using Encounter Ages, In Proceedings of ACM MobiHoc, June 2003.

[95] A. Jain, M. Murty, and P. Flynn, "Data Clustering: A Review," ACM Computing Surveys, vol. 31, no. 3, September, 1999.

[96] L. Denoeud, H. Garreta, A. Guenoche, "Comparison of distance indices between partitions," International Symposium on Applied Stochastic Models and Data Analysis, May 2005.

[97] J. Faruque and A. Helmy, "RUGGED: RoUting on finGerprint Gradients in sEnsor Networks," In Proceedings of the The IEEE/ACS International Conference on Pervasive Services (ICPS'04), Jul. 2004.

[98] R. Hogg and E. Tanis, "Probability and Statistical Inference," sixth edition, Prentice Hall, 2001.

[99] A. Rapoport and W. Horvath, "A Study of a Large Sociogram," Behavioral Science 6, 279-291, 1961.

[100] C. Gkantsidis, G. Goel, M. Mihail, and A. Saberi, "Towards Topology Aware Networks," in the Proceedings of IEEE INFOCOM mini-symposium, Anchorage, Alaska, May 2007.

[101] C. Nuzman, I. Saniee, W. Sweldens, and A. Weiss, "A Compound Model for TCP Connection Arrivals for LAN and WAN Applications," Computer Networks, 40:319V337, October 2002.

[102] NOMADS: Network of Mobile Adhoc Devices and Sensors, research group lead by Dr. Ahmed Helmy. Group homepage http://nile.cise.ufl.edu/

[103] D. B. Johnson and D. A. Maltz, Dynamic Source Routing in Ad-Hoc Wireless Networks, Mobile Computing, pp.153-181, 1996.

[104] S.Tanachaiwiwat and A. Helmy, "On the Performance Evaluation of Encounter-based Worm Interactions Based on Node Characteristics" ACM Mobicom 2007 Workshop on Challenged Networks (CHANTS 2007), Montreal, Quebec, Canada, Sep. 2007.

[105] P. Costa, C. Mascolo, M. Musolesi, and G. Picco, "Socially-aware Routing for Publish-Subscribe in Delay-tolerant Mobile Ad Hoc Networks," to appear in IEEE Journal on Selected Area of Communications.

[106] A. Miklas, K. Gollu, K. Chan, S. Saroiu, K. Gummadi, and E. Lara, "Exploiting Social Interactions in Mobile Systems," in Proceedings of 9th International Conference on Ubiquitous Computing, Sep. 2007.

[107] M. Thomas, A. Gupta, and S. Keshav, "Group Based Routing in Disconnected Ad Hoc Networks", in Proceedings of 13th Annual IEEE International Conference on High Performance Computing, Dec. 2006.

[108] W. Zhao, M. Ammar, and E. Zegura, "A Message Ferrying Approach for Data Delivery in Sparse Mobile Ad Hoc Networks," in Proceedings of ACM Mobihoc 2004, May 2004.

[109] W. Hsu, D. Dutta, and A. Helmy, "Profile-Cast: Behavior-Aware Mobile Networking," in Proceedings of IEEE WCNC, Las Vegas, NV, Mar. 2008.

[110] M. Motani, V. Srinivasan, and P. Nuggehalli, "PeopleNet: Engineering A Wireless Virtual Social Network." in Proceedings of MOBICOM 2005, Sep. 2005.

[111] J. Ghosh, M. J. Beal, H. Q. Ngo, and C. Qiao, "On Profiling Mobility and Predicting Locations of Wireless Users," in Proceedings of ACM REALMAN, May 2006.

[112] W. Hsu and A. Helmy, "Analysis of Nodal Encounter Patterns in Wireless LAN Traces," manuscript in preparation. Latest version available at http://nile.cise.ufl.edu/~weijenhs/SmallWorld.pdf

BIOGRAPHICAL SKETCH

Wei-Jen Hsu was born in Taipei, Taiwan, in March 1977. He received the B.S. degree in electrical engineering and the M.S. degree in communication engineering from National Taiwan University, respectively, in June 1999 and June 2001. Wei-Jen started his study towards the Ph.D. degree in 2003 at University of Southern California. He received the Engineer degree in electrical engineering from University of Southern California, in August 2006, and transferred to University of Florida to continue his study.

Wei-Jen's research interests include analysis of user data and behavior-aware protocol design. Wei-Jen is a student member of IEEE and ACM.