ANALYZING WIRELESS NETWORKS

A DISSERTATION
SUBMITTED TO THE DEPARTMENT OF COMPUTER SCIENCE
AND THE COMMITTEE ON GRADUATE STUDIES
OF STANFORD UNIVERSITY
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

Diane Tang
October 2000

I certify that I have read this dissertation and that in my opinion it is fully adequate, in scope and quality, as a dissertation for the degree of Doctor of Philosophy.

Dr. Mary Baker
(Principal Advisor)

I certify that I have read this dissertation and that in my opinion it is fully adequate, in scope and quality, as a dissertation for the degree of Doctor of Philosophy.

Dr. Pat Hanrahan

I certify that I have read this dissertation and that in my opinion it is fully adequate, in scope and quality, as a dissertation for the degree of Doctor of Philosophy.

Dr. Nick McKeown

I certify that I have read this dissertation and that in my opinion it is fully adequate, in scope and quality, as a dissertation for the degree of Doctor of Philosophy.

Dr. Jennifer Widom

Approved for the University Committee on Graduate Studies:

Abstract

In recent years, the number and variety of wireless network installations have dramatically increased, from small-scale installations spanning buildings and campuses to much larger-scale installations spanning cities and metropolitan areas. As these wireless networks proliferate, the population of users taking advantage of these networks for communication and access to online services and information also increases. To aid this growing population of users, many research and development efforts focus on upgrading and enhancing the services provided to the users.

As part of these efforts, it is crucial to analyze real wireless networks to understand better how users take advantage of them. These analyses are important for at least two reasons. First, they are helpful for creating more realistic models of users when simulating new services, which is a common technique used in the mobile networking community to predict performance. Second, they are also helpful in focusing research on topics that will impact users the most.

This thesis analyzes two very different wireless networks: the Metricom Ricochet packet radio network, a high latency, low bandwidth, metropolitan-area wireless network, and the WaveLAN network installed in the Gates Computer Science Building, a low latency, high bandwidth, local-area network. These analyses answer questions about network utilization, traffic characteristics, and user mobility rates and patterns. Among other results, we find that traffic peaks are usually caused by a single user rather than multiple users, and that significantly asymmetric network throughput would be undesirable for the WaveLAN network users. We also determine that users can indeed be categorized based on their mobility in the Metricom analysis and based on their usage characteristics (applications, time of usage, and number of remote hosts visited) in the WaveLAN trace. We also present clustering and visualization as two tools not commonly used in the networking community that are very useful in doing such analyses.

While the results from these analyses are necessarily valid only for the particular networks and user communities studied, we believe many of the conclusions will hold for similar network

environments, and the analyses also provide a blueprint for others desiring to perform similar investigations.

Acknowledgements

This thesis would not have been possible without the support I have received from numerous individuals.

First, I thank my advisor, Professor Mary Baker, for her invaluable guidance throughout my graduate career. Her many insights, questions, and advice have helped me develop into the researcher that I am today. I would also like to thank Professor Pat Hanrahan, an amazing researcher, for helping me to learn more about information visualization and sharing his vision with me as we move forward with Rivet. In addition, I thank both Professor Jennifer Widom and Professor Nick McKeown for serving on both my orals committee and reading committee. Both of them contributed remarks that improved the quality of this thesis. I would also like to thank Professor Tom Kenny for serving as the chairman of my orals committee and his insightful questions during my defense. I am also thankful to Professor Mendel Rosenblum for his friendship and advice throughout my time at Stanford.

I would be remiss if I did not also thank Professor Margo Seltzer, my undergraduate thesis advisor at Harvard, who "passed me along" to her best friend, Professor Mary Baker when I came to Stanford. She, along with Professor Mike Smith, Professor H.T. Kung, and Dr. Joe Marks, all introduced me to the research process while at Harvard.

Without the discussions, support, and most importantly, friendship, from my Rivet cohorts, Robert Bosch and Chris Stolte, I would never have finished my thesis. I am deeply indebted to the two of them.

While at Stanford, I benefitted from interacting with a number of very talented people. I am grateful to the MosquitoNet crew: Stuart Cheshire, Xinhua Zhao, Petros Maniatis, Ed Swierk, Guido Appenzeller, Kevin Lai, and Mema Roussopoulos. I am also grateful to the graphics group, who have adopted me as one of their own, especially Greg Humphreys, Matt Eldridge, Francois Guimbretiere, Tamara Munzner, Lucas Pereira, Maneesh Agrawala, and Brad Johanson.

Finally, I thank my friends and family for their support throughout my graduate career. I especially thank Elizabeth Seamans for her friendship and chocolate chips, as well as Kinshuk Govil, Dave Ofelt, Gordon Stoll, and Andrew Beers. I also thank my family, especially my cousin, Dan Chao, and my parents.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Detailed analyses of wireless networks are important for understanding how users take advantage of such networks. In this thesis, we present not only techniques and tools for doing this type of analysis, but also results and conclusions from the analyses.

1.1 Motivation

Fifteen years ago, wireless networks existed mainly in the form of packet radios, and only the military and a few research labs had access to them. Today, people can access the Internet via cellular phones, personal digital assistants (PDA's), or laptops using a variety of wireless networks, from small-scale networks, such as local-area networks spanning buildings or campuses, to much larger-scale networks spanning cities, larger metropolitan areas, and even countries.

One of the main motivations behind these wireless networks is to meet the demands of users who wish to stay connected to the network, communicating with others and accessing on-line information no matter where they are.

However, supporting these users requires not just the development of a physical wireless network infrastructure, but also the development of software providing the services users require, such as mobility support and adapting applications to changing bandwidth. As a result, mobile networking is currently the topic of many research and development efforts to upgrade and enhance wireless services for the growing number of users.

One major problem with existing research and development efforts is their dependence on assumed models of user behavior, rather than on actual observations of users. This dependence on assumptions about user behavior is troublesome for two reasons. First, researchers focus their work on areas they believe will impact users based on these conjectures. Second, the networking community commonly uses simulation to analyze the effectiveness of their research. In simulation, these assumptions of user behavior are encoded as user models, such as user mobility models or user traffic models. Thus, if these assumptions are wrong, then not only are the results questionable, but even the focus of the research itself is debatable.

Given the increasing prevalence of wireless networks, researchers and developers should base their work on actual observations of how users take advantage of wireless networks. These observations can in turn help focus research on areas that will impact users the most and can help create more realistic models of user behavior.

1.2 Contributions

The goal of this dissertation is to increase our understanding of how users actually take advantage of wireless networks by testing the expectations about users, both to verify their correctness and to test their sufficiency to see whether there are behavior patterns we have not considered.

To accomplish this goal, we present detailed analyses of two wireless networks with vastly different characteristics. We also present two tools, clustering and visualization, which are very useful when doing such analyses. The hope is that others will take these analyses and tools to do similar analyses of other wireless networks to further increase our understanding of wireless users.

The two wireless networks we analyze are a low bandwidth, high latency metropolitan-area wireless network, specifically the Metricom packet radio network, and a high bandwidth, low latency local-area wireless network, specifically the Lucent WaveLAN network installed in the Gates Computer Science Building at Stanford University. In the analyses, we answer questions such as when users use the network, how often users use the network, and how often users take advantage of the mobility a wireless network offers. We also answer questions specific to each type of network. For example, with the large user base (24,773 users in February, 1998) of the Metricom packet radio network, we examine user mobility patterns to understand better how often and when users change location. With the traffic data gathered in the WaveLAN analysis, we examine user traffic patterns to understand which applications users employ and to how many hosts users connect.

The contributions of this dissertation fall into two categories: detailed analyses of wireless networks and tools useful in doing these analyses. Specifically, the contributions are:

1. An analysis of a network trace of the Metricom packet radio network, a metropolitan-area

wireless network, to find answers to overall network questions such as when the mobile network is the most active, how active the network gets, where the network is active, as well as radio mobility questions such as how far, how often, and when users move.

Amongst other results, we verify that more users are active during the day and on weekdays, and that most users move within a local area as opposed to longer distance movements. As for our expectation that many users are mobile, we find instead that a majority of users are stationary on any given day, but that there is a small fraction of users that are highly mobile.

2. An analysis of a 12-week trace of a local-area wireless network installed throughout the Gates Computer Science Building at Stanford University. Our goal is to answer questions such as how much users take advantage of mobility, how often peak throughput rates occur, what causes the peaks, and what application mix is used.

   Some of the more surprising results in this analysis are:

   - The unexpectedly high number of users using chat software, indicating that many users are taking advantage of their laptops for synchronous communication with other users.
   - The discovery that users fall into distinct location-based sub-communities as opposed to being a single, fairly homogeneous community. Each sub-community has its own distinct behavior regarding movement and periods of activity.
   - The result regarding the symmetry of traffic: while incoming traffic dominates outgoing traffic overall (34 billion bytes compared to 12 billion bytes), the opposite tends to be true during periods of peak throughput, implying that significant fixed asymmetry in network capacity could be undesirable for the users. This result is especially relevant given the current research and development efforts on asymmetric networks, such as satellite communications or cable modems.

3. A description and explanation of the issues involved in using clustering techniques for analyzing network traces. Clustering is not commonly used in the networking community, but can be very useful in finding patterns. We also present a novel two-stage clustering algorithm that leads to more understandable patterns.

4. Three examples illustrating the effectiveness of using data-dense, interactive visualization to explore and understand network traces. Visualization is another highly useful tool not yet commonly used in the networking community.

Expectation	Result	Impact Area
More users active during the day	Verified	Network planning
More users active on weekdays	Verified	Network planning
Most users mobile	Many stationary users with fewer mobile users, but a few very mobile users	Network design (handoffs)
Fairly uniform amount of mobility across users	Large range of mobility, from stationary to highly mobile	Network design (mobility support / optimization)
Most movement local	Verified	Mobility models (local vs. long distance movement)
Only a few mobility patterns (home-office, traveling salesman)	Many (11) different patterns of mobility and time (home-office, traveling salesman, train commuters, etc.)	Mobility models (need to consider when users are active as well)
Peak usage periods caused by multiple users and applications	Most peaks caused by a single user and application	Network or hardware design
Asymmetry acceptable to users	Significant fixed asymmetry in wireless networks undesirable to users	Network design
Web & session applications most common	Verified	Application optimizations
Only a few different traffic usage patterns	More traffic patterns, users differ greatly in the number of hosts they connect to	Application optimizations (not all applicable, need to take hosts into account)
Single homogenous community in WaveLAN analysis	User sub-communities with differing behavior regarding mobility and traffic	Network planning (need to take communities into account when allocating resources)

Table 1.1: A list summarizing the research community's typical expectations regarding mobile user behavior, and my results showing whether those expectations are justified.

Table 1.1 summarizes some of the typical expectations the research community has regarding mobile users as well as presents our results regarding whether those expectations are justified and how our results might impact the mobile networking community.

While these results are only necessarily valid for the particular wireless networks and user communities we analyze, we believe that similar environments may exhibit similar behavior and trends. We hope these analyses along with the two atypical tools we use will serve several purposes:

- To test our expectations and see if our intuitive understanding of mobile users is accurate.
- To increase our understanding of mobile users and thus to create better user models.

- To focus research on the topics that will achieve the most improvement in user experience. For example, these results indicate that development of fixed asymmetric networks will be less useful than the development of symmetric networks or networks with adaptive asymmetry.
- And finally, to act as a blueprint for others when doing similar analyses to further increase our understanding of mobile users.

1.3 Thesis Outline

The rest of this dissertation is organized as follows:

Chapter 2 presents an analysis of a metropolitan-area wireless network, specifically the Metricom packet radio network. This analysis focuses not only on overall network usage issues, but also on user mobility patterns.

Chapter 3 presents an analysis of a very different type of network, a local-area wireless network installed in the Gates Computer Science Building at Stanford University. Because of the smaller scale of this network, the analysis does not contain an in-depth study of mobility patterns. Instead, this study focuses more on user traffic patterns in addition to overall network usage issues.

In Chapter 4, we discuss related network analyses and how the analyses in Chapters 2 and 3 compare.

We present the issues that come up in using clustering algorithms for analyzing network traces in Chapter 5. Clustering algorithms are especially useful for finding patterns, such as user mobility patterns, in large data sets. This chapter also further explains a novel two-stage clustering algorithm used in the Metricom packet radio analysis.

In Chapter 6, we present three visualizations illustrating the effectiveness of using interactive visualization to explore these network traces.

Finally, Chapter 7 summarizes the contributions of this work.

Chapter 2

Analyzing a Metropolitan-area Wireless Network

This chapter presents an analysis of a network trace of the Metricom packet radio network, a metropolitan-area wireless network, to find answers to overall network questions such as when the mobile network is the most active, how active the network gets, where the network is active, as well as radio mobility questions such as how far, how often, and when users move. The answers to such questions are crucial in planning a future mobile network infrastructure, and in understanding how people actually take advantage of a mobile network. While these results are only known to be valid for this particular network, we hope future analysis of other networks will add to a growing understanding of mobile network behavior.

Amongst other results, we verify that more users are active during the day and on weekdays, and that most users move within a local area as opposed to longer distance movements. Verifying these expectations allows network designers to take these gross user behavior patterns into account when planning the mobile network infrastructure, including when to prepare for peaks in network usage. As for the expectation that many users are mobile, we find instead that a majority of users are stationary on any given day, but that there is a small fraction of users that are highly mobile. This conclusion might lead network designers to decide that optimizing handoffs in this type of metropolitan-area networks is not a high priority since handoffs will not affect a large percentage of users.

In this chapter, we first present some background information about the data before presenting the actual results from the analysis. The analysis is divided into two parts: overall network behavior and radio mobility.

2.1 Background

In this section, we describe the network traced, how the data was collected, and some issues that arose in the data analysis.

2.1.1 Data Collection

The traces we study here were obtained from Metricom [12][41]. Metricom has installed a Ricochet^{TM} packet radio network infrastructure in three major metropolitan areas (San Francisco Bay Area, Washington D.C., and Seattle), as well as in some airports, hotels, and college campuses scattered across the United States. This infrastructure consists of "poletop" repeaters distributed throughout the covered areas. Each poletop is one of two types:

- a wireless repeater, which just forwards packets on to another poletop via the radio interface, or
- a wired access point, which has both a radio interface and a wired interface. Typically, wired access points have 8, 16, or 24 radios on them, as they are the focal point of many other wireless repeaters.

The range of a repeater is normally about half a mile. This range may vary depending on external conditions, such as the weather or the location of buildings and hills.

When a subscriber radio is first turned on, it scans the network for poletops, and chooses one with which to register. This choice is usually based on signal strength, but may also be based on load-balancing considerations. This poletop is responsible for forwarding all packets to and from the Metricom network on behalf of that radio. Radio registrations also occur whenever the radio changes its primary poletop and according to a predetermined pattern when the radio is stationary (see Section 2.4). Note that while a radio registers with only one poletop at a time, the radio does keep an internal list of all poletops within its range. Each registration, whether at the same or at a different primary poletop, is logged by a central nameserver. This nameserver is responsible for keeping the mappings of radio name to current location.

The trace consists of a nameserver log covering a seven-week period from February 1, 1998 through March 23, 1998, with the exception of three holes in the trace data, during which no registrations were logged: February 16, 6am through 1pm; February 16, 5pm through February 17, 4am; and February 17, 4pm through February 18, noon. The network at the time of the trace consisted of 14,053 poletops and 24,773 radios. There were a total of 7,726,678 events logged over these seven weeks. Of those, 5,982,846 are registrations and the other 1,743,832 are queries. A **registration** occurs when the radio informs the nameserver of its current primary poletop, i.e., the poletop to which packets destined for this radio should be sent. A **query** occurs when the radio queries the nameserver about some other entity in the network, such as another radio or poletop. Queries are usually made at the start of a connection, and the radio can register at different poletops while a connection is on-going.

Each log entry consists of the following information:

- a timestamp taken at the nameserver with accuracy to the second
- a sequence number
- the radio id
- the wired access point used
- the name of the radio's choice for primary poletop
- whether the entry is a registration or a query

A poletop's name is an encoding of its latitude and longitude, so a poletop's name gives its geographic location and therefore the approximate geographic location of the radio registering with it.

2.1.2 Data Analysis Techniques

In analyzing this trace, the main difficulty is in dealing with the sheer volume of data — 7,726,678 events, 24,773 radios, and 14,053 poletops — making it impractical, if not impossible, to look at each radio individually. As a result, we needed methods to make the raw data more understandable. We turned to a technique commonly used in data mining and machine learning called clustering, also known as an unsupervised learning technique. Clustering algorithms take a set of points in n-dimensional space and find coherent subsets. Each subset consists of points that are close together. The advantages of using clustering algorithms are the ability to categorize radios automatically by some metric, and the ability to find groupings that we might not otherwise find. The disadvantages are that the results are dependent on the parameters and distance functions used.

We use three different clustering algorithms:

- k-means [54], an iterative clustering algorithm,

- hierarchical agglomerative clustering [50], a tree-forming clustering algorithm,
- and expectation-maximization (EM) [15], a method to find the means and variances of a mixture of Gaussian distributions. EM is especially useful when the range of values differs widely (i.e., orders of magnitude) between dimensions.

We use both hierarchical agglomerative and k-means clustering algorithms to cluster poletops into **locations**. This grouping of poletops is needed because without physically moving, the radio may still be within range of up to 20 different poletops at the same time and may therefore register at any of these poletops. Thus, if we were to consider a radio as moving when it registers with a different poletop, we would overestimate local mobility. Instead, we determine that a radio has moved when it registers with a poletop in a different cluster (a different location). In Section 2.3, we use both EM and k-means to categorize the radios into different patterns of mobility. We discuss the clustering algorithms themselves in more detail in Chapter 5.

2.2 Overall Network Behavior

The first three questions we asked have to do with overall network usage:

1. When is the network the most active?
2. How active does the network get?
3. Where is the network the most active?

The answers to these questions can help network planners plan future extensions to the network infrastructure. Also, they provide a way both to compare a simulated network to an actual network and to provide a real basis for simulated networks.

To summarize the results presented in this section, we find that the network is more active on weekdays than on weekends, that the most active poletops handle up to 182 distinct radios an hour and handle over 1,600 registrations an hour, and that the network is most active where there is a high concentration of technical people. We now look at each question in more detail.

Knowing when users are actively using the network helps mobile network designers anticipate peak usage hours, and perhaps in designing usage plans, similar to the cellular phone networks. Thus, to answer the first question, Figure 2.1 shows that the network is more active on weekdays, especially the days in the middle of the week. This pattern holds regardless of whether we examine

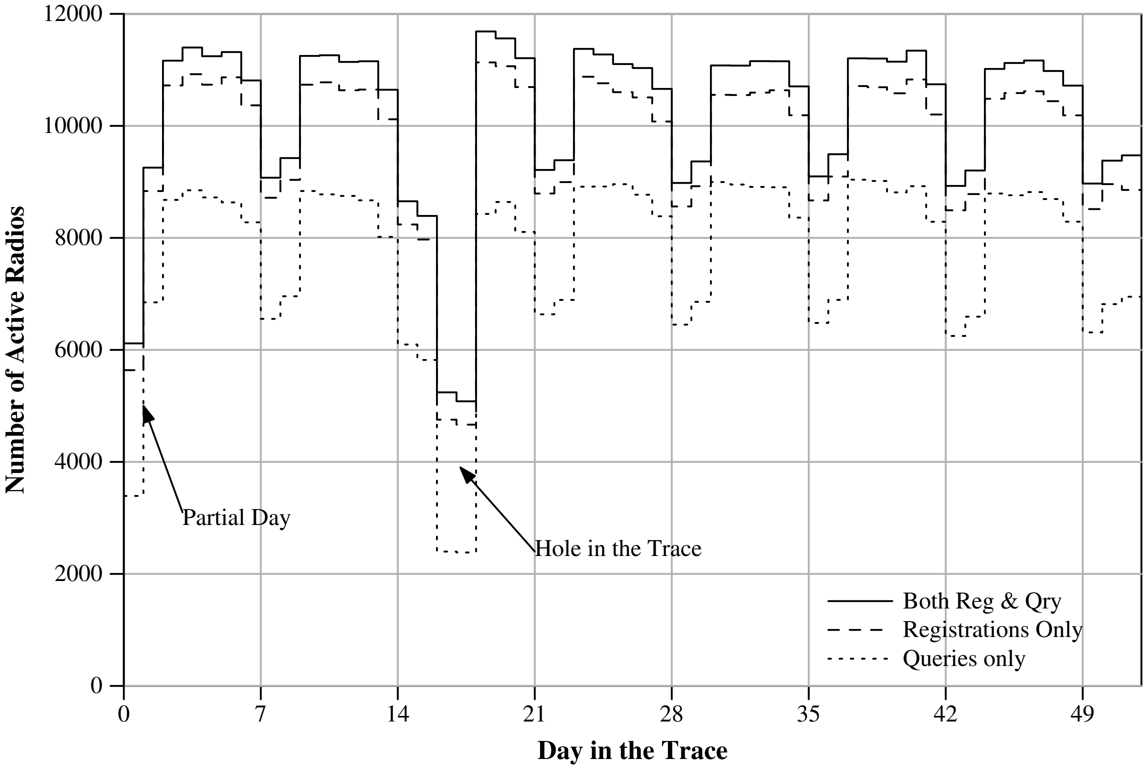


Figure 2.1: Histogram of the number of active radios on each day of the trace. One corresponds to Sunday, two to Monday, etc. There is a dip around the 16^{th} through the 18^{th} days corresponding to the holes in the trace data.

all events, registrations only, or queries only. Figure 2.2 shows that the network is least active between 3am and 5am, when most people are asleep. The two lines in Figure 2.2, one lower in the evening hours and one higher, correspond to weekend and weekdays respectively. Even in Silicon Valley, where the majority of radio users and poletops are, people are less likely to work on weekends. There is a slight decrease in the number of active radios during weekday evenings, perhaps corresponding to when people stop working for the day. There is also a slight rise around 8pm on weekends. We define an **active radio** as one that has at least one registration or query logged at the nameserver within the given time period.

The next question deals with how active the network gets. Answering this question tells network designers what peaks to expect, so that they can build into their network the capacity to handle that load. However, these peaks can be measured in several different ways. The metrics used here, given the data available in the trace, are the overall number of distinct radios a poletop has to handle over the course of the trace, the number of distinct radios poletops have to handle over an hour, and the

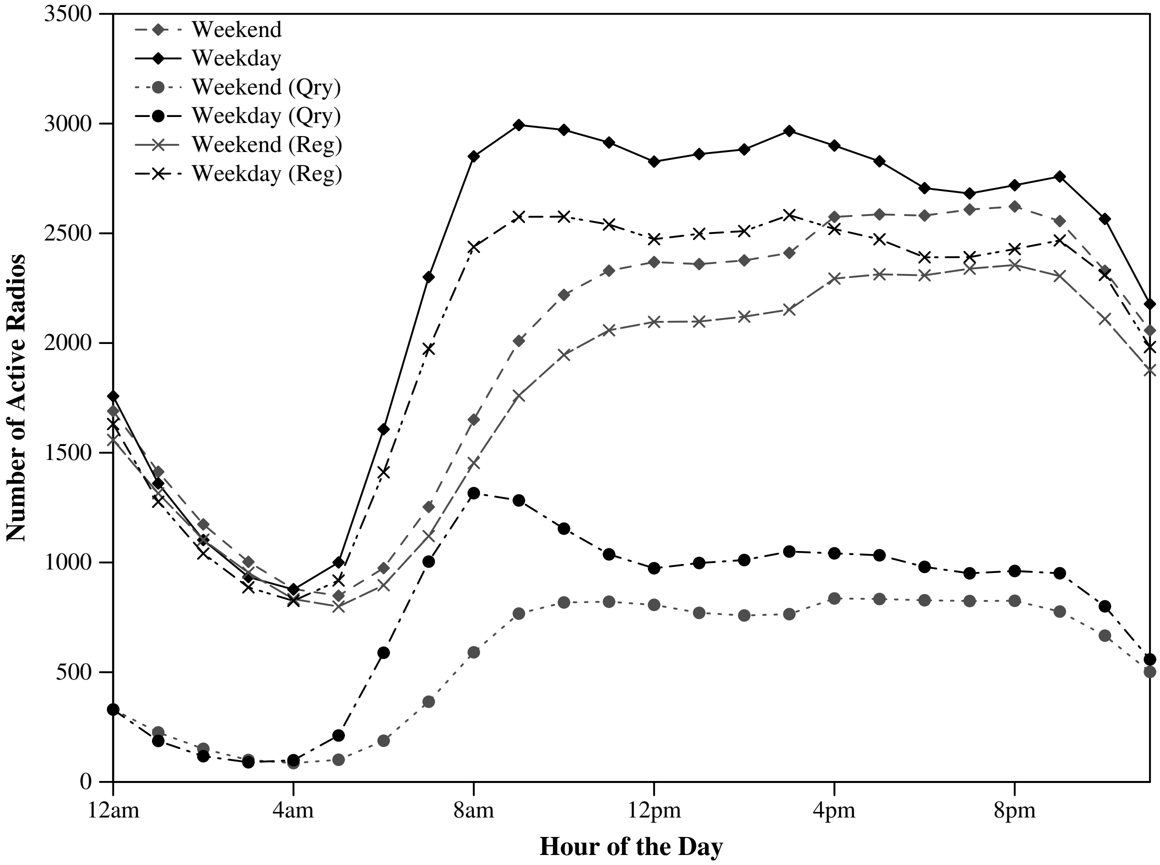


Figure 2.2: Graph showing the average, minimum, and maximum number of active radios for each hour of the trace. The two sets of lines, one lower in the evening hours and one higher, correspond to the weekend and weekdays respectively. The days containing the holes in the trace during which no events were logged are not factored in.

number of events poletops have to handle over an hour.

The first metric of network activity examined is the overall number of distinct radios that register with or query from a poletop over the course of the entire trace. This number ranges from some poletops handling only one distinct radio event over the course of the entire trace to other poletops handling 6,677 events (6,064 registrations, 434 queries), which occurs at the poletop at Metricom headquarters. Figure 2.3 shows the distribution within this range. This distribution shows that the network infrastructure could be further optimized: 23.3% of the poletops handle three or fewer radios over the course of the entire trace, indicating that some of these poletops may not be necessary, while 6% of the poletops handle 75 or more radios, indicating that some of these poletops may need additional poletops nearby to help handle the load.

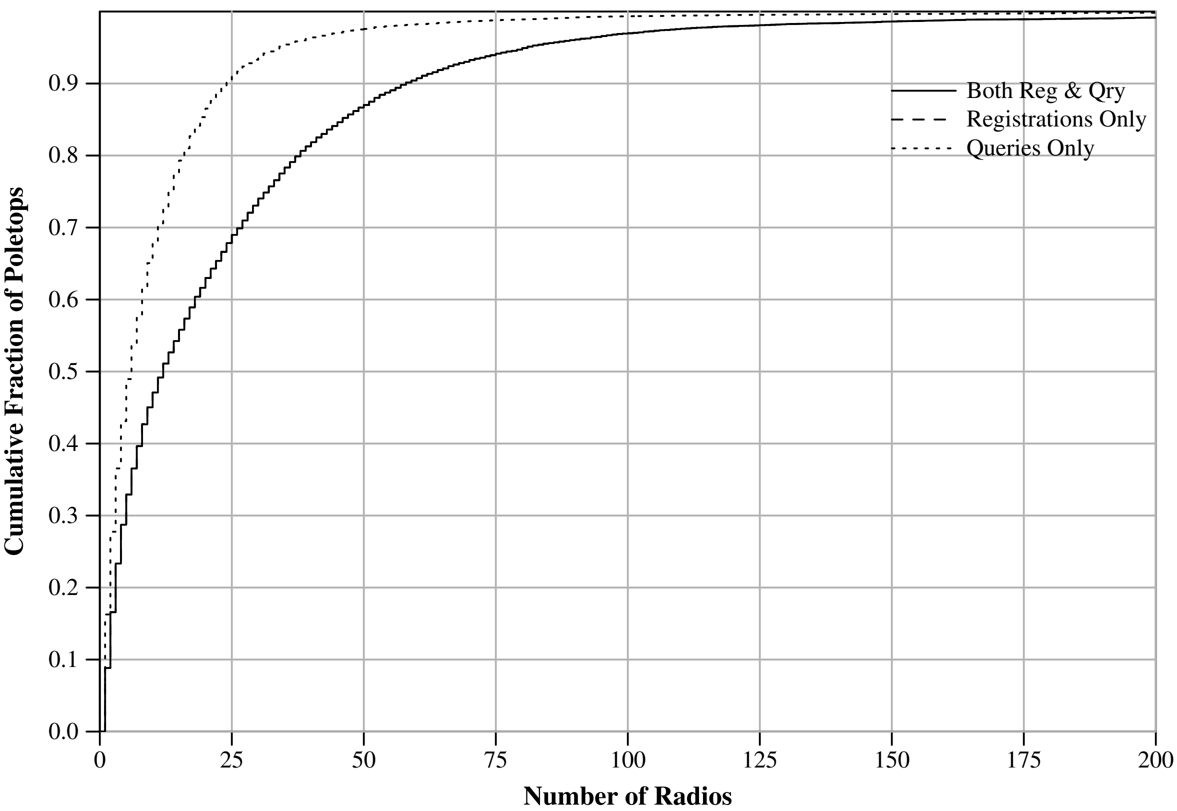


Figure 2.3: Cumulative histogram of the number of poletops (y-axis) that handle a certain number of distinct active radios over the course of the entire trace (x-axis). Note that to show the detail, the tail of the graph (which extends out to 6,677 radios for all events, 6,064 radios for registrations only, and 434 radios for queries only) has been removed.

The other metrics of network activity examined are the number of radios and radio events per hour that a poletop needs to handle. These metrics indicate the peaks the network infrastructure needs to handle. Figure 2.4 shows that while the majority of poletops only handle one radio an hour, some poletops may handle up to 182 distinct radios an hour. Figure 2.5 shows that most of the time, most poletops handle 500 or fewer radio events per hour, although some particularly busy poletops handle up to 1,600 events in an hour. Note that the shapes of both the dark areas and the peak values in Figure 2.4 resemble Figure 2.2 with the low points falling between 3am and 5am, whereas in Figure 2.5, only the dark areas at the bottom portion of the graph resemble Figure 2.2. The peaks in Figure 2.5 do not follow the rise and fall of radio activity. This lack of a clear rise and fall during peak hours could be due to frequent radio registrations regardless of radio activity.

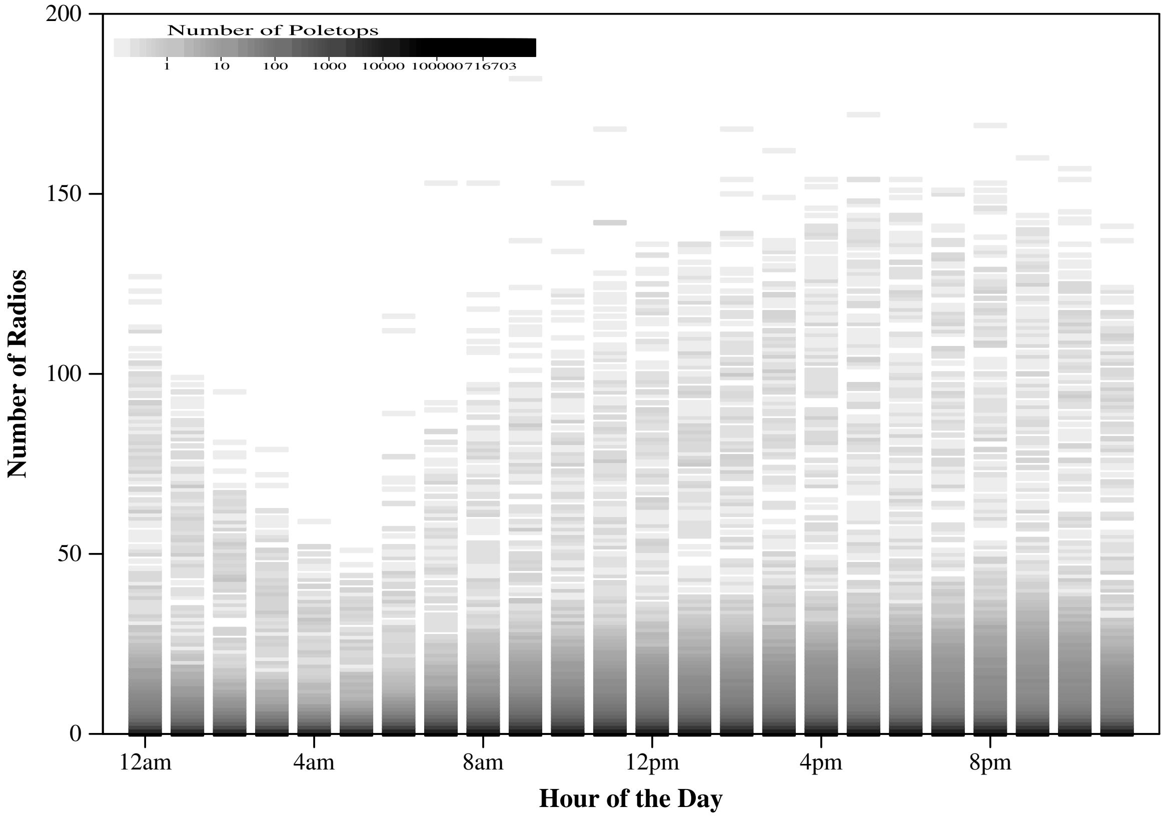


Figure 2.4: Two-variable histogram, where the darkness of the bar reflects the number of poletops that receive events from a given number of radios, by hour of the day. No matter what time of day it is, most poletops see events from fewer than 10 radios in an hour. The largest number of radios a poletop receives events from within an hour is 182. Note that the maximum number of radios at any point in the graph is the total number of poletops (14,053) multiplied by the number of times that hour occurs during the trace (once for each of the 52 days).

We also look at how often the network needs to handle events, both in the overall network and per poletop. Knowing how often events occur is useful for determining noticeable downtimes, both for key services in the overall network such as the nameserver and for individual poletops: if even during non-peak hours the time between events is very short, then the noticeable downtime is also very low.

In the overall network, the three largest intervals between two successive events by any radio, discounting the holes in the trace, are 1293 seconds (21 minutes), 744 seconds (12 minutes), and 59 seconds. Each of these intervals occurs only once. Figure 2.6 shows that by far, however, the network had either no time, or one to two seconds between events. Looking only at registrations in

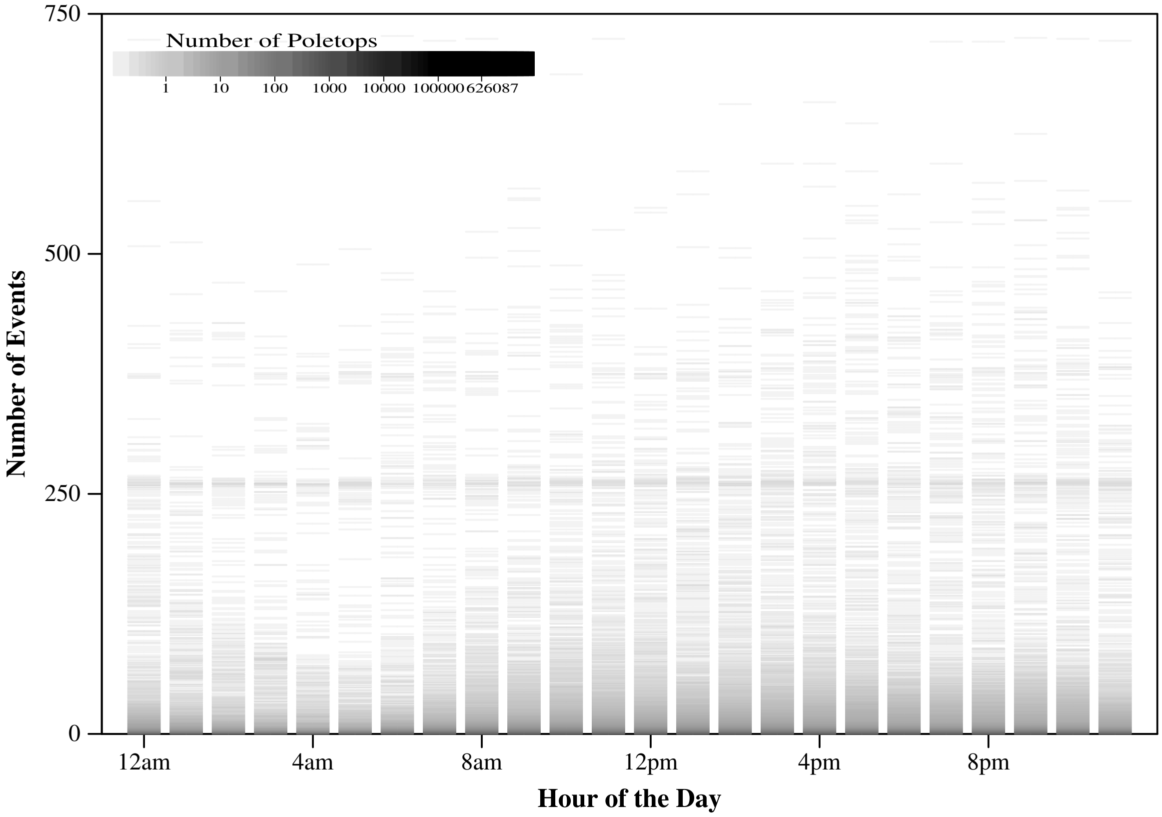


Figure 2.5: Two-variable histogram, where the darkness of the bar reflects the number of poletops receiving a given number of events per hour, by hour of the day. Note that the top of the graph, which extends to 1,601 events, has been cut off to show the detail.

the network, the distribution of intervals is almost identical. However, if looking only at queries, the curve is shallower, with some larger intervals up to 10 to 12 seconds occurring more frequently. This result is due to the many fewer queries than registrations that occur. However, this result indicates that the tolerable downtime for up-to-date read access on the nameserver database is larger than for write access, which can be logged and played back anyway.

Figure 2.7 shows the cumulative distribution of intervals between events at distinct poletops, rather than in the overall network. This figure differs significantly from Figure 2.6, with a much longer tail, due to different poletops having different loads. Some poletops are basically inactive, processing only two or three events during the entire trace, while other poletops handle thousands of events in an hour. One similarity between the two graphs is the lower curve for queries only, in both cases due to there being fewer queries overall. The distribution of times between events shown in Figure 2.7 indicates that tolerable downtimes for poletops is longer than for the overall network.

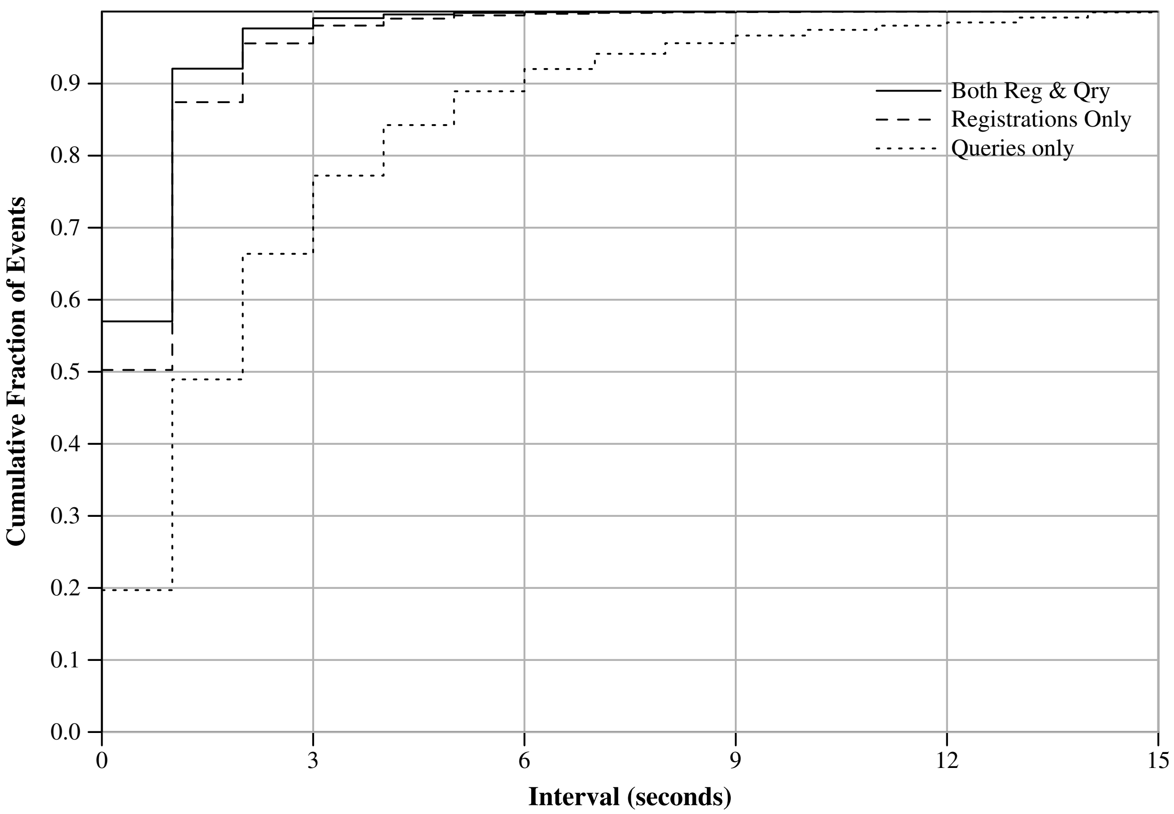


Figure 2.6: Cumulative histogram of the interval between successive events over the entire trace as a function of how often that interval occurs. For clarity, the tail (which extends to 1,293 seconds) is not plotted.

Finally, Figure 2.8 shows the most active poletops in the San Francisco Bay Area. A picture of the entire United States, covering all areas in which Metricom has established an infrastructure, is too crowded, so the figure shows just the Bay Area since it has the highest concentration of poletops. The darker dots correspond to the more active poletops. Several hot spots are worth noting. First, the dark area in San Francisco corresponds to the Financial District. The second strip of scattered dark dots running northwest-southeast corresponds to Highway 101, on which many high tech companies have their headquarters. Another hot spot is the dark dot in the upper part, which corresponds to Berkeley. Also, towards the bottom is a hot spot corresponding to Metricom headquarters. The lowest island of activity corresponds to Santa Cruz. This figure, as a follow-on to the results in Figure 2.3, indicates which poletops in which geographical areas might need more infrastructure to handle the load, and which areas (mostly around the fringes, near the mountains) may not yet need as much infrastructure as is provided.

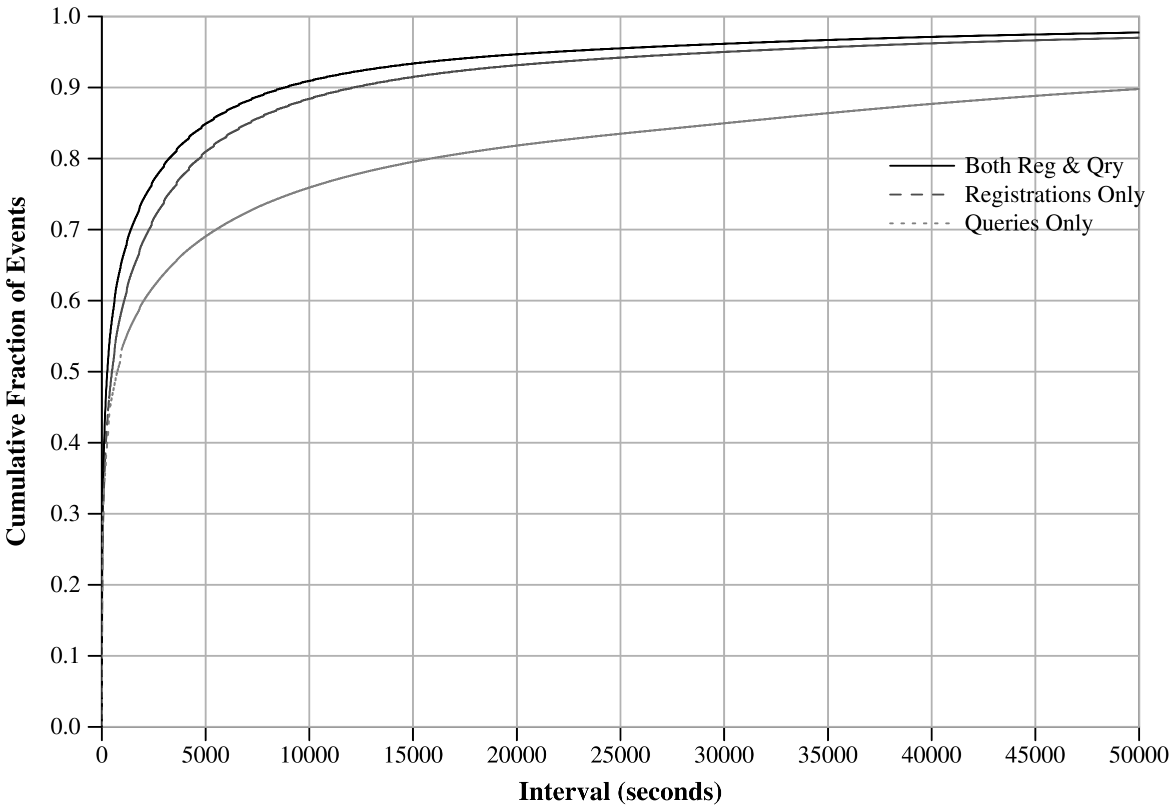


Figure 2.7: Cumulative histogram of the interval between successive events at a poletop as a function of how often that interval occurs. For clarity, the tail of the graph (which extends out to 4,319,696 seconds (71,994.9 minutes, 1199.9 hours, or 49.9 days) for all events and registrations only, and to 4,370,269 seconds (72,837.8 minutes, 1213.9 hours, or 50.6 days) for queries only) is not plotted.

We also want to investigate how the activity in the network is distributed geographically, rather than temporally as was shown in Figure 2.6. Now we look at the interval in distance between successive events in the entire network. As seen in Figure 2.9, there are several common distances: 100 miles or less, around 650 miles, around 1000 miles, 1400 miles, 2000 miles, and 2400 miles. These distances all correspond either to traffic within a Metricom installation or to the approximate distances between various Metricom installations. Because the Bay Area contains about 65% of the poletops in the Metricom infrastructure, distances of 100 miles or less are prevalent. In other words, two successive registrations in the network are more likely to both be in the Bay Area than to be distributed across the country. However, Figure 2.9 also shows a Gaussian distribution around each of the common distances. This implies that given the radius of the network, the distance between registrations of any radio in the network is likely to be a Gaussian distribution around the

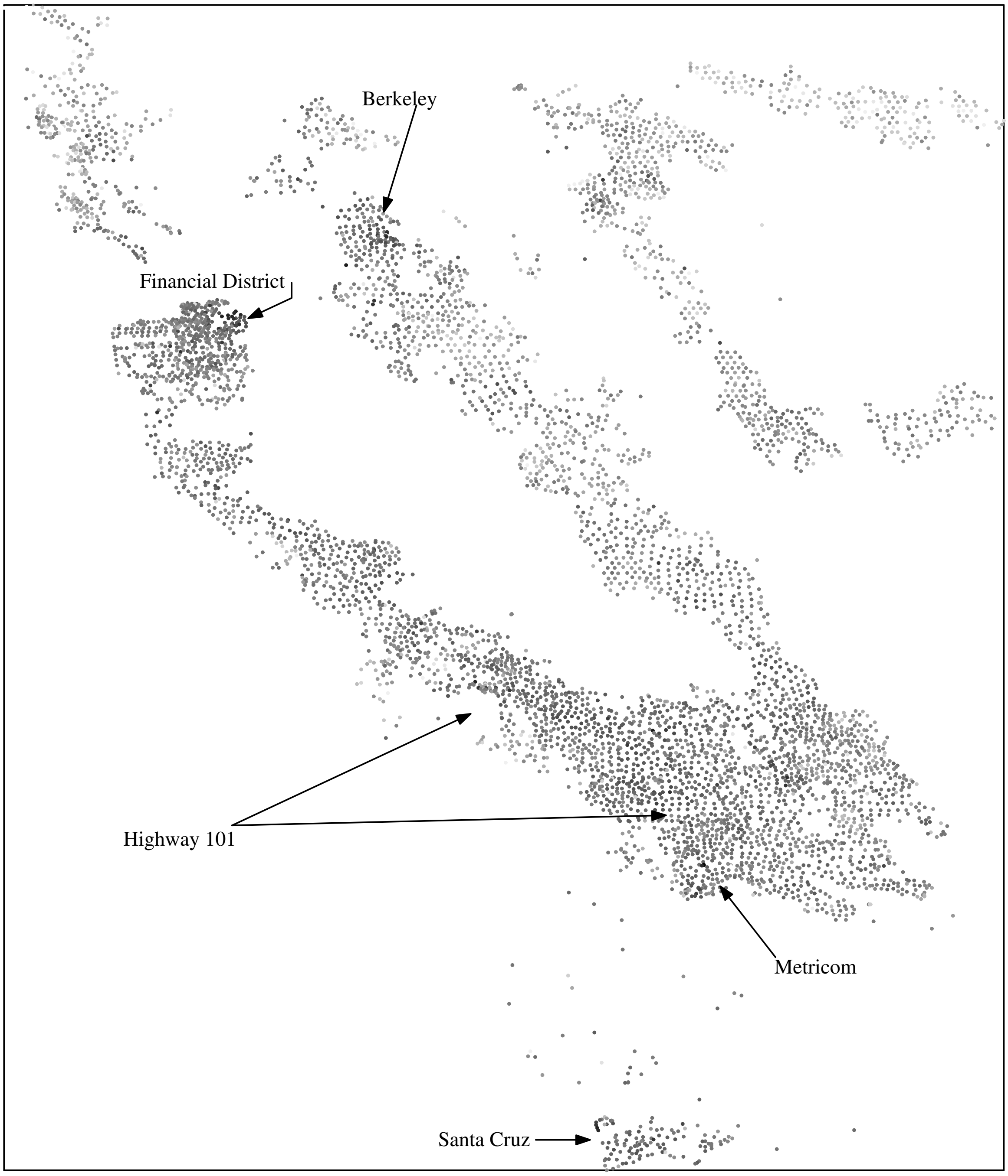


Figure 2.8: Picture of the San Francisco Bay Area. Each dot corresponds to a poletop. The darker the dot, the more radios visit that poletop over the course of the trace.

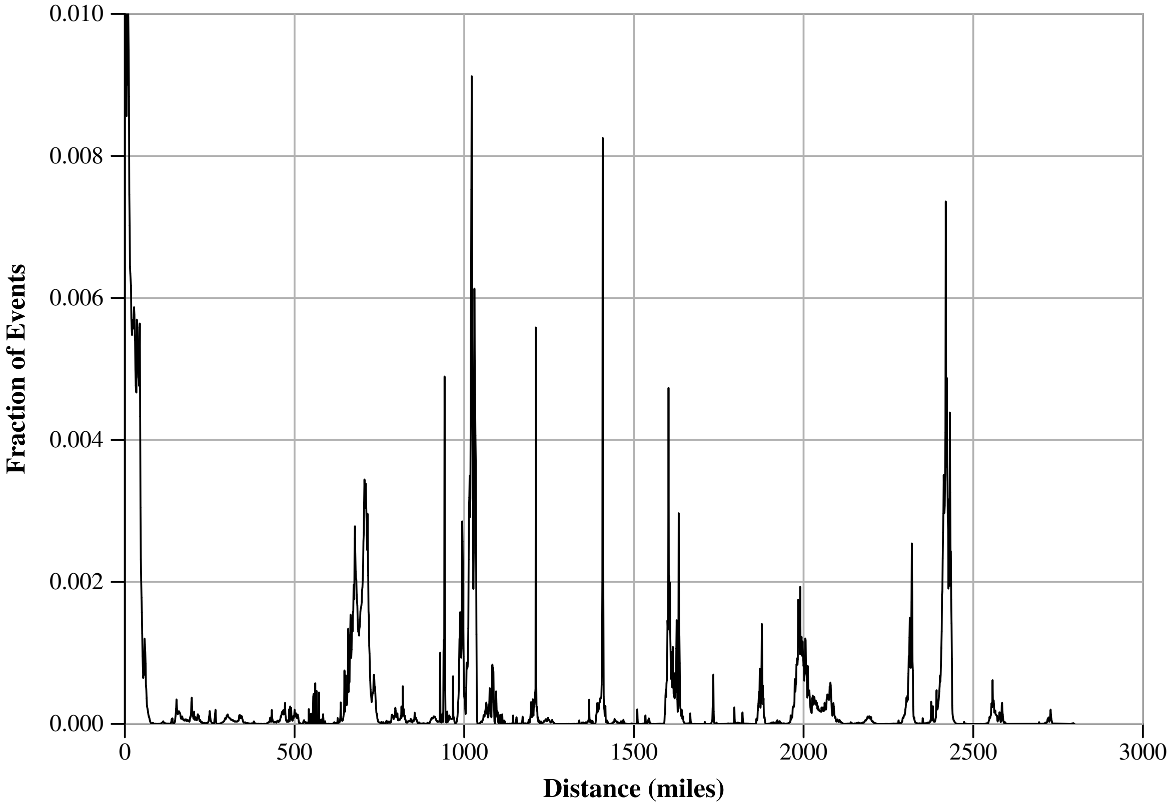


Figure 2.9: Histogram of the distance between successive events in the network by any radio as a function of how often that distance occurs.

radius. This distribution could be useful as a sanity check when simulating large mobile network infrastructures.

2.3 Radio Mobility

The other major set of questions we want to answer concerns radio mobility:

1. How often do radios move?
2. How far do they move?
3. Can we identify patterns of mobility?

Answering these questions is crucial for understanding whether and how people actually take advantage of a mobile environment. Also, understanding current radio mobility helps in choosing

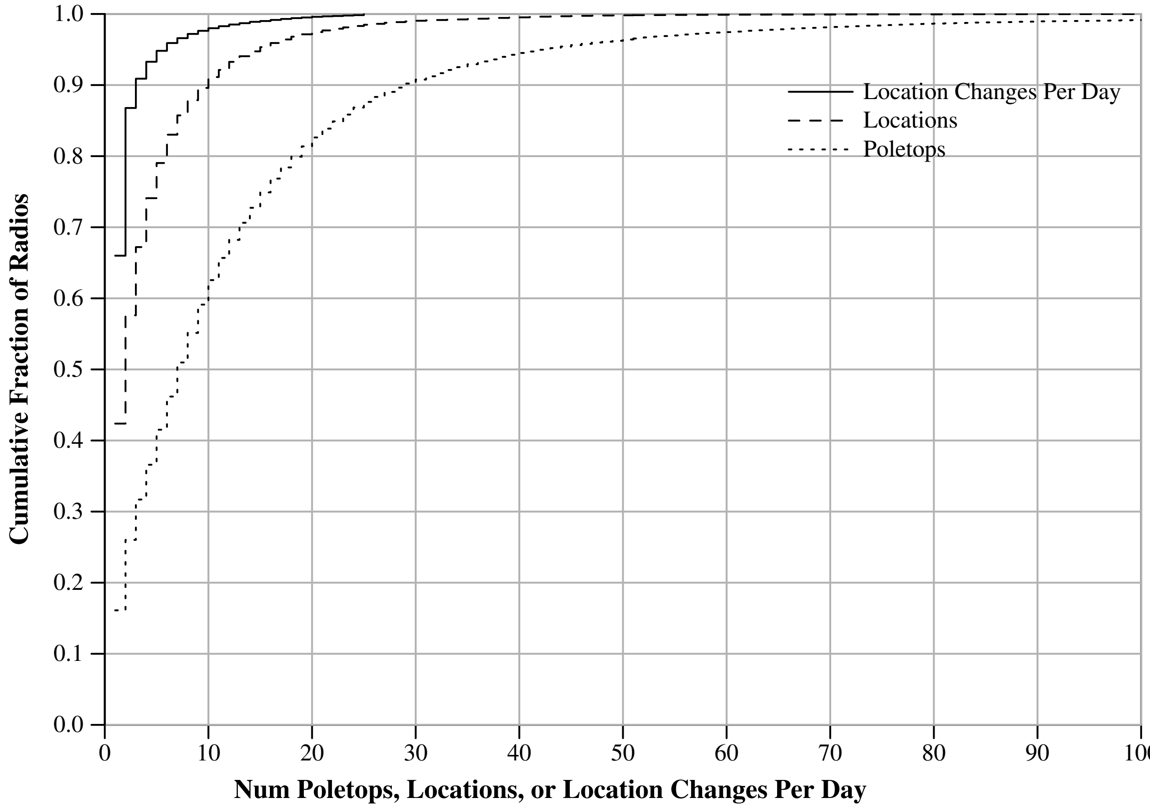


Figure 2.10: Cumulative histogram of the percentage of radios that visit a given number of poletops, locations, or locations per day, over the course of the trace. For example, 64% of radios visit only one location per day, 42% visit only one location over the course of the entire trace, and 16% of radios visit only one poletop over the course of the entire trace. Note that to show the detail, the tail of the graph (which extends to 25 locations per day, 176 locations, and 1,025 poletops) has been cut off.

parameters for simulations of mobile networks. Unrealistic movement models lead to unrealistic simulation results.

To summarize the results in this section, we first find that users who are mobile do not move frequently, and that 64% of all users only appear at one location per day. As for how far users move, most users move within their local area (i.e., easily reachable by car), with fewer users traveling the long distances between different Metricom installations. In addition, as the number of locations visited by a user increases, the average distance traveled between each location decreases. Finally, we are able to find patterns of mobility for users, such as the number of users who are active both day and night versus users active only during the day. We now examine these findings in more detail.

We first calculate the number of different poletops and locations at which radios register over the course of the trace, indicative of the overall mobility of radios. Contrary to the expectation that most, if not all, users are mobile, Figure 2.10 shows that in fact, 42% of all radios are stationary throughout the course of the trace while an additional 22% of all radios visit only one location a day (i.e., stationary on any given day, but may change location on a day by day basis). This implies that although users do move around with their radios, the movement is not very frequent. This graph also shows that more users move on a day-to-day basis rather than during a single day. This graph also emphasizes the difference between poletops versus locations. While 42% of radios are stationary with respect to location, only 16% of radios are stationary with respect to poletops, meaning that radios often register with poletops that fall in the same cluster or general area. Our location finder underestimates user mobility in that there may be more actual locations within a local area than we have found (see Chapter 5). However, because the potential underestimation only applies to mobility within a local area of approximately half a mile, movements between locations farther apart are accurate.

We also measure how many radios appear, i.e., register or query, at a different location from their last appearance. This result, shown in Figure 2.11, is indicative of when users move, or change location. While the shape of the curve mimics the curve in Figure 2.2, the overall numbers are much lower, ranging from 122 (0.5%) to 1,484 (6.0%) radios changing location at a time. Again, the lower numbers of mobile users in comparison to the total number of users contradicts the expectation that many radios are constantly mobile. There are, however, some interesting features to the weekday curve in Figure 2.11. First, there is a dip around 11am to 2pm, approximately corresponding to lunchtime, showing that many people do not take their laptops to lunch with them. Second, there is an increase in the number of radios changing location around 5pm, about when many people stop working for the day. The dip in the weekday evenings similar to Figure 2.2 occurs in this graph as well.

We next look at how far a radio moves when it changes its primary poletop or location. Figure 2.12 shows the distribution of how often radios move a given distance, in terms of movement both between poletops and between locations. Over 79% of the events (or 80% of registrations, 90% of queries) involve no change in location. This high number is due in part to the frequent radio re-registrations (see Section 2.4). Then, looking at Figure 2.13 shows that while the majority of successive events occur within 50 miles of one another, there are small peaks corresponding to the long-distance routes between airports and major cities in which Metricom has established an infrastructure. These two graphs together indicate that most users, as might be expected, move within a

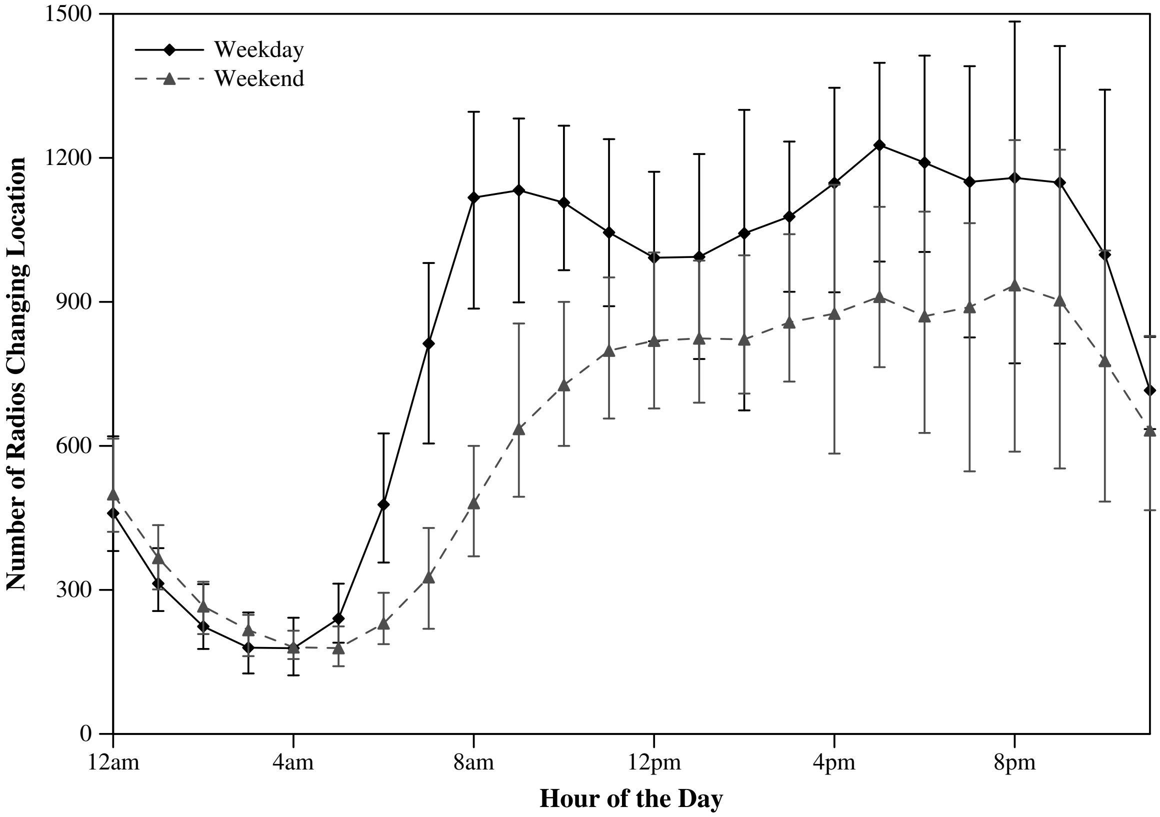


Figure 2.11: Histogram of the average, minimum, and maximum number of radios changing locations during a particular hour of the day. The time plotted is the second of two successive events.

local area. However, some users do take their radios on longer trips crossing the various network islands and use their radios at both ends of their trip.

We further examine radio movement in Table 2.1, which shows the breakdown of radios by how often they move, both overall throughout the trace and per day, and how far they move. The trends pointed out in this table need to be taken into account when creating user mobility models.

First, a radio that visits more locations overall does not necessarily also visit proportionally more locations per day. For instance, the majority of radios that visit three or more locations overall in the trace still only appear at one to two locations on a daily basis. Furthermore, only 3.5% of radios that visit more than four locations overall appear at more than four locations per day on average. (Note that one distinct location may be counted as multiple visited locations per day. For example, a user that visits location A followed by location B before returning to location A in one day is counted as appearing at three locations in that day.) This result implies that users who visit many locations do not necessarily move around more on a daily basis. Rather, they visit the different locations over a

Num Loc's	Num Radios (%)	Average Distance (miles)	Median Distance (miles)	Num Locations per Active Day (max)	Num Radios (%)	Average Distance (miles)	Median Distance (miles)
1	10,459 (42.2%)	0	0	1 (1.0)	10,459 (100%)	0	0
2	2,746 (15.1%)	6.9	2.9	1 (1.0)	2,152 (84.1%)	11.6	1.6
				1 - 2 (1.98)	510 (13.6%)	2.1	1.7
				2 (2.0)	84 (2.3%)	1.7	1.3
3	2,371 (9.6%)	5.2	1.6	1 (1.0)	1,367 (57.7%)	11.3	1.7
				1 - 2 (1.98)	914 (38.5%)	3.5	1.6
				2 (2.0)	47 (2.0%)	2.3	1.9
				2 - 3 (2.92)	31 (1.3%)	2.3	2.0
				3 (3.0)	12 (0.5%)	2.2	2.2
4	1,694 (6.8%)	4.2	1.5	1 (1.0)	569 (33.6%)	8.4	1.6
				1 - 2 (1.98)	1,013 (59.8%)	4.6	1.5
				2 (2.0)	44 (2.6%)	2.3	1.5
				2 - 3 (2.98)	55 (3.2%)	2.1	1.5
				3 (3.0)	3 (0.2%)	2.0	1.9
				3 - 4 (3.65)	8 (0.5%)	1.5	1.3
				4 (4.0)	2 (0.1%)	7.9	2.7
> 4	6,503 (26.3%)	7.9	2.4	1 (1.0)	438 (6.7%)	12.2	1.7
				1 - 2 (1.98)	4,434 (68.2%)	11.2	1.8
				2 (2.0)	126 (1.9%)	4.2	1.3
				2 - 3 (2.98)	1,020 (15.8%)	4.9	1.7
				3 (3.0)	36 (0.5%)	6.0	1.7
				3 - 4 (3.96)	210 (3.2%)	4.3	2.0
				4 (4.0)	13 (0.2%)	4.3	2.0
				> 4 (24.48)	226 (3.5%)	8.0	6.9

Table 2.1: How far users move given the number of locations throughout the trace and locations per day a user visits. For example, 510 users visit only two locations throughout the entire trace, but on average, visit between one and two locations per day.

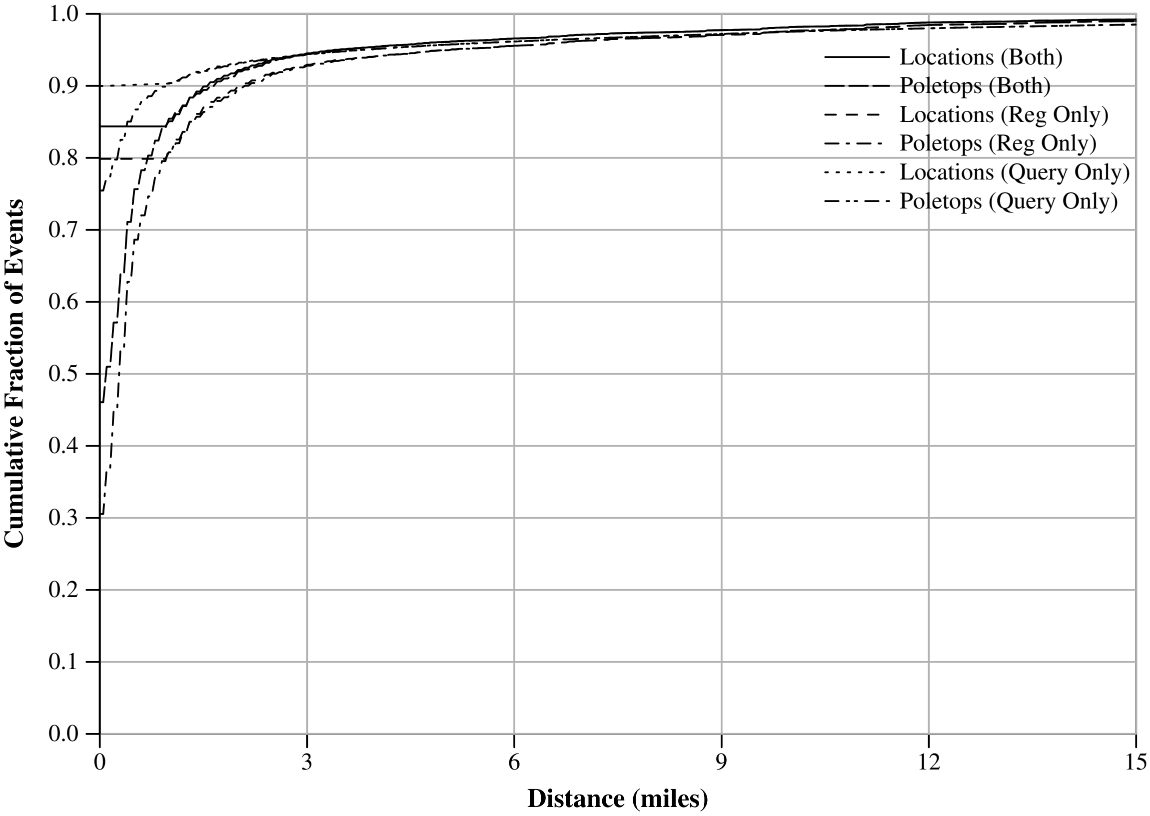


Figure 2.12: Cumulative histogram of the number of events as a function of the distance traveled. Note that the tail of the graph (which extends to 2,576 miles) has been removed to show detail. Also note that each pair of corresponding poletop and location lines converge at one mile. The straight line between 0 and 1 miles for the location graphs reflects the fact that locations cluster poletops together: poletops that are close together are considered to be a single location, and therefore represent a movement of distance zero.

longer period of several days or weeks, which is a statistic that could be taken into account when designing user mobility models.

More surprisingly, as the number of locations visited on a daily basis increases, the average distance traveled decreases. In other words, the more someone moves around each day, the closer the locations are. This overall trend may correspond to the flatter distribution of the range of distances among users who visit multiple locations per day, although the majority of locations are still relatively close together. This distribution could stem, for example, from users using their laptops on a train, such as the train that runs along Highway 101, causing them to "visit" numerous locations on their way to work.

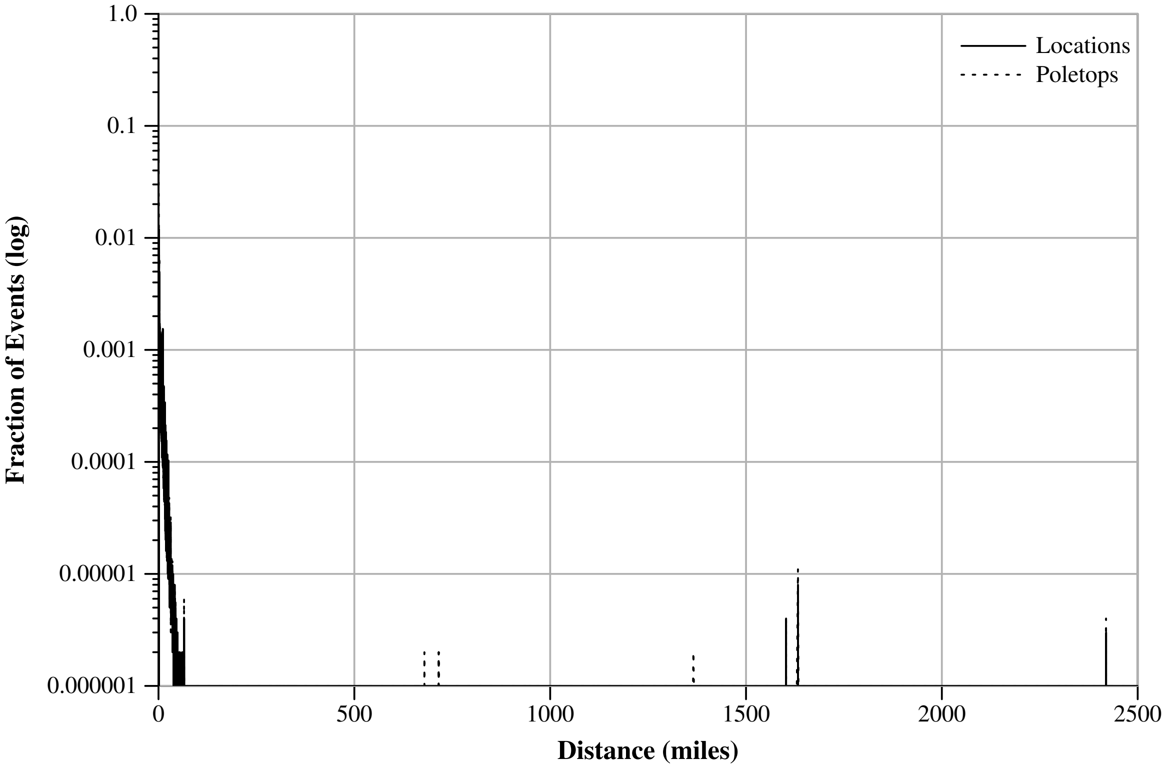


Figure 2.13: Histogram of the percentage of successive events from a radio as a function of the distance traveled between those events. Note the logarithmic scale on the y-axis.

Users who visit one location per day, independent of the total number of locations visited throughout the trace, have a large discrepancy between the average distance traveled and the median distance traveled. This discrepancy is the result of these users actually being split into two sub-groups: those who travel locally (for example, people who take their laptops home on the weekends), and those who travel long distances (perhaps people who only use their laptops when traveling). The median distance traveled reflects the usage patterns of this first sub-group, while this second sub-group of users increases the average distance traveled for all the groups of users who, independent of the total number of locations visited throughout the trace, only visit one location per day. This split into sub-groups indicates that not all users behave similarly, another consideration when modeling user mobility.

Despite this overall decreasing trend in the average distance, the median distance stays relatively constant, reflecting the many users who only move short distances. Further, the average and median distances, in terms of the total number of locations visited rather than the average number of locations visited per day, are also relatively constant, implying that movement is dependent more on how many locations a user visits on a daily basis than on how many locations a user visits overall.

2.3.1 Radio Queries

Before we look into categorizing radios into different patterns of mobility, we examine radio behavior with respect to queries. Queries are usually made when a radio makes a new connection, although some radios are used as part of Metricom's diagnostic utilities and make many more queries than are normal. Note that a query is made regardless of whether the user is using the radio in modem mode (connection-based mode) or StarMode (packet mode) [12]. If the radio is being used in StarMode, the radio could have multiple outstanding connections at a time. Since we do not have information about the end of a connection, we do not account for this situation. Using the radios in modem mode is prevalent, however. Because very few radios are used for diagnostic purposes and very few radios are used in StarMode (less than 0.1%), we believe that our results are accurate.

Our main interest with regard to queries is in how much activity there is between queries, or, in other words, how much activity is associated with each connection, both in terms of events between connections and movement between connections. Answering these questions can help network maintainers determine how effective the network is at maintaining connections across changing user state.

We first look at how many times radios register between queries. Figure 2.14 shows the cumulative distribution of how many registrations occur before a query, which is indicative of the amount of activity between the beginnings of different connections. Because 30% of queries occur after no previous registrations, and 53% occur after only one registration, this distribution implies that there is very little activity between the start of consecutive connections.

We take a closer look to see how much users move between the start of consecutive connections: in Figure 2.15, we examine how often a radio changes its location or poletop before starting a connection. This graph is indicative of how much movement there is between connections, and therefore how good the network should be at maintaining a connection when the user moves. Note that a connection can be long-lived and last for days, explaining the tail of the graph, which extends out to 1,193 locations or 3,334 poletops. Also note that we count the number of changes in location; if a radio visits location A then B then C then B and then A again, we count this sequence as four location changes. Figure 2.15 shows that 70.4% of connections have no movement before them at all, while 95% of radios change location fewer than four times. In other words, most radios do not move around much between connections. Unfortunately, this trace does not provide any information about when connections terminate so we cannot determine how much of the movement

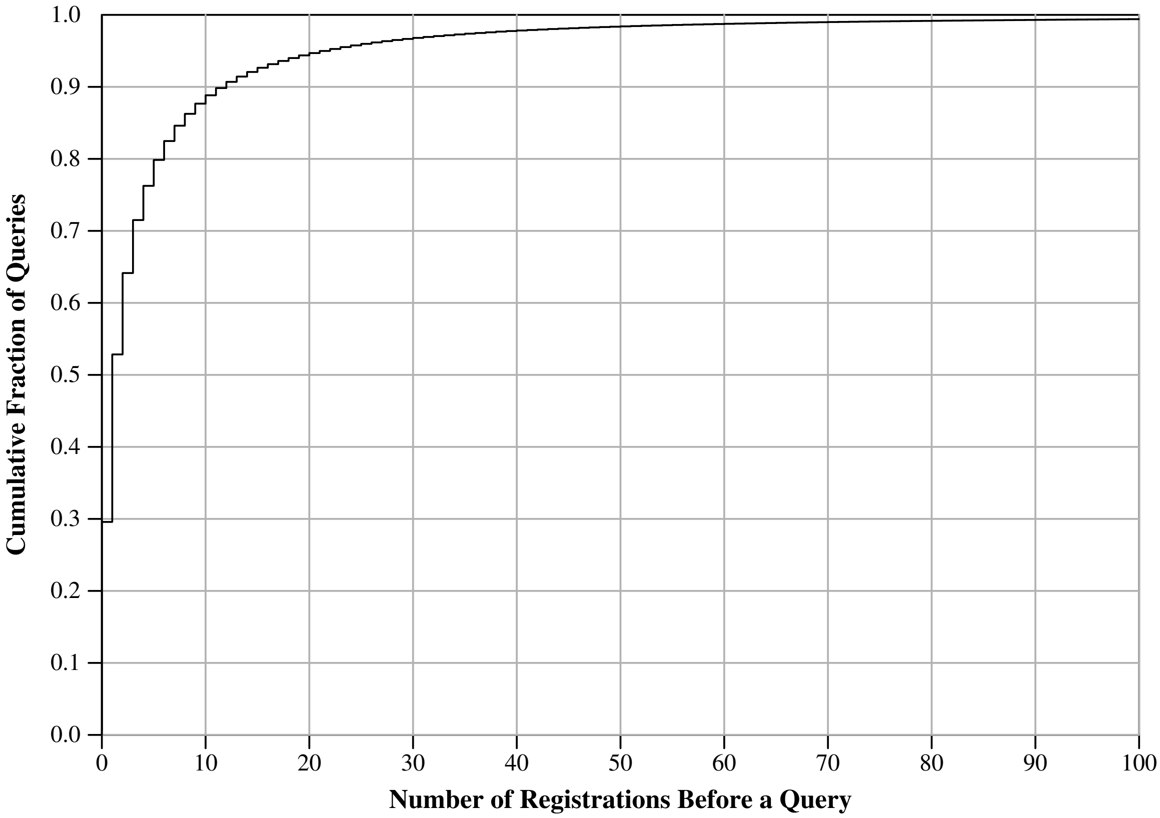


Figure 2.14: Cumulative histogram of the number of times a certain number of registrations occur before a query. Note that to show the detail, the tail of the graph (which extends to 7,948 registrations between queries) has been cut off.

occurs while the connection itself is maintained. As a result, we do not know if the user wanted to maintain the connection across more movement and the network dropped the connection when the user moved, or if the user actually disconnected, moved, and then reconnected. This lack of information is attributable to the data collection rather than the analysis.

2.3.2 Patterns of Mobility

Our final goal is to categorize the radios into different patterns of mobility. One expected pattern, for example, might correspond to a user who moves between using the radio at home and at the office. Another pattern would correspond to the user who just uses the radio at home. Yet another pattern might correspond to a traveling salesman who uses the radio to keep in touch with the warehouse. These patterns are interesting for seeing whether and how people take advantage of mobility. They

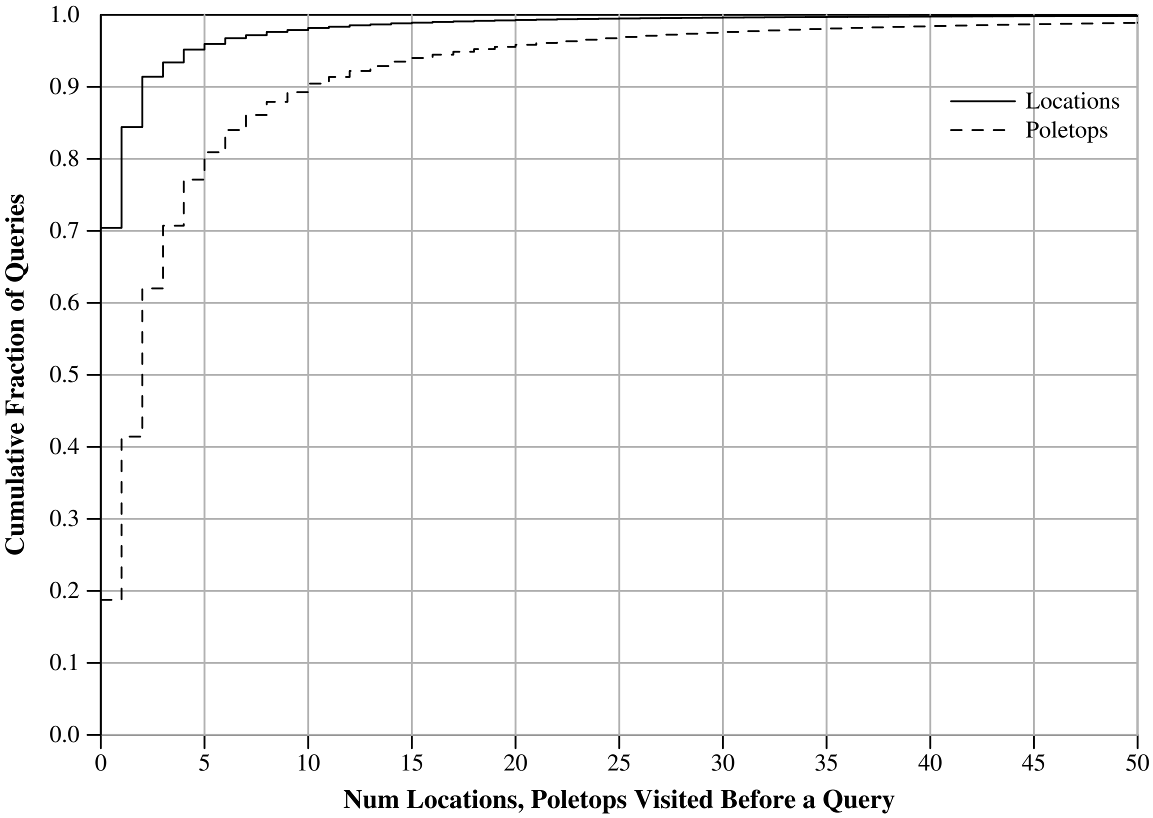


Figure 2.15: Cumulative histogram of how often a certain number of poletops or locations are registered at before a query is made. Note that to show the detail, the tail of the graph (which extends to 1,193 locations and 3,334 poletops) has been cut off.

can also be used to create models of user mobility for use in simulating mobile networks.

Without comprehensively going through the radios individually, there is no way to find all the categories, much less categorize each radio, without using clustering algorithms. We use a two-stage clustering because it avoids the difficulty in finding a single stage clustering that balances both the location and time information together. By separating out the clustering into two stages, we can first cluster the radios based on how mobile they are and on some overall frequency parameters. Once we know how mobile the radios are, we can adjust the second stage to reflect the results of the first stage, and sub-cluster based on when radios are active.

More specifically, in the first stage, we cluster the radios based on mobility using nine parameters:

- the total number of unique locations a radio visits throughout the trace,

- the total number of unique locations a radio queries from throughout the trace,
- the average number of locations a radio visits per active day,
- the average number of locations a radio queries from per active day,
- the total number of active days for a radio,
- the total number of query days for a radio,
- the average number of active days in an active week for a radio,
- the average number of query days in an active week for a radio, and
- the average number of query days per week for a radio.

An **active day** is defined as a day during which the radio visits some poletop, a **query day** is defined as a day during which the radio makes at least one query, and an **active week** is defined as a week during which at least one day is active. We use EM (expectation maximization) [15] to do this first-stage clustering, since EM takes the differing ranges of the parameters into account, whereas the other clustering algorithms, such as k-means, do not.

We then take each cluster resulting from the first stage, and sub-cluster it based on more specific time information, namely the times of day that the radio is used. Specifically, for each radio we create two histograms counting the number of times the radio has an event per hour of the day, one for weekdays and one for weekends. These histograms show when the radio is active. We then normalize these histograms and use them as the parameters for the second-stage clustering, thus grouping radios with similar active times. We do this sub-clustering, rather than just re-cluster the entire space of radios, to show how each mobility pattern changes by time of day.

We now look at the results from the first-stage clustering and second-stage sub-clustering, shown in Tables 2.2 and 2.3, respectively. The first-stage clustering results in eleven clusters. Cluster 1 consists of radios that are essentially not used during the trace. Cluster 2 is comprised of stationary radios, perhaps the ones used on desktop machines for home connectivity. Cluster 3 consists of radios that are minimally mobile, i.e., users that predominantly use their radios from a single location, but may occasionally travel to the office to download some software or data. Clusters 4, 5, 6, and 7 all consist of radios that, while they do visit several locations, do not necessarily visit all of those locations during a single day. In other words, their movement is spread out over a larger amount of time. The main difference between these three clusters is in how often the radios are used, with the

#	# of Radios	Total Loc's (all, σ, qry only, σ)	Avg Loc Changes per Active Day (all, σ, qry only, σ)	Total Active Days (all, σ, qry only, σ)	Avg Active Days per Active Week (all, σ, qry only, σ)	Avg Queries per Day (σ)	Description
1	5,877	1.02 (0.17) 0 (0)	1 (0) 0 (0)	2.98 (7.34) 0 (0)	1.41 (0.95) 0 (0)	0 (0)	Not Used
2	4,608	1 (0) 1 (0)	1 (0) 1 (0)	27.65 (18.43) 20.31 (15.69)	3.98 (1.94) 3.12 (1.68)	2.41 (2.21)	Stationary (home connectivity)?
3	4,765	2.98 (1.50) 1.60 (0.81)	1.40 (0.69) 1 (0)	22.66 (17.02) 16.04 (13.84)	3.51 (1.79) 2.77 (1.49)	2.28 (1.51)	Minimal mobility, moderate usage
4	1,117	5.59 (3.67) 2.87 (2.12)	2.81 (1.66) 1.26 (0.77)	5.63 (3.21) 3.78 (2.85)	1.76 (0.60) 1.38 (0.86)	3.91 (4.32)	Moderate mobility, minimal usage
5	1,422	5.13 (2.53) 3.65 (1.57)	1.50 (0.41) 1.16 (0.10)	15.50 (5.75) 13.63 (5.17)	2.66 (0.65) 2.51 (0.63)	2.42 (1.05)	Moderate mobility, low usage
6	1,603	4.55 (2.32) 3.17 (1.26)	1.39 (0.36) 1.12 (0.10)	35.22 (6.88) 31.23 (6.26)	4.69 (0.73) 4.24 (0.70)	2.59 (1.23)	Moderate mobility, moderate usage
7	984	5.19 (3.0) 3.43 (1.66)	1.84 (0.79) 1.25 (0.26)	47.99 (2.77) 46.16 (3.22)	6.05 (0.33) 5.83 (0.39)	3.64 (2.04)	Moderate mobility, high usage
8	882	5.23 (3.98) 2.41 (1.12)	9.43 (8.14) 1.15 (0.17)	37.62 (15.36) 17.41 (13.19)	5.15 (1.48) 2.76 (1.39)	1.68 (0.66)	Moderate mobility, train commuters
9	1,459	12.05 (6.44) 6.85 (3.46)	3.06 (1.49) 1.43 (0.28)	22.21 (7.26) 19.67 (6.88)	3.51 (0.78) 3.34 (0.81)	3.23 (1.80)	High mobility, moderate to low usage
10	1,299	12.14 (7.83) 6.83 (4.09)	4.17 (2.51) 1.55 (0.39)	42.91 (5.63) 39.50 (5.88)	5.51 (0.62) 5.13 (0.66)	3.18 (1.33)	High mobility, high usage
11	757	18.85 (19.7) 9.41 (10.57)	6.87 (6.02) 1.80 (1.31)	30.92 (15.08) 21.34 (16.80)	4.54 (1.47) 3.33 (2.07)	12.45 (21.83)	High mobility, moderate usage, high connection rate

Table 2.2: Results from the first-stage clustering, which results in 11 clusters. Each cluster represents a parameterization for a different pattern of mobility. σ is the standard deviation from the average (mean) value.

radios in cluster 4 used the least, and the radios in cluster 6 used the most. However, the radios in cluster 4 also move around the most per day of usage, with more location changes per day. Cluster 8 consists of radios that on any given day, visit approximately twice as many (not necessarily distinct) locations as they see different locations in the entire trace. This behavior corresponds to users who commute by train and use their laptops while on the train, thus visiting most of their locations twice in one day, once on the way to work and once on the way back. In fact, the users in this cluster consist not just of train commuters, but also other public transit commuters (subway, bus, ferry,

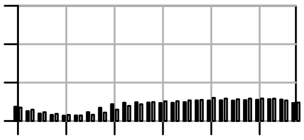
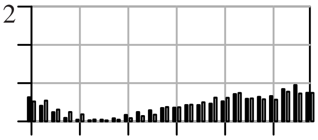
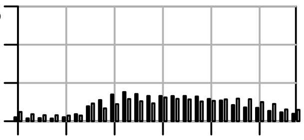
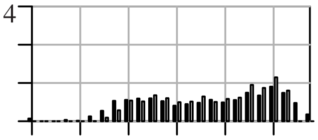
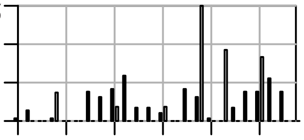
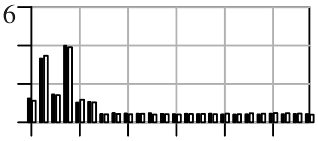
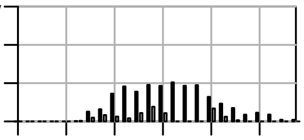
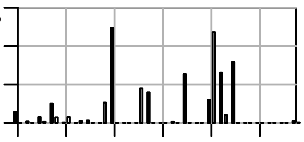
	Representative Histogram	2	3	4	5	6	7	8	9	10	11
1		42.1% 1,939	26.6% 1,267			12.5% 200	20.4% 201	33.9% 299		12.5% 163	28.1% 213
2						3.9% 62	15.3% 151			5.1% 66	
3		21.4% 984	39.2% 1,865	33.1% 370	46.8% 666	38.1% 611	35.9% 353	49.3% 435	19.7% 287	44.3% 575	37.0% 280
4		18.9% 873	8.7% 415	10.4% 116	28.6% 407	42.3% 679	27.0% 266	13.5% 119	71.7% 1,046	36.9% 479	28.5% 216
5		0.2% 8	1.4% 69	1.9% 21	1.9% 27			0.2% 2	0.2% 3	0.1% 1	0.7% 5
6				0.4% 4		0.1% 2	0.9% 9	0.6% 5	0.2% 3	0.8% 11	0.5% 4
7		15.2% 705	23.4% 1,114	48.6% 543	20.1% 286	2.7% 43		1.9% 17	7.9% 115	0.1% 1	5.2% 39
8		2.2% 102	0.7% 35	6.1% 63	2.6% 36	0.4% 6	0.4% 4	0.6% 5	0.3% 5	0.2% 3	

Table 2.3: Results from the second-stage clustering. Each row corresponds to a time pattern, which is shown with a representative histogram. The x-axis of the histogram is the hour of the day, and the y-axis is the percentage of events occurring during that hour. The black bars correspond to weekday usage, and the white bars correspond to weekend usage. Eight time patterns emerge, and the distribution of these patterns differs amongst the mobility patterns resulting from the first-stage clustering. Note that each first-stage cluster corresponds to one column of the table.

etc.). Clusters 9 through 11 correspond to radios that move around much more, differing mainly in how often the radios are used and how many connections are formed per day.

Table 2.3 shows the eight prevalent timing patterns resulting from the second-stage clustering:

1. a smooth curve similar to Figure 2.2, with usage balanced across weekends and weekdays.

2. a curve similar to the previous one, but the times are shifted so that the time of least usage is later (around 5 to 6am rather than 4am), perhaps consisting of “techies” who are used to working later hours.

3. a distribution in which the usage later in the day primarily occurs during the weekend and the usage during the day primarily occurs on weekdays.

4. a distribution in which there is almost no activity all night, and the peaks are a little more obvious. In fact, slight peaks around 9 to 10am and 6 to 8pm can be seen in the weekday and weekend usage.

5. a distribution with very obvious peaks, usually one to two big weekend peaks and two to three smoother weekday peaks. The histogram pictured is merely representative, since other time usage distributions in this cluster have the peaks at different times.

6. a distribution we call system administration hours, with usage throughout the day, but higher usage late at night.

7. a relatively smooth weekday usage curve, predominantly during the day, with almost no weekend usage.

8. and finally, a very sporadic distribution, with one to two peaks during both the weekday and weekends. Similar to (6), these peaks can be shifted in time.

Table 2.3 shows not all users within a first-stage cluster execute their particular mobility pattern at the same time. Instead, users can be further divided into time patterns. For example, Cluster 2 consists of 4,608 stationary radios. This set of radios can further be broken down into users who balance their radio usage across weekends and weekdays (time pattern 1), and users who almost never use their radios at night (time pattern 4, and to a lesser extent, time pattern 3), for example. Also note that not all time patterns occur in every mobility pattern and that the percentages of different time patterns occurring in different mobility patterns vary.

As a result of these clusterings, not only have we found user mobility patterns, but we also found user time patterns, finding that different users may have the same mobility pattern, but execute these patterns at different times. These patterns contradict the common simplification of a single mobility and time pattern for all users, and can therefore be used to create more realistic mobility models.

2.4 Tracing

One lesson we learned that is specific to this analysis concerns the difficulty in analyzing a trace that was not gathered for the purpose of our analysis. The trace gathered by Metricom for their own purposes is missing data we would have found useful. Several unanswered questions and ambiguities are a direct result of not having control over what data was recorded and where the data was recorded.

First, one question we wanted to answer was how long a radio stays active, or in other words, how long connections last. However, connection delimiters are not included in the trace. Therefore, connection duration results which might be directly calculated from a tcpdump trace using TCP SYN and FIN markers, for example, had to be inferred here. In fact, we tried several different methods to infer connection duration results, and concluded that while we cannot infer the connection duration accurately enough, we can infer that the radio has a linear backoff registration scheme.

Figure 2.16 shows how often a given interval between successive registrations by a single radio occurs. The histogram shows exponentially decreasing peaks every 600 seconds, or 10 minutes. If the radio is not reset or if the radio does not change its primary poletop, the radio first registers 10 minutes after the initial registration, then 20 minutes, then 30, and so on. Rather than the typical exponential backoff found in networking protocols, a linear backoff is used.

Second, we wanted to investigate patterns of communication between radios. However, the trace has been anonymized to protect Metricom customers' privacy. As a result, we have no information about the identity of the other end of a radio's connection, or mappings from radio to user, and so we can learn nothing about patterns of communications between radios.

Third, since we have a log from the nameserver only, we could not differentiate problems at the nameserver from problems in the network. For example, the trace includes three periods during which no radio registrations were logged: February 16, 6am through 1pm; February 16, 5pm through February 17 4am; and February 17, 4pm through February 18, noon. We do not know if these holes in the trace were due to a nameserver malfunction or a series of network outages. Metricom has since suggested that this outage is likely a failure in the logging mechanism rather than

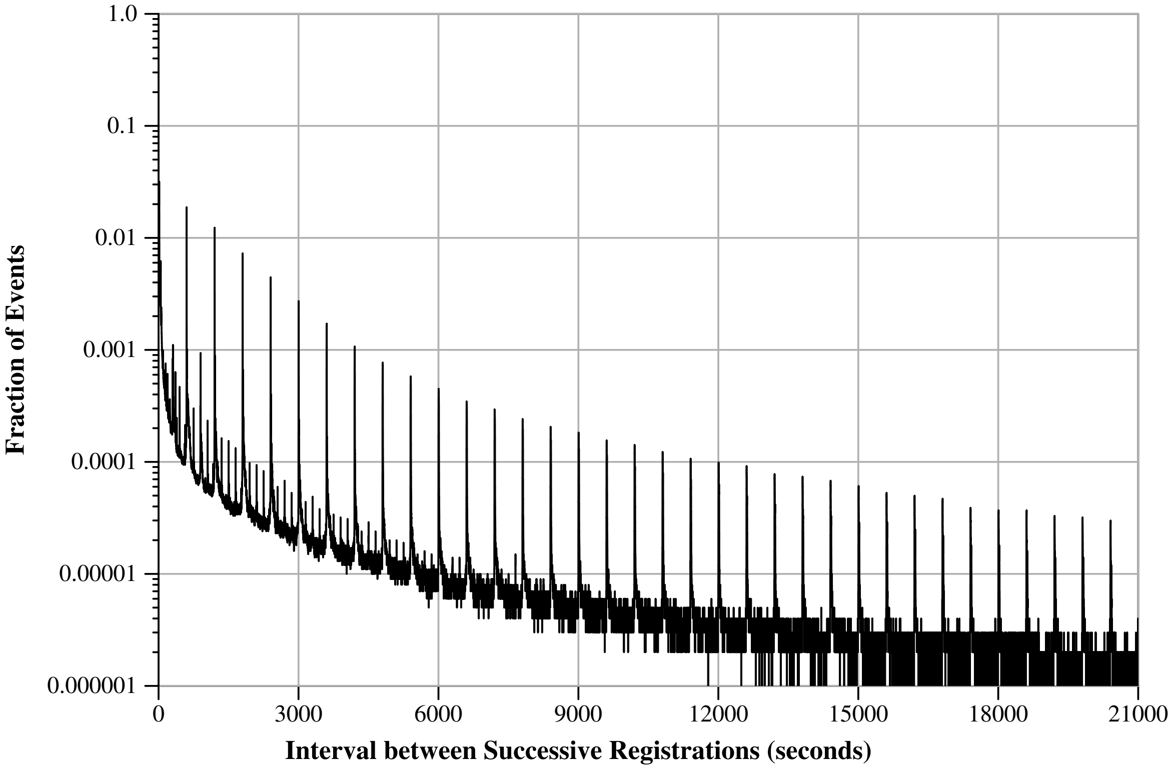


Figure 2.16: Histogram of the number of times a time interval (between successive registrations by a radio) occurs. For example, an interval of length 600 seconds (10 minutes) occurs about 0.025% of the time, or about 150,000 times. Note the logarithmic scale for the y-axis, and that the tail of the graph, which extends to 4,444,118 seconds (74,068.6 minutes, 1234.5 hours, or 51.4 days) has been cut off to show the details.

any network malfunction that would affect users [52].

Another example of such an ambiguity is a potential network reconfiguration on March 19. On that day, several poletops, previously unseen in the trace, first appear near a poletop at Metricom headquarters and then within the period of an hour, 5pm to 6pm on March 19, all of these poletops reappear at a new location in the Midwest. We do not know whether these poletops malfunctioned, or whether the network was in fact reconfigured. We suspect the poletop radios were being tested before installation in the Midwest.

Finally, the timestamps are the last problem we encountered in analyzing this trace. They are taken at the nameserver, which means that while clock skew is not a problem, network traversal time between the radio and the nameserver is not taken into account. Although we cannot determine the upper bound for this error, we can approximate the average error, since Metricom states that each packet will make no more than three wireless hops before reaching a wired access point [40]. Traversing three wireless hops takes on average between 200ms and 400ms, which is less than the accuracy of the timestamp itself, making the clock skew a non-issue. These unanswered questions and ambiguities are all a direct result of not having the necessary data and not controlling the data collection.

2.5 Future Work

The main body of future work needed is the comparison of this trace to other mobile metropolitan-area network traces. It is possible, even probable, that at least some of the results are a product of the Metricom network and its features, such as its latency, network infrastructure, and trace information or lack thereof. It would be especially interesting to compare the analysis of this trace to a trace of a wireless network where the majority of the devices are much smaller, such as the Personal Digital Assistant (PDA) class of devices. In a network with smaller devices, several of the results may differ, including how mobile the users are and when peak network usage occurs.

2.6 Chapter Summary

Although the information in the traces limits the obtainable results and the results themselves are particular to the Metricom network, with its metropolitan-area coverage and high latency (compared to a local-area network), our analysis is a start on understanding how people take advantage of a mobile environment.

We verify the expectation that most movement is local rather than long distance and that more users are active during the day and on weekdays rather than weekends. However, rather than verifying the expectation that most users are mobile, we instead find many stationary users, and only a few highly mobile users. We also find that more users move day-to-day rather than during any given day. Perhaps the most interesting result is that we can indeed find user mobility and time patterns, and that different users with the same mobility pattern may not have the same time pattern. Given that many simulations of radio mobility assume a static time between radio movement or some other simple model of user movement, our results present a more realistic basis for a model of movement. We hope that these results can be used to guide simulations to yield projected network performance results based on observed movement.

The next chapter presents an analysis of a local-area wireless network, which differs significantly from the Metricom network. Chapter 5 and Chapter 6 will discuss how clustering and visualization are especially useful for these type of analyses.

Chapter 3

Analyzing a Local-area Wireless Network

In the previous chapter, we presented an analysis of a metropolitan-area wireless network, exploring how users move around in the network and how much hand-off activity poletop transceivers experience in the network.

In this chapter, we consider a very different wireless environment. We analyze a 12-week trace of a local-area wireless network installed throughout the Computer Science Building of Stanford University. Our goal is to answer questions such as how much users take advantage of mobility, how often peak throughput rates occur, what causes the peaks, and what application mix is used. The scale of this network is much smaller than that of the previous study, and its characteristics in terms of delay and bandwidth are much more favorable. We thus find that not all questions asked by the previous analysis make sense in this context; for instance, we do not analyze user mobility for frequently-used paths through the network. In contrast to the previous study, however, we explore information about network data traffic and can ask questions about application mixes, symmetry of outgoing and incoming traffic, and traffic throughput.

Some of the more surprising results in this analysis are the unexpectedly high number of users using chat software, indicating that more users than expected are taking advantage of their laptops for synchronous communication with other users, and the discovery that users fall into distinct location-based sub-communities as opposed to being a single, fairly homogeneous community, each with its own behavior regarding movement and periods of activity. Also unexpected is the result regarding the symmetry of traffic: while incoming traffic dominates outgoing traffic overall (34 billion bytes compared to 12 billion bytes), the opposite tends to be true during periods of peak throughput, implying that significant fixed asymmetry in network capacity could be undesirable for the users. This result is especially relevant given the current research on asymmetric networks, such as satellite communications and cable modems.

We hope that the results presented here will help researchers and developers determine how users take advantage of a local-area wireless network, helping focus their efforts on topics that will achieve the most improvement in user experience. While these results are only necessarily valid for this particular local-area wireless network and user community, we believe that similar environments may exhibit similar behavior and trends.

In this chapter, we first present background information about the data we collected and then present the results of our analysis. We divide the analysis into three sections: overall user behavior, overall network traffic characteristics, and user traffic characteristics. We conclude with some possible directions for future work.

3.1 Background

In this section we describe the network analyzed and our tracing methodology. In the Gates Computer Science Building at Stanford University, administrators have made a "public" subnet available for any user affiliated with the university [2]. Users desiring network access via this subnet must authenticate themselves to use their dynamically assigned IP address [16] to access the rest of the departmental and university networks and the Internet.

This subnet, as shown in Figure 3.1, is accessible both from a wireless network and from Ethernet ports in public places in the building, such as conference rooms, lounges, the library, and labs. The wireless network is a WaveLAN network with WavePoint II access points acting as bridges between the wireless and wired networks [62]. The access points each have two slots for wireless network interfaces; both slots are filled, one with older 2 Mbps cards to support the few users who have not updated their hardware yet, and the other with WaveLAN IEEE802.11- compatible 10 Mbps cards.

To help explain the results presented in the next sections, we briefly describe the Gates Computer Science building and its user community. The building is L-shaped (the longer edge is called the a-wing, and the shorter the b-wing). It has four main floors with offices and labs, a basement with classrooms and labs, and a fifth floor with a lounge and a few offices. Each of the main floors has two access points, one for each wing. Additionally, the first floor has an access point for a large conference room, and the library, which spans both the second and third floors, also has an access

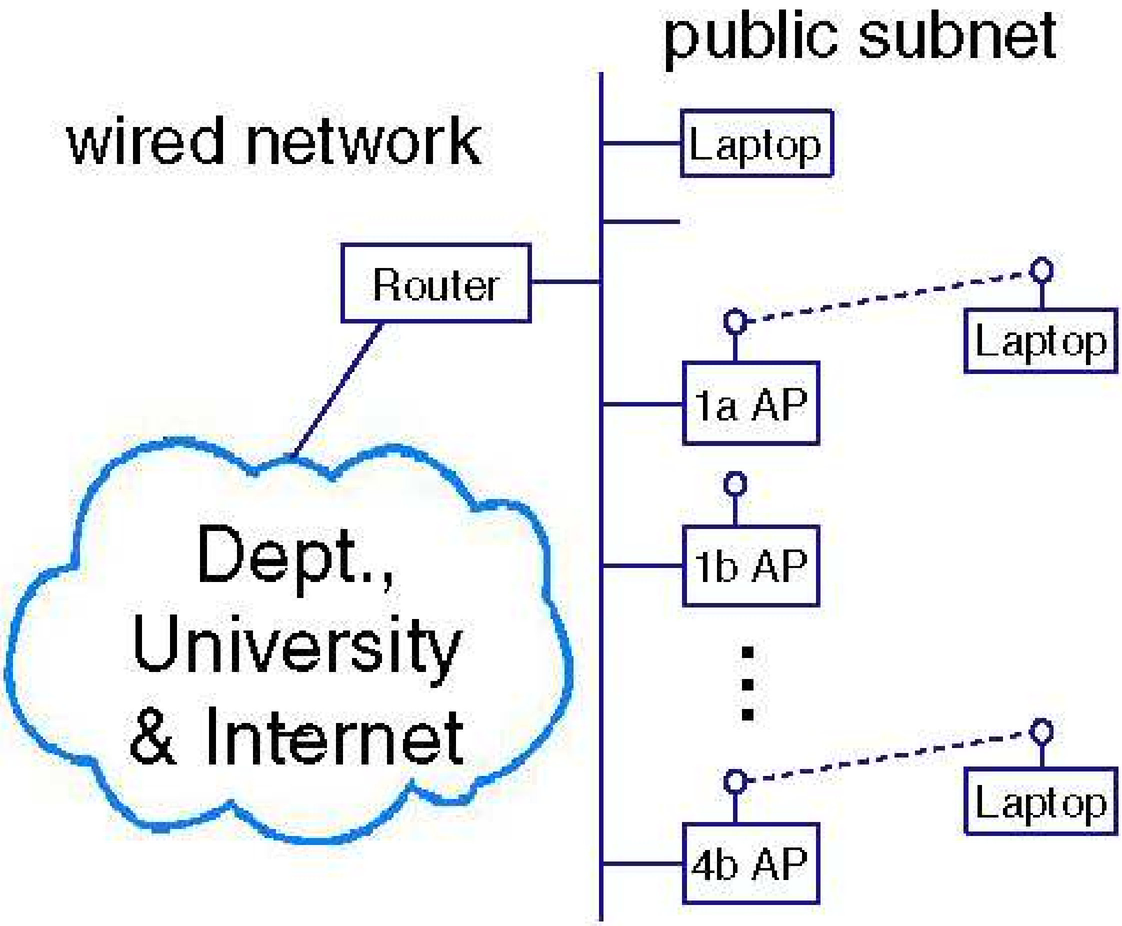


Figure 3.1: The public subnet and its connectivity to the rest of the departmental and university networks and the Internet. An AP is an access point for wireless connectivity

point. The basement has two access points, one near the classrooms and one for the Interactive Room, a special research project in the department [20]. The smaller fifth floor only has one access point. The wireless user community consists of 74 users who can be roughly divided into four groups:

- 35 first year PhD students, who were each given a laptop with a WaveLAN card upon arrival (which corresponds to the beginning of the trace). Their offices are primarily in the 2b wing.
- 22 graphics students and staff, the majority of whom received laptops with WaveLAN cards a week into the tracing period. Their offices are primarily in the 3b wing.

- Three robots, used by the robotics lab for research. The robots do not have to authenticate themselves to reach the outside network. While the robots are somewhat mobile, they stay in the 1a wing. Although these WaveLAN cards are intended to be used by the robots, students in the robotics lab also use the network cards for session connections and web-surfing.

- 14 other users (students, staff, and faculty) scattered throughout the building. In addition to these 74 users, there were also four users who authenticated themselves but only connected to wired ports on the public subnet rather than the wireless network. We do not consider these users in the rest of this analysis of the wireless network.

We obtained permission to collect these traces from the Department Chair and informed all network users that this tracing was taking place. We additionally informed users we would record packet header information only (not the contents) and that we would anonymize the data. Knowledge of the tracing may have perturbed user behavior, but we have no way of quantifying the effect.

Because all of the wireless users are on a single subnet (which promotes roaming without the need for Mobile IP or other such support), we gathered traces on the router shown in Figure 3.1 that connects the public subnet to the rest of the departmental wired network. The router is a 90 MHz Pentium running RedHat Linux with two 10 Mbps network interfaces. One interface connects to the public subnet, and the other connects to the departmental network.

To gather all of the information we wanted, we collected three separate types of traces during a 12-week period encompassing the 1999 Fall quarter (from Monday, September 20 through Sunday, December 12). The first trace we gathered is a tcpdump trace of the link- and network-level headers of all packets that went through the router [28]. We use this information in conjunction with the second two traces, described next.

The second trace is an SNMP trace [10]. Approximately every two minutes, the router queries, via Ethernet, all twelve access points for the MAC addresses of the hosts currently using that access point as a bridge to the wired network. Once we know which access point a MAC address uses for network access, we know the approximate location (floor and wing) of the device with that MAC address. We pair these MAC addresses with the link-level addresses saved in the packet headers to determine the approximate locations of the hosts in the tcpdump trace.

The overhead from the SNMP tracing is low: 530 packets or 50 KBytes is the average overhead from querying all twelve access points every two minutes. The overhead for querying an individual access point is 3.2 KBytes if no MAC addresses are using that access point; otherwise, the base overhead is 14.5 KBytes for one user at an access point, plus 1 KByte for every additional user.

The last trace is the authentication log, which keeps track of which users request authentication to use the network. Each request has both the user's login name as well as the MAC address from which the user makes the request. We pair these MAC addresses with the link-level addresses saved in the tcpdump trace to determine which user sends out each packet.

We use the common timestamp and MAC address information to combine these three traces into a single trace with a total of 78,739,933 packets attributable to the 74 wireless users (an additional 37,893,656 packets are attributable to the SNMP queries and 1,551,167 packets are attributable to the four wired users). Thus, for every packet sent over the course of this twelve-week period we record:

- a timestamp,
- the user's identity,
- the user's location (current access point),
- the application, if the port is recognized, otherwise, the source and destination ports,
- the remote host the user connects to, and
- the size of the packet.

Note that because we gather information about packets on the router, we do not record traffic between laptops on the subnet. Also note that because we do not record any signal strength information, and since the access points generally cover a whole wing of a floor, we cannot necessarily detect movement within a wing but only movement between access points.

3.2 Overall User Behavior

In this section we consider the network-related behavior of users, focusing on their activity and mobility. Specifically, we ask the following questions:

1. When and how often do people use the network?
2. How many users are active at a time?
3. How much do users move?

The answers to these questions help researchers understand whether and how users actually take advantage of a mobile environment. Also, by understanding user behavior, network planners can better plan and extend network infrastructure. In general we find that most users do not move much within the building, but a few users are highly mobile, moving up to seven times within an hour. We also find that users fall into location-based sub-communities, each with its own movement and activity characteristics. For example, the sub-community in the 2b wing tends to move around a fair amount and use the network sporadically, whereas the 4b wing sub-community steadily uses the network but does not move around very much.

3.2.1 Active Users

We first look at average user activity by time of day; knowing when users are active can help network planners anticipate and handle peak usage periods and help researchers build models of mobile users. We consider a user to be active during an hour in the trace if that user sends or receives a packet sometime during that hour. Figure 3.2 shows that on weekdays more people use the network in the afternoon than at any other time (on average there are 11 to 12 users in the mid-afternoon, with a maximum of 28 users between 2 and 4 in the afternoon). Also, the steady number of active users throughout the night and weekend show that four to five users, on average, leave their laptops turned on in their offices rather than take them home, usually to act as mail clients.

Figure 3.3 is a graph analogous to Figure 3.2, presenting the number of active users per day in the trace rather than per hour of the day. This graph shows a weekly pattern with more active users during the week than on weekends. There are also some notable trends through the course of the trace: the network supports the most users at the beginning of the trace (up to 34 on the second Monday of the trace), when many users first received their laptops, with a lull in the middle of the quarter corresponding to midterms and comprehensive exams, followed by an upswing corresponding to final project due dates, before a drop during finals week and an exodus for winter vacations. We also believe that the number of users falls off as new Ph.D. students received their permanent office assignments elsewhere in the building. It seems that many users still prefer stationary desktop machines over laptops when both are available to them.

Figure 3.4 presents overall activity from a user point of view: the total number of days users are active during the traced period. While some users rarely connect their laptops to the network (17 users do so on 5 days or fewer), others connect their laptops frequently (14 users are active at least 37 days during the traced period). This graph shows the partitioning of users into different behavior types, from barely active to highly active, contrary to the common expectation that all users are

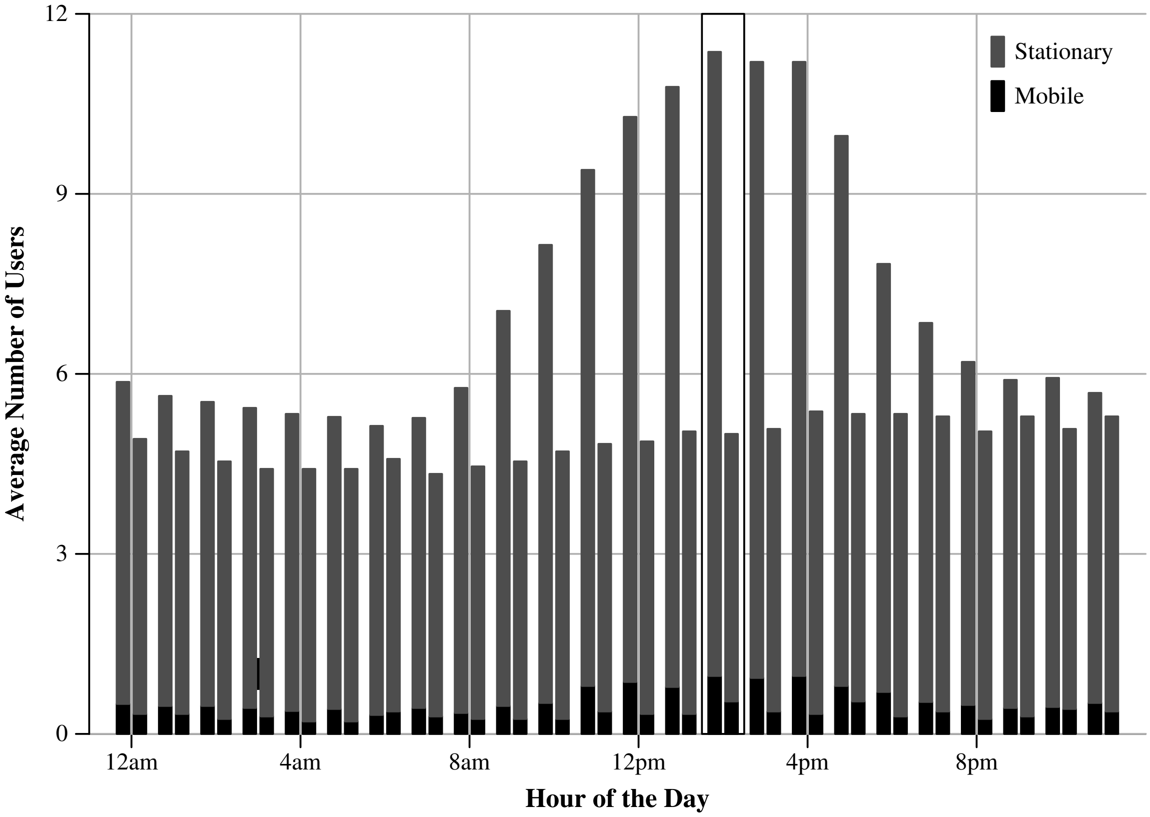


Figure 3.2: The average number of active users of the mobile network and the number of locations they visit each hour of the day. Each hour has two bars, the left one for weekdays and the right one for weekends. The darkness of the bar indicates how many locations the users visit over the course of the hour. For example, the highlighted bars show that at 2pm, on average 11.4 users use the network on weekdays (0.8 of which, on average, visit at least 2 locations over the course of the hour), and on average 5.1 users use the network on weekends (0.5 of which, on average, visit at least 2 locations over the course of the hour).

fairly similar.

3.2.2 User Mobility

We next explore user mobility. Figure 3.2 also shows us information about average user mobility by time of day. Most users are stationary, meaning they do not move from one access point to another. Only a few users (1.3 on average) move between access points during any given hour. However, some users are highly mobile with a maximum of seven location changes for a user within an hour. Note that we are underestimating user mobility since we can only detect user movement is a user

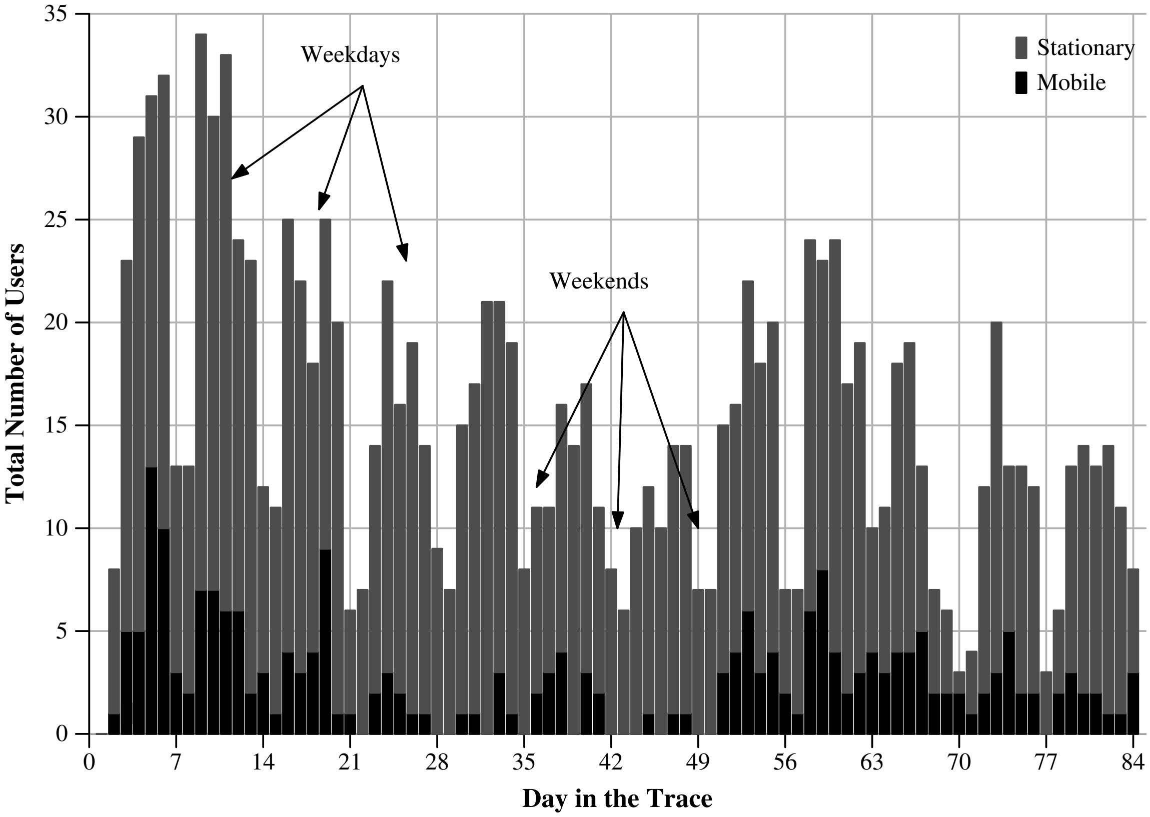


Figure 3.3: The total number of users of the mobile network per day and the number of locations they visit during each day of the trace. The 1st day is Monday, the 2nd day is Tuesday, etc. The darkness of the bar indicates whether the users are stationary or mobile during the course of that day.

changes which access point he is using. If a user just moves within a wing and does not change access point, we cannot detect that movement.

Figure 3.3 shows that the number of highly mobile users follows the same overall pattern as active users over the course of the trace, with up to 13 mobile users in a day at the beginning of the trace and as few as one or two mobile users per day towards the end of the trace. As in Figure 3.2, however, most users are stationary on any given day with only a few (3.2 on average) moving around.

Looking at total mobility across the trace, Figure 3.5 shows that while 37 users are stationary throughout the entire period, a few users exploit the mobile characteristic of the network: 13 users visit at least five distinct access points during the course of the trace and one user visits all twelve access points.

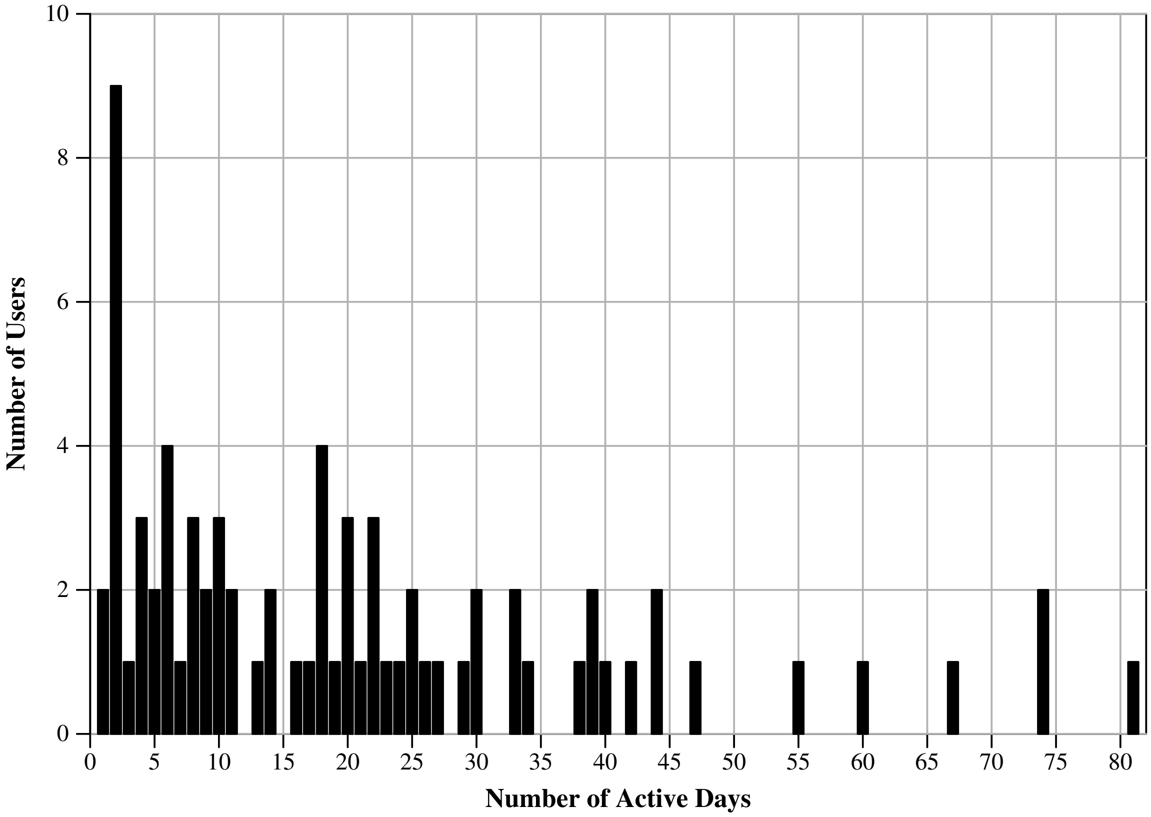


Figure 3.4: Number of active days for mobile users over the course of the entire trace.

These results are a direct contrast to the customary expectation that all users are mobile, showing instead a continuum of users between stationary and highly mobile. Thus, several different models of user behavior are needed, rather than a single model.

3.2.3 User Sub-communities

We now turn to location-based user behavior by associating user activity and mobility with access points. Figure 3.6 shows that the access points in the 2b and 3b wings handle the most users (up to 12 or 10 users, respectively, within a five-minute period), which is not surprising given the large number of mobile users with offices in those two wings. Figure 3.7 shows how many handoffs access points have to handle. Contrasting Figure 3.6 with Figure 3.7 shows that the number of users is not necessarily correlated with movement. The users on the 3b wing rarely move, while the users on the 2b wing move around more often. The few users in the 1b wing move even more frequently.

Table 3.1 summarizes user activity by access point location. The basement and the conference

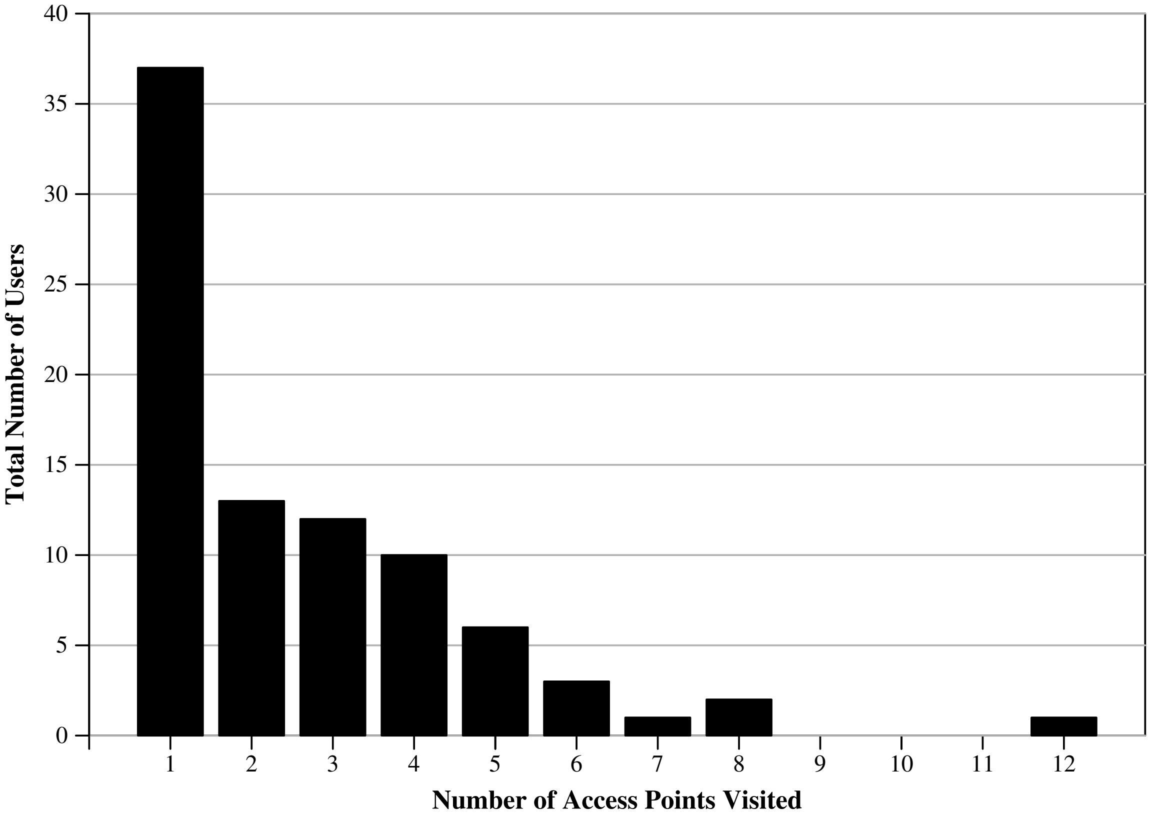


Figure 3.5: Number of users visiting some number of access points over the course of the entire trace.

room in 104 are primarily used only when meetings occur, while the 5th floor lounge is used when people take a break in the late afternoon. The 4th floor users are steady users who rarely move, the 3rd floor users connect to the network more sporadically, and the 2nd floor users are also sporadic but more mobile. These results imply that while each access point covers approximately the same amount of space, the load on each access point depends on the behavior of the community it serves. Thus, while we originally expected the users to behave as a single homogeneous community comprised primarily of computer science graduate students, we instead find that the users are partitioned into sub-communities, indicating that future network analyses and planners need to consider not only the overall user behavior but also the differing behavior by sub-community.

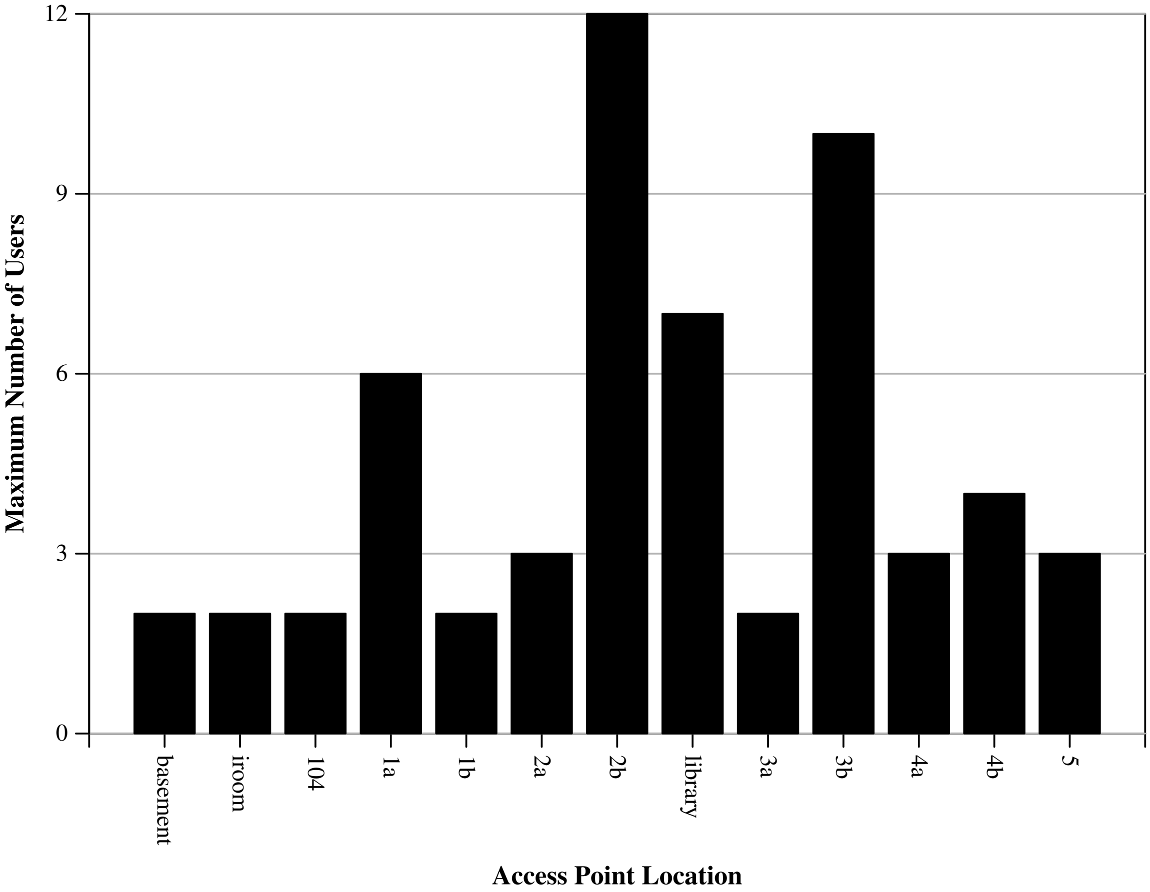


Figure 3.6: Maximum number of users each access point handles within a five-minute period. iroom = Interactive Room.

3.2.4 Access Point Handoffs

One side effect of user mobility is the need for access points to perform handoffs. We thus take a closer look at how many handoffs access points handle. A **handoff** is defined as a user appearing at one access point and then moving to a different access point within a given period of time. Figure 3.7 shows that handoffs are not a major burden on access points: an access point handles at most five handoffs within a five-minute period, or ten within a 15-minute period. Note that 95% of all user location changes occur within 15 minutes. This result indicates that the research in optimizing handoffs may not be a high priority for this network environment.

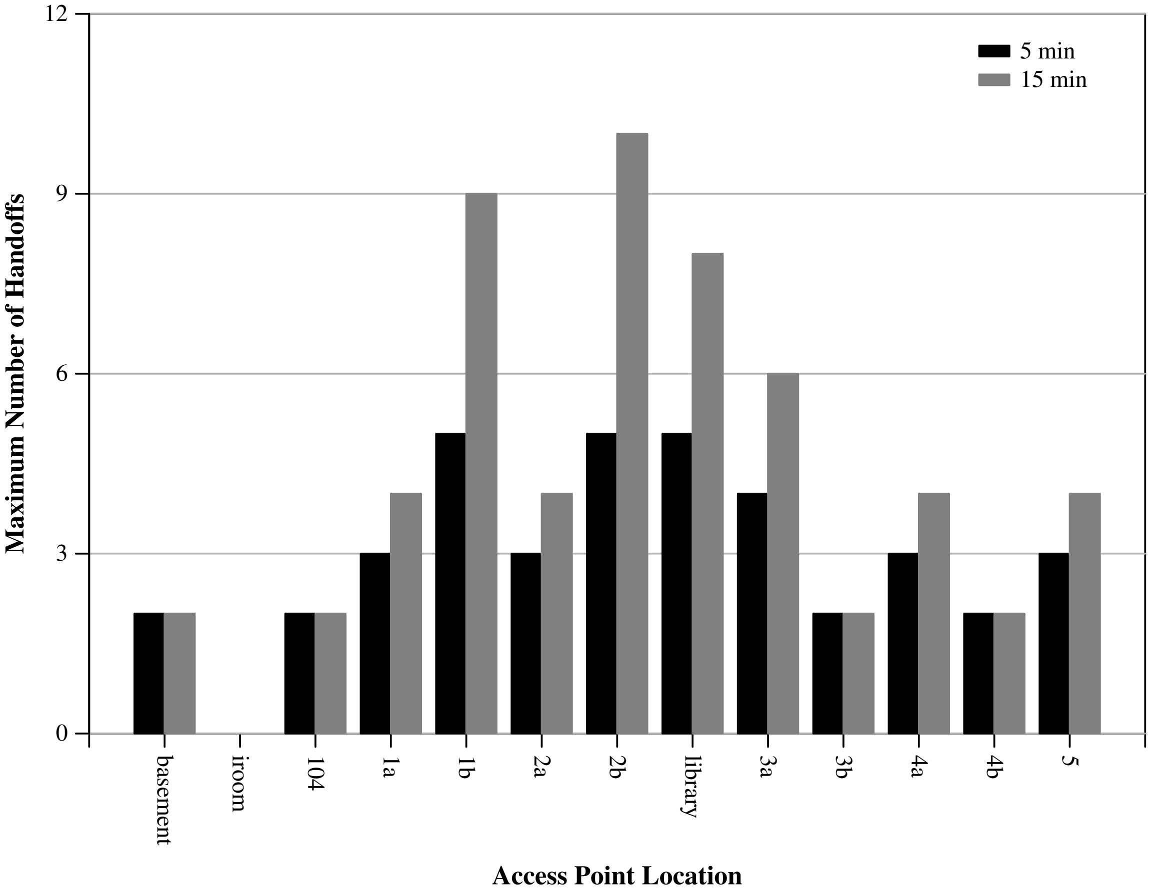


Figure 3.7: Maximum number of handoffs each access point handles within a 5-minute or 15-minute period. iroom = Interactive Room.

3.3 Overall Network Traffic Characteristics

In this section we consider overall characteristics of the network, such as throughput, peak throughput, and incoming and outgoing traffic symmetry. Specifically, we ask the following questions:

1. What is the throughput through the router? Through the access points?
2. What is the peak throughput?
3. How often is peak throughput reached?
4. What causes the peaks? Several users or only one or two? Multiple applications or only a few?

Access Point	**Description**
basement	occasional spikes corresponding to meetings
iroom	big peak in weeks 8, 9 (project deadline)
104	occasional spikes corresponding to meetings
1a	heavy usage weeks 1-3, occasional afterwards
1b	occasional usage corresponding to network testing
2a	occasional usage, small peak towards end
2b	closely follows overall pattern in Figure 3.2
library	meeting in weeks 1-3, slight peak weeks 6-7
3a	lower usage, follows overall pattern in Figure 3.2
3b	follows overall pattern in Figure 3.2
4a	1-2 users regularly
4b	1-3 users constantly
5	1-2 users in late afternoons, Monday - Friday

Table 3.1: Brief description of the activity at each access point throughout the course of the trace.

5. How symmetric is the traffic? (How similar is incoming traffic to outgoing traffic?)
6. How much traffic is attributable to small versus large packets?

The answers to these questions help determine how wireless hardware and software should be optimized to handle the amounts of traffic wireless networks generate. Such optimizations may include using asymmetric links or optimizing for a few large packets versus many smaller packets.

While we believe that latency is critical to users, the latency of the WaveLAN network is equivalent to wired ethernet, and we thus choose not to analyze our trace for this metric. The latency users see on the Gates network is attributable almost entirely to the outside network, especially the Internet [47].

In general, we find that router throughput reaches peaks of 5.6Mbps and that peak throughput is caused 80% of the time by a single user and application, usually a large file transfer. On average, the incoming traffic is heavier than outgoing traffic, but the periods of peak throughput are actually skewed more towards outgoing bytes. From this result, we conclude that significant fixed asymmetry in network capacity would not be desirable for the users of the wireless subnet. We also find that given the wireless network's application mix, low per-packet processing overhead to handle many small packets is just as important as high overall attainable byte throughput.

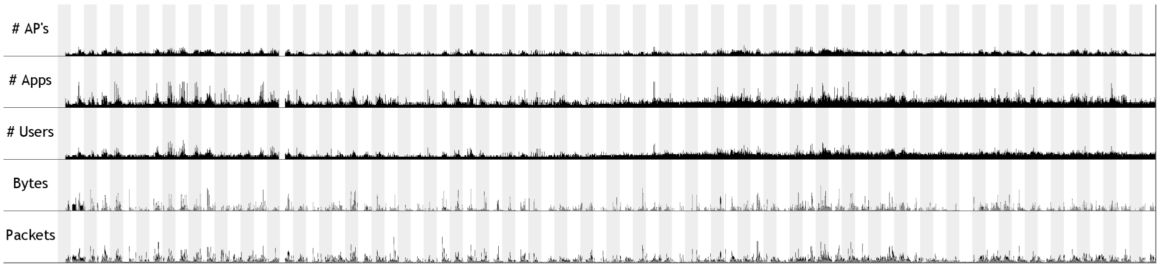


Figure 3.8: Overview of throughput over the entire trace, both in bytes (maximum of 5.6 Mbps) and packets (maximum of 1,376 packets), as well as the number of access points (AP's, maximum of 9), applications (maximum of 56), and users (maximum of 17) responsible for generating the traffic.

3.3.1 Network Throughput

Figure 3.8 gives an overview of throughput over the traced period, as well as how many access points, users, and applications are responsible for generating the traffic. Throughput through the router is typically around one to three Mbps. Usually, the throughput as a whole increases as the number of users increases. The throughput through the router reaches peaks of 5.6 Mbps. Table 3.2 shows the maximum throughput attained through the router and each access point. In no case is the peak throughput maintained for more than three seconds, indicating that the network is not overwhelmed, but rather that traffic is heavy enough to hit the peak rate on occasion. In other words, upgrading the wireless network to 100Mbps is not necessary in a community of this size.

Because the highest peaks occur at the access points with the most users, we expected that multiple users would cause most peaks. Instead, for the majority of peaks, the maximum throughput is achieved by a single user and application, rather than distributed across several users or even one user running several applications. Specifically, of the 1,492 peaks of magnitude 3.6 Mbps or greater, 80% of those peaks have 94% of their traffic generated by a single user and application, and 97% of their traffic generated by a single user. The application responsible for 53% of those peaks is ftp, with web traffic responsible for 15%, and the remainder caused by applications such as X, session traffic (e.g., ssh and telnet), and mail downloads (e.g., eudora, imap, and pop). These numbers indicate that optimizing access points for these peak situations might benefit users. This result also implies that synchronizing multiple users to create a peak usage situation when simulating networks is not necessary.

From this data, we also observe that evidence of user sub-communities with different behaviors carries over to traffic throughput characteristics. While the wings with the most users (2b and 3b) also have the highest peak throughput, the users on the 3b wing attain that throughput more often

Location	Max Packets (pps)	Max Bits (Mbps)	Percentage of Peaks > 3 Mbps
router	1,376	5.6	100%
ethernet	1,096	5.1	5.8%
basement	530	3.2	0.1%
iroom	446	3.6	3.3%
1a	521	3.4	0.6%
1b	455	3.6	2.0%
104	429	3.4	0.7%
2a	783	3.1	0.02%
2b	824	4.5	8.9%
library	745	4.5	2.4%
3a	737	3.6	6.3%
3b	883	4.6	69.7%
4a	804	1.7	0%
4b	675	3.9	0.07%
5	703	3.4	0.1%

Table 3.2: Maximum throughput attained through the router, the public Ethernet ports, and each access point.

(69% of peaks of magnitude greater than 3 Mbps are attributable to the 3b wing), indicating that although these users may not be very mobile, their traffic causes more load on the network.

3.3.2 Network Symmetry

Another network characteristic we investigate is the symmetry of incoming and outgoing traffic. One might expect that because the most common application is web-surfing (see Section 5) that incoming packets and bytes would overwhelm outgoing packets and bytes. Instead, we find that while the total incoming traffic (34 billion bytes and 62 million packets) is larger than the total outgoing traffic (12 billion bytes and 56 million packets), the peaks are actually skewed more towards outgoing traffic. Of the peaks of magnitude greater than 3.6 Mbps, 60% are dominated by outgoing rather than incoming traffic. One possible explanation for this result is that outgoing traffic is logged immediately at the router, whereas incoming traffic must traverse the Internet before reaching the router and the destination host, thus spreading it out somewhat due to variable delays on the

Internet. From this data, we conclude that fixed and significantly asymmetric capacity in wireless networks would be undesirable to users in environments similar to the Gates wireless subnet. This result is especially relevant given the current work into developing asymmetrical networks, such as satellite links and cable modems, and supporting asymmetrical links, such as by modifying routing algorithms to recognize asymmetric links.

3.3.3 Packet versus Byte Throughput

The last overall network characteristic we explore is how packet throughput differs from byte throughput. Figure 3.9 presents a closer look into the distribution of packet sizes in the network, showing that over 70% of packets are smaller than 200 bytes. However, this same number of packets represents only about 30% of all bytes transmitted. Note that fragmented packets are not reassembled for this graph. However, of the 78,738,933 total packets, only 206,895 (0.26%) are fragments and should therefore not impact the distribution much.

We thus conclude that low per-packet processing overhead is just as important to users in this environment as high overall attainable throughput, which is the usual focus of research. Per-packet processing optimizations need to exist not only in the hardware, but also in the operating system and even application software.

We further look at the packet size distribution across several commonly used applications, shown in Figure 3.10, to determine how these applications can be categorized in terms of packet size. For example, http and database applications should be optimized to handle large incoming and small outgoing packets. In contrast, session, chat, mail, and X applications should be optimized to handle many small outgoing and incoming packets. While the optimizations for mail, an application for asynchronous personal communication, may be independent of latency, the optimizations for session, chat, and X applications must not only optimize for the many small packets but also minimize delay in order to facilitate user interactivity.

3.4 User Traffic Characteristics

In this section we consider traffic characteristics from a user perspective. The specific questions we ask in this section are:

1. Which applications are most common?

2. How much does application mix vary by user?

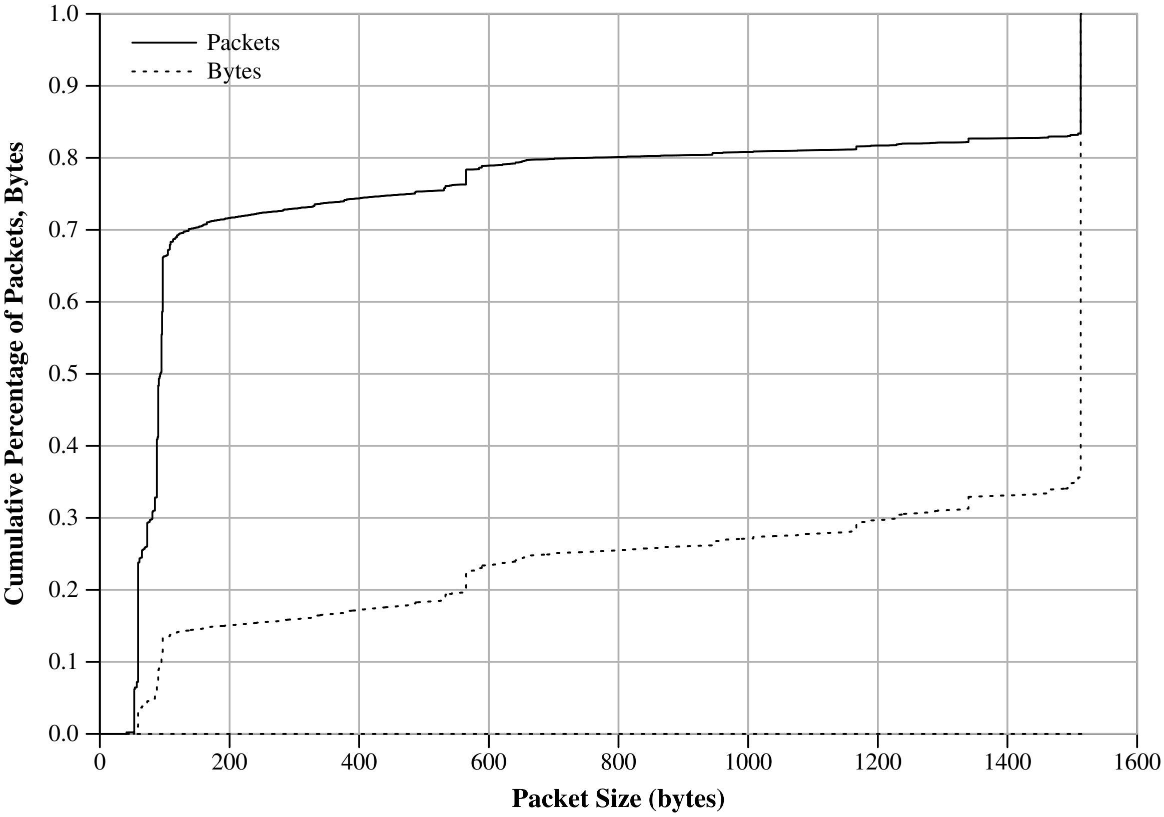


Figure 3.9: Cumulative histogram showing the percentage of packets that are a certain number of bytes long and the percentage of bytes transferred by packets of that length.

3. How many hosts do users connect to?

4. How long are users active?

Answering these questions helps determine which applications and application domains to optimize for mobile usage. Knowing the traffic mix and how it varies by time can also help researchers model user traffic better, which is important when simulation is used to evaluate mobile protocols. Finally, knowing which and how many remote hosts users connect to also helps when modeling network connectivity.

We find that the most popular applications are web-browsing and session applications such as ssh and telnet. These two classes of applications are frequently run together, so some optimization of their interaction might be useful. About half of the wireless subset users frequently execute chat-oriented applications (such as talk, icq, irc, and zephyr), showing that some users exploit the mobile network for synchronous communication with others. We also find that user application

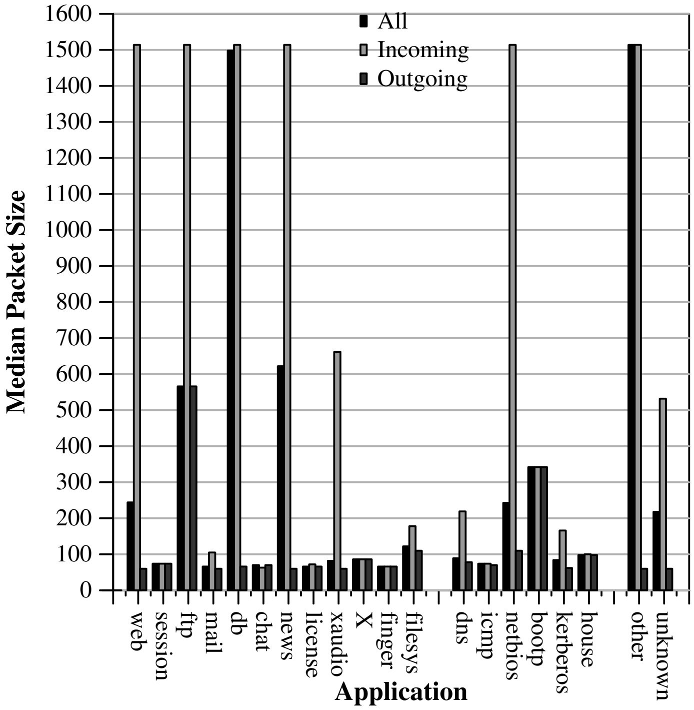


Figure 3.10: Median incoming and outgoing packet sizes for some commonly used applications. Session applications include ssh and telnet; mail includes pop, imap, and Eudora; filesys includes nfs and afs; chat includes talk, icq, zephyr, and irc. House includes housekeeping applications such as ntp.

mixes can be classified into several patterns, such as the terminal pattern, wherein people primarily use their laptops to keep sessions to external machines open, or the web-surfing pattern, wherein most network traffic is web traffic to many different hosts.

3.4.1 Application Popularity and Mixes

Table 3.3 lists the most common classes of applications by number of users and total number of packets and bytes. These statistics indicate which applications are most common, and therefore changing which applications will affect users the most. This table also shows trends in laptop usage, such as the increasing number of chat software users and the decreasing number of terminal (e.g., X and remote filesystems, such as nfs and afs) users. These changing trends can be used to target emerging areas of research, such as chat software, and to predict future usage patterns. We now discuss Table 3.3 in more detail.

Unsurprisingly, basic service applications such as dns, icmp, and house-keeping applications

App Class	Num Users	Num In. Pkts	Num In. Bytes	Num Out. Pkts	Num Out. Bytes
dns	74	0.2	45	0.21	17
icmp	74	0.74	69	0.73	64
netbios	71	7.3	6200	7.2	1200
bootp	56	0.007	2.3	0.007	2.2
kerberos	34	0.004	1.8	0.002	6.8
house	33	0.046	10.8	0.054	6.8
web	73	14.4	15700	11	1100
session	63	6.7	1400	6.8	789
ftp	62	2.8	2600	4	2400
mail	47	0.7	418	0.5	55
db	44	0.005	6.4	0.002	0.2
chat	38	0.03	14.7	0.03	2.2
news	21	0.24	266	0.15	9.6
license	21	0.001	0.5	0.002	0.2
xaudio	21	0.002	1.2	0.001	0.2
X	20	1.5	854	2.2	218
finger	17	0.001	0.2	0.001	0.08
filesys	16	0.07	37	0.06	36
other	33	0.1	141	0.1	51
unknown	68	6	3500	3.7	1000

Table 3.3: The most common applications by user, incoming packets and bytes, and outgoing packets and bytes (all in millions of packets or bytes). Session applications include ssh and telnet; mail includes pop, imap, and Eudora; filesys includes nfs and afs; chat includes talk, icq, zephyr, and irc. House includes housekeeping applications such as ntp.

such as ntp are used by everyone. The high amount of netbios traffic indicates that almost all the users (71) run some version of Windows on their laptops. This reflects the system administrators' choice to install Windows as the default system on laptops.

Of the end-user applications, http, session applications, and file transfer applications are the most popular (with 73, 63 and 62 users, respectively). There are several interesting points of comparison in this data. First, 62 people use some file transfer protocol compared to 16 people who use some remote filesystem such as NFS [55] or AFS [27]. This disparity indicates that many users find it necessary to transfer files to and from their laptops, but only a few users are either willing or find it necessary to use a distributed filesystem, perhaps due to the lack of support in distributed filesystem servers for dynamically assigned addresses.

Also interesting is the unexpectedly small number of people who use their laptops as a terminal compared to the number of people who run applications directly on their laptops. Specifically, only 20 users run X. In comparison, 47 users run some direct mail client (pop, imap, eudora, smtp, etc.); 21 connect to some license server (Matlab, Mentor Graphics, etc.), presumably to run the application directly on their laptops; 21 connect directly to a news server; and 38 users run some sort of chat software (talk, icq, zephyr, irc, etc.). These numbers reveal a tendency to use laptops as stand-alone machines with connectivity, rather than terminals. However, most users do still use session applications such as ssh or telnet, showing that users still need to connect to some other machines. Finally, over half of the users execute chat software, indicating that some users treat their laptops in part as personal synchronous communication devices.

In addition to overall application usage, we also look at eleven characteristic user application mixes, shown in Table 3.4. The main characteristics considered in this categorization are the percentage of traffic in a given period of time that can be attributed to each application, when each application dominates the user's traffic, and the number of hosts to which a user connects. We only use a coarse-grained time characterization: weekday versus weekend and during the day versus late at night. We also confine the categorization to six common applications: web-surfing, session applications, X, mail applications, file transfer, and chat applications.

Figure 3.11 provides more detail for three of the application mixes. The first mix is the home mix, wherein the user is active in the morning uploading information or work from the laptop, at lunch downloading information or work, and in the evening downloading materials before heading home. These users typically connect to only one or two sites which are frequently repeated. The next mix is the web-surfer mix, wherein users contact many different web sites (up to 3,029 distinct web sites for one user). Many of these sites (up to 1,982 for one user) are visited more than once by the same user. The last application mix we focus on is the dabbler mix, in which users run all of the application types at least once.

We derive several conclusions from this application mix characterization. First, while at some point every possible combination of applications is run together, the applications most commonly run together are web and session applications. Second, while http and ssh are the most popular applications across all users, different users do run different mixes of applications, and they do so at different times of day. There is no single application mix that fits all mobile users. Finally, not only

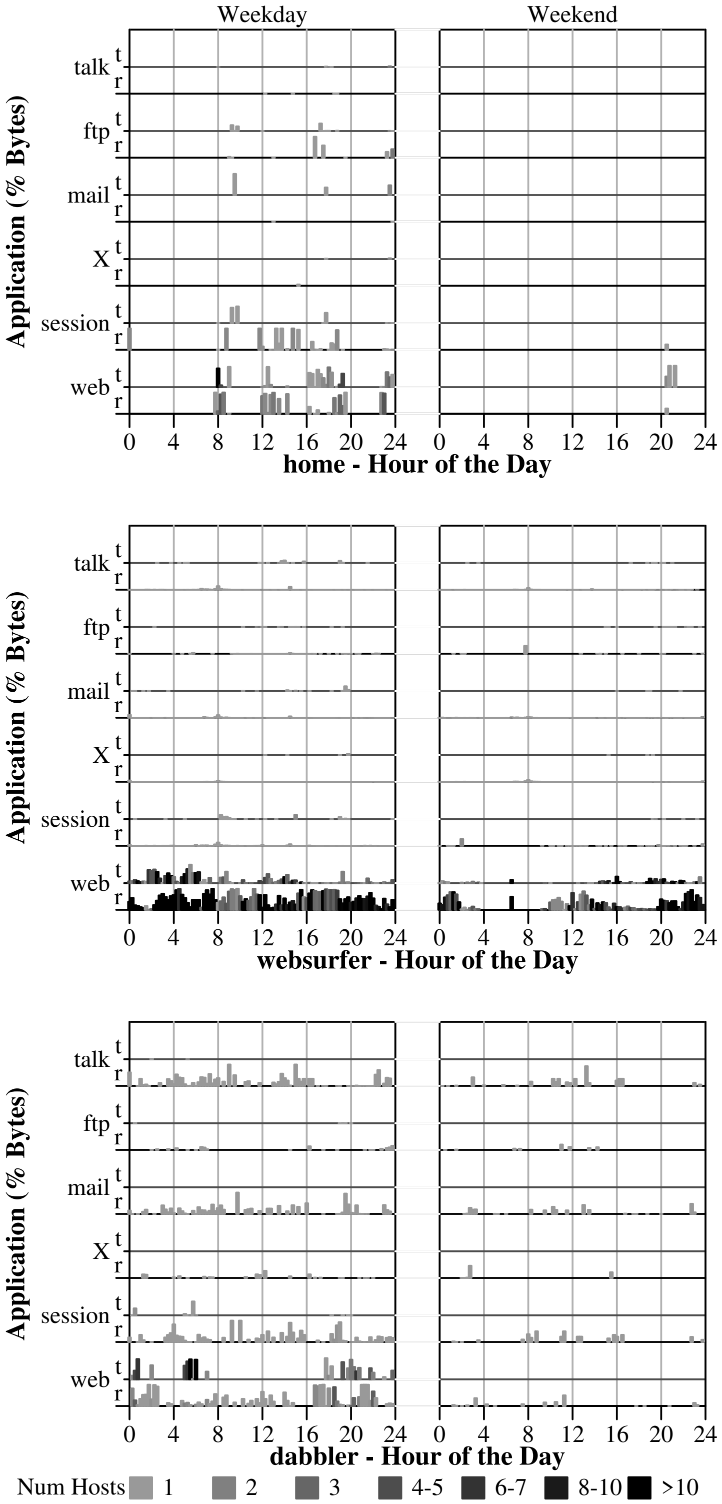


Figure 3.11: Three application mixes (home, web-surfer, and dabbler). Each graph shows the percentage of bytes sent at that time of day per application, for both weekdays and weekends. Each application is split into two graphs, one for traffic to "repeat" hosts (r), and one for traffic to "throwaway" hosts (t). A repeat host is one the user connects to on at least two different days. The darkness of the bar indicates how many different hosts the user connects to. The darker the bar, the more hosts.

Name	Num Users	Description
home*	13	upload in the morning, download for lunch or before going home. Mostly web and ftp, some session and mail traffic. Usually weekday only, occasional weekend traffic.
web-surfer*	12	big web-surfer with lots of sites, session to one or two sites, plus a bit of the other applications.
rare	10	1 or 2 single peaks, always web, sometimes session.
dabbler*	9	fairly evenly distributed among the applications, 3 weekday only, 6 weekday and weekend traffic.
talkies	8	fairly normal hours, weekday and weekend, mostly web and session, but significant chat traffic too.
terminal	7	weekday only, mostly session traffic, some web, occasional ftp.
mail-client	6	3 weekday only, 3 weekday and weekend, leave their laptops overnight as mail-clients, plus some web-surfing and other applications during the day.
late-night	5	lots of chat, web, and ftp late at night. More “normal” traffic (session, web) during the day.
X-term	3	lots of X, session, and web traffic. A little bit of traffic from the other applications.
day-user	3	web and email during the day, a little bit of session, ftp, and X traffic.
ftp	2	lots of ftp with some session and web traffic, a little bit of the other applications.

Table 3.4: Description of the eleven application mixes and number of users per application mix. The only applications considered are web, session, X, mail, ftp, and chat. The starred entries are shown in more detail in Figure 3.11.

do the mixes vary by application and time, but also by the number of hosts to which users connect. Some users connect to as few as six hosts, while others connect to as many as 3,054 distinct hosts. (The router connects to a total of 15,878 distinct hosts over the course of the entire trace; 13,178, or 83%, of those hosts are accessed via the web.)

3.4.2 Web Proxies

Given these access patterns, we can ask whether a web proxy might be an effective technique in the Gates wireless subnet environment. For a rough evaluation, we looked for web sites visited multiple times, either on different days or by different users. Of the 13,178 hosts connected to via the web, 3,894 (30%) are visited multiple times by more than one user, 5,318 (40%) are visited on more than one day during the trace, and 5,359 (41%) are visited either by more than one user or on more than one day. These results indicate that proxies would be at least a partially effective technique in an environment such as the Gates wireless subnet.

3.4.3 Network Sessions and Lease Times

The final question we ask is how long people use the network at a sitting. Since the wireless network is part of the "public" subnet, users must authenticate themselves when they want to access any host outside the subnet. The current policy is to require users to authenticate themselves every twelve hours. Twelve hours was believed to be a good balance between security concerns and user convenience [2]. Of the 1,243 leases handed out over the course of the trace, 23% (272 leases) are renewed within one second of the previous lease's expiration, 27% (310 leases) are renewed within 15 minutes, 30% (339 leases) are renewed within one hour, and 33% (379 leases) are renewed within three hours. Of the 69 users who authenticate themselves, 48 users authenticate themselves again within an hour of a previous lease expiration at least once during the traced period. Given the high percentage of users who re-authenticate themselves very quickly, we conclude that 12 hours is not very convenient for the wireless subnet users and that 24 hours might be a better balance between security and ease of use.

3.5 Future Work

The greatest weakness in our work is its possible specificity: our results only necessarily apply to the Gates wireless network, user community, and choice of device used, specifically, laptops. While we believe many of our observations would hold true in other similar environments, we have not verified this. We would thus like to study other local-area networks, including a much larger building-wide or even campus-wide WaveLAN network to explore whether our conclusions are affected by scale. With a larger network, it might also make sense to look for geographical patterns of user mobility as people go to classes, offices, lunch, and so forth. We also wonder whether our results are specific to an academic environment and would like to perform a similar study in a corporate or commercial setting. Comparing these results to an analysis of the network used by smaller devices such as the PalmPilot would also be interesting. Only through the collection of several different studies can we detect important trends that hold for many wireless environments.

3.6 Chapter Summary

Although these results are specific to this WaveLAN wireless network and this university user community, we hope our analysis is a start on understanding how people exploit a mobile network.

We find that the community we analyze can be broken down into sub-communities, each with its own unique behavior regarding how much users move, when users are active (daily, weekly, and over the course of the trace), and how much traffic the users generate. We also find that although web-surfing and session applications such as ssh and telnet are the most popular applications overall, different users do use different sets of applications at different times and connect to different numbers of hosts. In addition to this user behavior, we also find that significantly asymmetric links would likely be unacceptable in this type of wireless network, and that optimizing packet processing is just as important as optimizing overall throughput.

Chapter 4

Related Network Analyses

In the previous two chapters, we presented detailed analyses of two wireless networks: the Metricom Ricochet packet radio network and the WaveLAN local-area wireless network. While we know of no other publicly available studies of a metropolitan-area mobile network of the same scale as the Metricom Ricochet packet radio network, there are studies of smaller mobile networks. However, these studies all tend to have a less user-oriented focus, as described below.

There have been several studies of the hardware characteristics of radio networks. Researchers at CMU examine their large WaveLAN installation [17]. This study focuses on characterizing how the WaveLAN radio itself behaves, in terms of the error model and signal characteristics given various physical obstacles, rather than on analyzing user behavior in the network as is the focus in our analyses.

Other studies of radio hardware characteristics include Bahl and Padmanabhan's [3] study of the radio transmission properties of local-area wireless networks, concluding that querying the base stations for signal strength information and using that information can determine user location. Walton, et al. [61] studied the link characteristics of a CDMA network and determined that measurements of the network differed significantly from the simulation and model results. Both of these studies focus more on the radio's properties rather than on the behavior of the users of the radios.

Other researchers also studied the campus-wide WaveLAN installation at CMU [7]. However, this study focuses on installing and managing a wireless network rather than on user behavior.

Another related set of papers is joint work from Berkeley and CMU [45]. The main paper in this set outlines a method for mobile system measurement and evaluation, based on trace modulation rather than on network simulation. This work differs from mine in several ways. First, they concentrate on latency, bandwidth, and signal strength rather than radio movement. Second, their emphasis is on using these traces to analyze new mobile systems, rather than on understanding the current system. In this thesis, our goals are to understand how people use an existing mobile system, with the eventual goal of providing parameters that could be used in simulating mobile networks in the future. In Chapter 2, we especially focus on radio movement rather than radio behavior characteristics such as latency and bandwidth.

We and others previously performed a study of a combined wireless and wired network [31] which investigated how users took advantage of both a wired and wireless network. However, this study was limited in that only eight users participated.

Another area of work is related to some results that can be derived from our analyses, specifically, mobility models. There are currently two categories of mobility models: the mobility models that others have proposed in their research, and the mobility models currently being used in the mobile networking community. Unfortunately, these two categories of models are vastly different.

Similar to the mobility models, another derived result from our analyses are traffic models. Both mobility models and traffic models are further discussed below.

4.1 Research into Mobility Models

In Section 2.3.2, we analyze the Metricom Ricochet packet radio network trace to find the common patterns of mobility. While there are very few publicly available studies of wireless networks, there has been a great deal of work in creating mobility models. These mobility models fall into three different categories: models based on common-sense assumptions about user behavior, mathematical models especially focused on location prediction, and models based on observations of a different domain.

Chung, et al. [13] devised a mobility model based on the probability of movement given time considerations, such as when users usually arrive or leave the office. This mobility model is used to determine the effects of bursts of location updates at these peak times on the resource and location management infrastructure. While they use the common sense observation of people traveling between the office and the home, they do not base their model on any actual observations. They also assume that most people arrive or leave the office at approximately the same time, which is contrary to our finding that users with similar movement models may be active at different times.

Liu, et al. [37] create a hierarchical mobility model used for location prediction. The global pattern is a pattern match to possible user mobility paths; these paths are put into a database by the user. The local mobility pattern is based on signal strength at the base stations and is used to help predict the next cell the user will enter; this local pattern is supposed to help catch any user movement that falls out of the global pattern due to outside disturbances, such as a road detour. This model combined with the patterns we found in Chapter 2 might form a reasonable mobility model.

Liu and Maguire [35][36] also present mobility models that use a history of where a user has been to predict the user's next location. Liu, et al. [38] present a mobility model for line-of-sight infrared networks based on a semi-Markov chain to predict user location in order to help position the antenna for best performance. Fong and Lui [19] present a mathematical mobility model that is a general representation of other mobility models used for location update and prediction.

Lam, et al. [32] create a mobility model for PCS networks. Unlike the models mentioned below, their model is based on observations of user behavior, specifically highway traffic data and phone call traffic. However, the model is for a very different type of network (voice rather than data), and the observed mobile user behavior is not based on network data, but rather inferred behavior from external observations, namely car traffic patterns.

Unlike these mobility models, the patterns we find in Section 2.3.2 are not only useful as a basis for a realistic mobility model for location tracking, update, and prediction, but also add to the overall understanding of how users are currently taking advantage of wireless networks. Also, our patterns suggest not only that different users have different mobility patterns, but also that different users have different time patterns.

4.2 Currently Used Mobility Models

While some of the proposed mobility models presented in the previous section are based on common sense, the models typically used when simulating mobile networks to get performance results are much more simplistic and unrealistic.

The most commonly used mobility model is the random waypoint model, in which each mobile node is assigned a new direction, speed, and distance every time the node reaches its destination. Some models incorporate a "pause" time, so that each node pauses for a short time at its destination. Typically, the speeds used are between zero and one meter per second, and the pauses are up to 60 seconds [46][25][11][29][53][26]. A much more infrequently used model is one based on assumed characteristics of disaster areas, conferences, and events, including obstacles and "zones" [29]. Zones define different areas of user movement: some zones exhibit minimal movement, while others have many nodes moving around.

While it would be difficult to compare results of simulations based on these models as opposed to a model based on the patterns presented in Chapter 2 due to the different scales (local-area versus metropolitan-area), the results of the simulations will only be more compelling and applicable to real deployed networks if more realistic mobility models are used.

4.3 Traffic Models

While the commonly applied user mobility models have little to no basis in reality, the state of user traffic models is less dire. Most often, researchers play back traces of TCP connections when modeling user traffic. Because the original traces are actual observations of users, one can argue that this is very realistic. However, typically the traces are of a single TCP connection rather than traces of multiple simultaneous connections. Given that we found users frequently using multiple applications together, a single connection is unrealistic. Also, many users connect to different hosts, another aspect not modeled by a single TCP connection.

There have been some studies of the breakdown of large traffic flows, such as Caceres' characterization of wide-area network traffic [9], Heimlich's characterization of the NSFNet backbone [24], and Crowcroft's analysis of an academic network between the United Kingdom and the United States [14]. However, these are all studies from at least nine years ago of large-scale, long-distance wired networks and do not delve into individual user behavior. We instead study a smaller, wireless network at the individual user level.

Chapter 5

Clustering Algorithms

Statistical techniques such as Markov chains and aggregate measures (e.g., minimum, mean, median) are currently the most common method of analyzing network traces. Visualization techniques are typically just graphing tools such as gnuplot or Microsoft Excel. While these techniques are useful, and we in fact use these tools extensively in our analyses, we propose that other techniques from data mining and information visualization should also be incorporated as standard tools in the networking community.

Currently, only a few research projects have incorporated data mining techniques. Lee, et al. [33] use association rules and similar techniques to mine network logs to build intrusion detection models. Sasisekharan, et al. [56] use decision trees and neural networks to find faults in large telecommunication networks. In this thesis, we discuss using clustering algorithms as another useful tool for finding patterns in large network traces.

Clustering algorithms, or unsupervised learning algorithms, are especially useful for finding patterns in large data sets. We use clustering algorithms extensively in the Metricom packet radio analysis in Chapter 2: we use hierarchical agglomerative clustering [50] to group poletops into locations, and Expectation-Maximization (EM) [15] and k-means [54] to find user mobility patterns. EM is especially useful when the ranges of the different parameters vary significantly.

Given the large amount of data collected for these analyses, trying to find patterns of mobility by manually going through the traces is infeasible. Clustering algorithms are useful for solving this problem because not only can they classify all of the radios, but they can even find patterns that the user might not expect. For example, in the metropolitan-area analysis, we expected some sort of home-office type of mobility and a traveling salesman type of pattern, but we did not expect a pattern reflecting train commuters.

We could have used other classification algorithms such as decision trees. However, these algorithms require that the user already knows the categories into which to classify the data. Because we did not know all of the categories *a priori*, we chose to use clustering, an unsupervised learning algorithm, instead.

Another approach, more commonly used in network analyses, is just to calculate aggregate statistical measures of the data. However, aggregates can hide more detailed patterns in the data. We want to take a more detailed look at the data, and clustering algorithms are ideal for that purpose.

While clustering algorithms have their benefits, we encountered several issues when trying to use them. While these issues may be familiar to the data mining community, they are less familiar to the networking community. Below, we describe the basics of each algorithm before discussing the issues faced in using clustering algorithms so as to facilitate their use in more networking analyses.

5.1 Background

During the course of the Metricom packet radio network trace analysis, we used three different standard clustering algorithms:

- Hierarchical-agglomerative clustering
- K-means
- Expectation-Maximization (EM)

Each clustering algorithm has as input a set of N d-dimensional points. The output assigns each point to one of k clusters. The number of clusters may either be specified *a priori* in k-means and EM, or determined at run-time for hierarchical-agglomerative clustering. We now briefly describe how each algorithm works.

5.1.1 Hierarchical-agglomerative clustering

Hierarchical-agglomerative clustering [50] involves building a tree with $2N-1$ nodes. Specifically, the algorithm starts with all of the original points in a list of available child nodes. A new node is created by finding the two nodes in the list that are closest together, and making the new node the parent of both nodes. The two child nodes are removed from the list of the available child nodes, and parent node, "located" at the midpoint between the two child nodes, is added to the list. This process of creating parent nodes is repeated until there is only one node, the root of the tree, available in the list.

Given this constructed tree, the clusters can be formed in one of two ways. The first way is to just choose a level of the tree as a cut-off. For example, if the third level (depth of three) is chosen as the cut-off, then each node at that level (a total of eight nodes) is the root of a cluster, and all the points that fall under that node in the tree are in that cluster. The second way (which is the one we use) is to choose a cut-off distance. Then, if two nodes are more than that cutoff distance apart, then those two nodes are separate clusters. The deepest nodes in the tree that are at least that distance apart form the roots of each cluster, and again, all the points that fall below a root cluster node are in that cluster.

5.1.2 K-means

K-means [54] is an iterative clustering algorithm, where the number of clusters, k, is known in advance (it is an input parameter for the algorithm). Initially, k cluster centers are randomly chosen. Then, each iteration has two stages:

- In the first stage, each point is assigned to the cluster with the nearest center. How close a point is to a cluster center is determined by a distance function, such as the standard Euclidean distance.

- In the second stage, each cluster calculates a new cluster center by averaging together all of the points assigned to that cluster in the previous stage.

The algorithm is finished either when no points change which cluster they are assigned to, or when the cluster centers move by less than ϵ, which is chosen *a priori*.

5.1.3 Expectation-Maximization (EM)

K-means is really a simplification of Expectation-Maximization (EM) [15], in that k-means assumes that each cluster is spherical. In other words, the variance in each dimension is the same. EM allows for ellipsoidal clusters and is based on Bayesian probability theory. Like k-means, EM is given the desired number of clusters, k, and starts with k random cluster centers. Unlike k-means, however, each cluster in EM is described not only with a center point, but also with a variance in each dimension. Then, each iteration again has two stages:

- For each point, calculate the probability that the point is in the space of input points given the variances and center describing the cluster ($P(x|j)$, if x is the point and j is a cluster) and the overall probability that the point is in the input space ($P(x)$).
- For each cluster, calculate the new mean and variance of each dimension in the cluster by averaging all input points together. Each input point is weighted by the probability of that cluster (given the old center) existing given that the point is in the input space ($P(j|x) = P(x|j) * P(j)/P(x)$, where j is the cluster and x is the point, by Bayes Theorem).

Each point is assigned to a cluster again by which cluster center it is closest to. Also, the algorithm is finished either when no points change which cluster they belong to, or when the cluster centers move by less than ϵ.

Intuitively, the differences between EM and k-means are:

- EM supports a covariance matrix allowing for ellipsoidal clusters. K-means assumes that the variance is the same across all dimensions, thus assuming spherical clusters. Having ellipsoidal clusters is especially useful when the ranges of the dimensions vary greatly (i.e., orders of magnitude), as is the case in the parameters we use in finding user mobility patterns.
- EM calculates the new cluster centers using all the points, rather than just the points assigned to those clusters. Each point is weighted by the probability of that point belonging to that cluster. This probability is calculated using Bayes Theorem.
- EM takes longer to converge than k-means does.

Both k-means and EM depend heavily on the parameters used.

5.2 Cut-off Distance for Hierarchical Agglomerative Clustering

The main difficulty we had in grouping poletops into locations was in choosing a cutoff distance. Hierarchical agglomerative clustering [50] builds a tree by iteratively finding the two closest nodes without parents to become the children of a new parent node. Since the distance associated with each parent node is the distance between its two children, we can use this information to differentiate between different clusters. However, we found that there is no good static value to use as a cutoff between clusters.

We choose to use a dynamic cutoff instead. Given all of the distances between children nodes, we look for the first distance which is greater than half a mile. We then look for an exponential

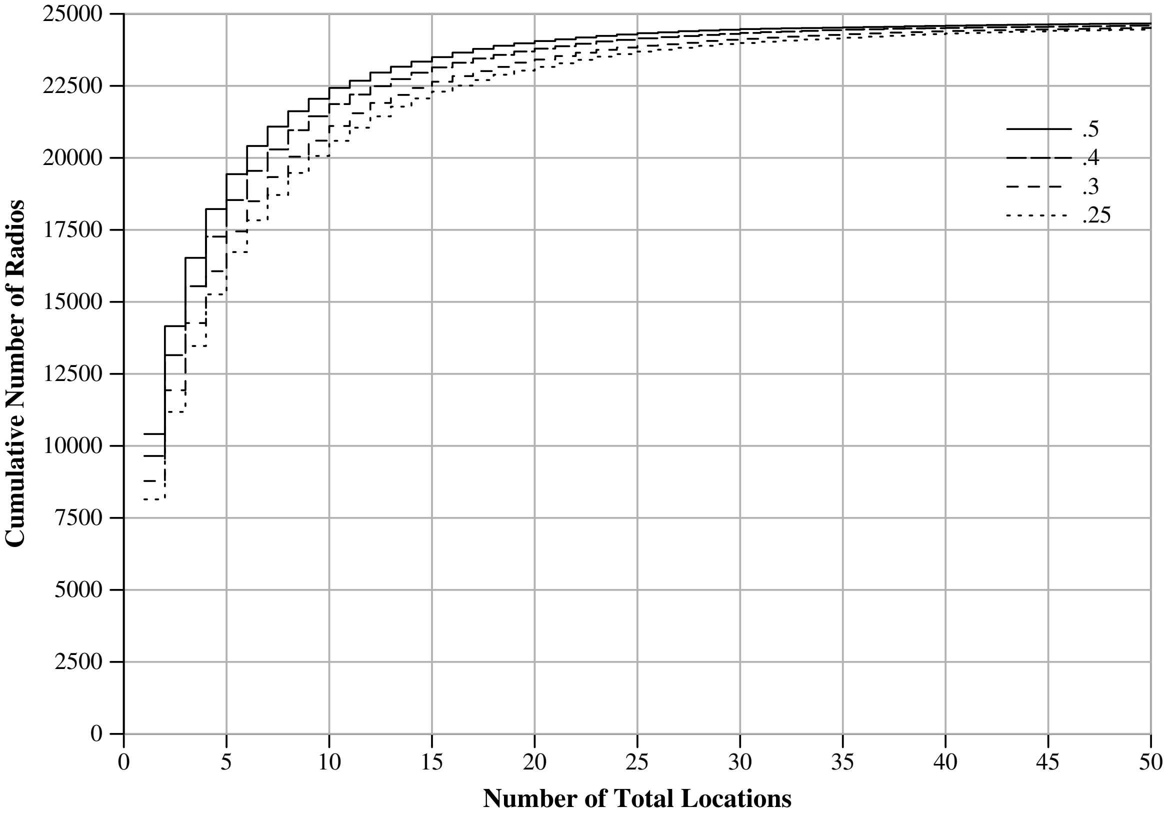


Figure 5.1: Cumulative histogram showing how many radios visit a given number of total locations using a minimal distance of 0.25, 0.3, 0.4, or 0.5 miles. The tail of the graph, which extends to 428 locations, has been cut off to show the detail.

increase (i.e., factor of two) over that distance. This new distance is the cutoff used to differentiate clusters of poletops into locations. We then use k-means to refine the clusters. In k-means clustering, the number of clusters is chosen *a priori*, and the points are assigned to clusters iteratively. This algorithm is repeated for each radio.

Because we do not know *a priori* the number of locations a radio visits over the course of the trace, we cannot just use k-means or EM in this situation. Hierarchical agglomerative clustering helps us determine the number of locations for a radio. We also do not use EM in this situation, since the units in both parameters are the same, thus nullifying the advantages of EM.

The Metricom network already has a coarse granularity: no movement within a building can be detected. We make the network coarser by clustering poletops into locations, with the choice of a minimal distance affecting the resulting coarseness. Figure 5.1 shows how changing the minimal distance affects the number of locations a radio sees. For example, the most locations any radio visits using 0.5 miles as a minimal distance is 176. Using a minimal distance of 0.25 more than doubles this number to 428. We chose to underestimate the number of locations visited per radio, and thus underestimate the total mobility in the network. However, only local mobility estimates are affected: mobility over a larger scale (over a mile) still remains the same. We can look back at Figure 2.10 to see the difference in counting radio movement in terms of poletops and locations.

5.3 Parameter Choice

Another known difficulty in using clustering algorithms is choosing which parameters to use. When we clustered on poletops merely to find the logical grouping into locations, the choice of parameter was obvious: the latitudinal and longitudinal position of the poletop. For finding user mobility patterns, however, we had many choices in how to parameterize each radio.

When we first clustered based on the overall mobility of the radio and on the frequency of radio usage, some other parameters we could have used were:

1. The total number of registrations, since we might want to differentiate radios that are barely active during the trace. However, as we saw from Figure 2.5, the registrations are not necessarily correlated with radio activity.

2. The total number of poletops at which a radio registers. we felt that it would be better to use the location information derived from the poletops, since this is more reflective of the radio's actual movement than the poletops themselves.

3. The distance traveled or the distance between poletops or locations. We chose not to use this type of parameter because of its dependence on the geography of the covered area.

We also could have chosen not to use all nine of the parameters we did use. Changing which parameters are used affects the results of the clustering, including what the resulting clusters are, which radios are assigned to which clusters, and how many distinct clusters there are. Because there is a random factor involved in the initial seeding of the clusters, each run does not necessarily have exactly the same result, making it difficult to compare different runs, especially when varying the input parameters.

Further, when we did the time-based second stage clustering and chose the overall peak times of day for the final dimensionality, we also could have chosen any of the following:

1. Peak time per location. Rather than two overall peaks, base the dimensionality on the total number of locations rather than the number of locations per active day. This choice does not work very well due to users who may visit many more total locations over the course of the trace than they visit per day, leading to unclear results. We could also look at peak time per day of the week, rather than overall peak times.

2. Durations rather than start times. Due to the linear backoff registration scheme, this parameter is hard to quantify.

Sasisekharan, et al. [56] discuss the parameterization they use when employing decision trees and neural networks to find faults in large-scale telecommunication networks.

5.4 Multi-stage Clustering

Perhaps the most important choice we made when finding user mobility patterns was to break the clustering into two separate stages. Multi-stage clustering algorithms existed previously; however, in the previous algorithms, each stage of the clustering uses the same parameters. In practice, what happens is that a subset of the most significant parameters get used to determine the first stage clustering. In the next stage, because each cluster already has common values for that subset of parameters, a different subset of parameters gets used to further break down each cluster into sub-clusters.

We instead choose to separate the parameters explicitly into two subsets, one used in the first stage and the other in the second stage, to facilitate a more intuitive understanding of the clustering results. The first stage of the user mobility pattern finds clusters based on location and usage frequency information, resulting in clusters representing user mobility. The second stage uses parameters based on when the radio is used. In some sense, the first stage finds the mobility patterns while the second stage finds the patterns in time for those mobility patterns.

While we found the results from this explicit separation to be more inherently understandable, a detailed comparison between using all the parameters in all the stages versus single-stage clustering versus our proposed modification is beyond the scope of this dissertation.

5.5 Chapter Summary

Four key issues in using clustering algorithms for network trace analysis are choosing the proper cutoff distance, distance function, parameters, and staging. While all of the issues have been discussed in previous clustering literature, because the networking community does not commonly use clustering when analyzing traces, we explicitly discuss our choices and our justification for making these choices. We also present a novel parameterization split for multi-stage clustering that provides intuitive results.

Chapter 6

Visualization

While clustering algorithms are useful for analyzing these two large network traces, they solve a particular focused problem, such as finding user mobility patterns. An interactive visualization is necessary to explore the data and gain an overall understanding of the data so that the analyst can better focus his analysis on the interesting questions.

There is a large body of work in network visualization, especially in network topology visualization. However, very few of these network visualizations are used (or useful) for analyzing large network traces, and even those that are useful, are not commonly used by researchers in the networking community. For example, Becker, et al. [4] created a series of useful visualizations for analyzing the AT&T phone network after the Loma Prieta earthquake in California. These visualizations, however, seem to have a larger impact in the visualization community than the networking community.

The networking community has developed a few simple visualizations. For example, nam [39] is a simple network animator developed at Berkeley to view simulated and real packet traces, supporting topology layout and packet level animation. Adhockey [1] is another animated visualization specifically designed for visualizing ad hoc networks. Both visualizations suffer from being very low-level, and are therefore only appropriate for smaller traces than the traces analyzed in this dissertation.

While simple visualizations like adhockey and nam are growing in use, network data analysts still typically use spreadsheets such as Excel or simple plotting tools such as gnuplot partnered with custom scripts to analyze the data. However, because the user initially just sees the data as a log file or table of numbers, the user has to decide which data to graph or which statistics to look for. While the results can be useful for describing overall network behavior, and even point out unusual behavior, they are not very useful in helping the user actually understand why the anomalies occur. An interactive visualization that allows the user to zoom in on the anomalies, and to look at them with several different views, can help the user understand the behavior much better. Also, encoding the data visually allows the system to exploit the pattern detection capabilities of the human perceptual system to detect trends and anomalies that the analyst might not think to look for otherwise.

We present three visualizations we developed using the Rivet visualization environment [8] over the course of these analyses:

- Metnet, which overlays the Metricom network on a map of the United States. We use Metnet to gain an initial understanding of the Metricom network itself and the data. For example, we found what is postulated to be a network reconfiguration by using Metnet (see Section 2.4).

- Mars, which lets the user explore the clustering algorithm results when finding user mobility patterns. Mars itself consists of two visualizations, one for the first stage clustering and one for the second stage. We use Mars to determine which parameters to use for the clustering. Mars also animates the clustering algorithms used; this feature we find useful in debugging the algorithm implementations. Using Mars also showed that a critical feature of a visualization environment is being able to incorporate custom algorithms directly into the visualization to enable more direct exploration of the data: we can run the algorithm and then immediately visualize the results. The second-stage clustering visualization allows the user to notice the common occurrence of various time patterns.

- Datagrove, which leverages the hierarchical multi-dimensional semantic structure of the WaveLAN network trace. We present TierBar as a novel user interface for exploring such structures. DataGrove and TierBar are presented in more detail than Metnet and Mars because of their applicability to a broader range of data sets.

Each visualization is developed for a specific purpose in the analysis, and is therefore only applicable for its stated task. However, they all illustrate how analysts can learn much more from being able to see and interact with the data, rather than just being lost in a sea of numbers or aggregate graphs. The specific visualization techniques of brushing (selecting a subset of the data and having that subset highlighted in all views of the data), selection (filtering the data down to a user-selected subset), integration of custom algorithms into the visualization, and leveraging the hierarchical structure of the meta-data are important design choices that help us answer our original set of questions.

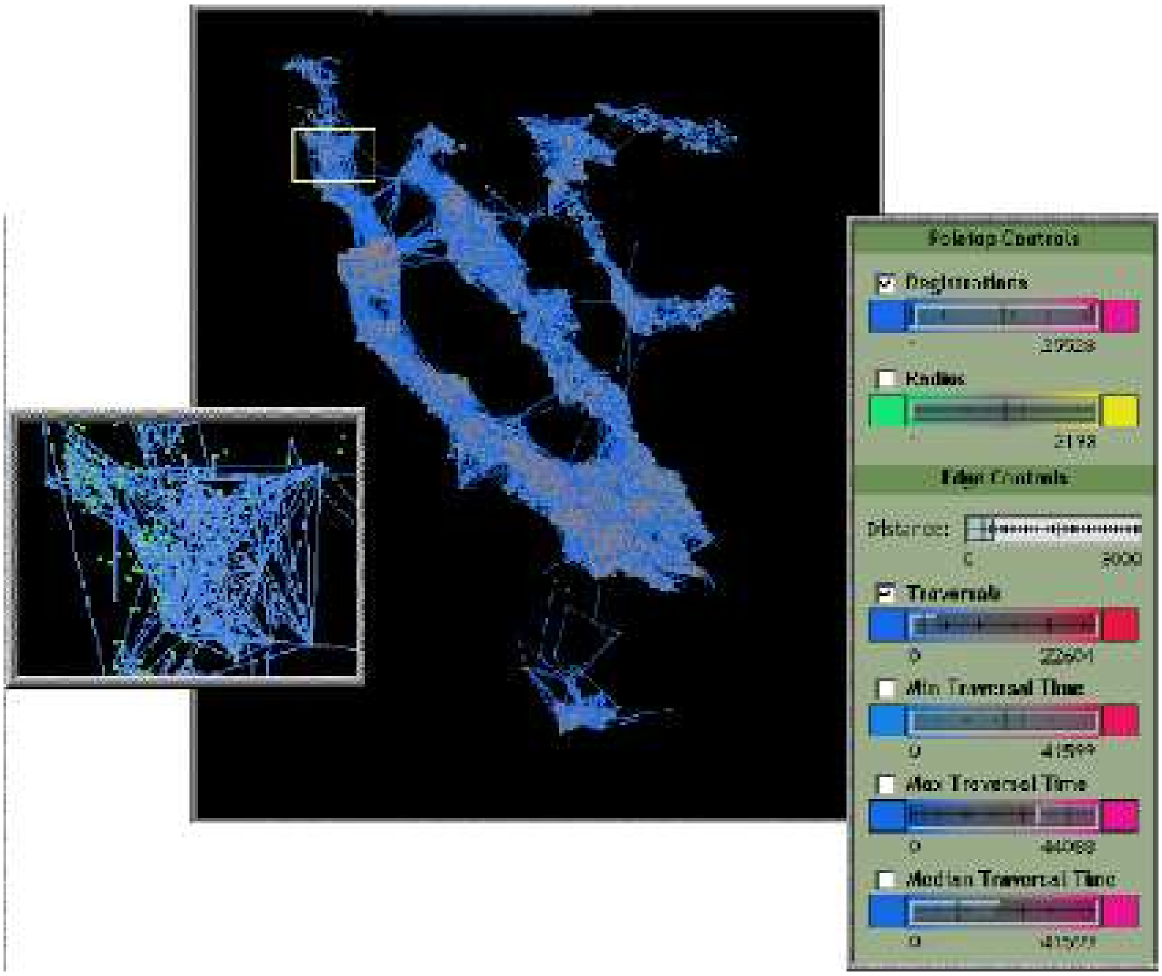


Figure 6.1: Visualization of the Metricom packet radio network infrastructure.

6.1 Metnet

Metnet, the first visualization we developed, is useful for understanding the overall structure of the Metricom network itself. A static picture of Metnet and the user interface is shown in Figure 6.1. The visualization is based on the linkmap and nodemap created by Becker, et al. [5] to visualize the AT&T telephone call network.

Each node in Metnet corresponds to a poletop receiver in the Metricom network. The nodes are laid out geographically on top of an outline map of the United States. The user can zoom and pan the display. Each node is color-coded by either the number of registrations or radios that poletop had to handle over the course of the trace. The nodes can be filtered to show only the subset of poletops that handled the user-specified range of registrations or poletops.

Each edge between two poletops exists only if at least one radio changed its registration from one poletop directly to the other poletop during the course of the trace. The user can choose to color-code each edge by the number of radios making that traversal or how fast (minimum, maximum, and median time) that traversal is made. Each end of the edge can have a different color, one end corresponding to to the traversals from the first node to the second node, and the other end corresponding to the traversals from the second node back to the first node. The edges can be filtered by distance in addition to showing only the subset of edges that handled a user-specified range of the above values.

The filtering is especially important, as Becker et al. discuss [5], to reduce visual clutter. Without filtering, the visualization is incomprehensible. The filtering is dynamic, so that the display automatically adjusts as the user adjusts the filter. This interactive feedback is important in allowing the user to fine-tune the desired filter.

This visualization allows us to find such unexpected anomalies as a network reconfiguration, where over the course of an hour (5pm to 6pm on March 17), several poletops changed location from the Bay Area to Texas. We found this reconfiguration by looking at the minimum traversal times for links, and noticed some long distance links with really short traversal times. This observation prompted us to take a closer look at the raw data, so that we then noticed this network reconfiguration.

6.2 Mars

We then developed Mars to help understand the results from the clustering algorithms used to find user mobility patterns. Mars has two parts, a first-stage clustering visualization and a second-stage clustering visualization.

6.2.1 First-stage Clustering

First-Stage Mars, shown in Figure 6.2, is based on scatterplot matrices [4][23], commonly used by statisticians to spot trends in large multi-dimensional data sets. Typically, each axis of the matrix corresponds to a variable, and each row or column corresponds to a different value for that variable. Then each entry in the matrix plots the data corresponding to the set values for the two matrix axis variables.

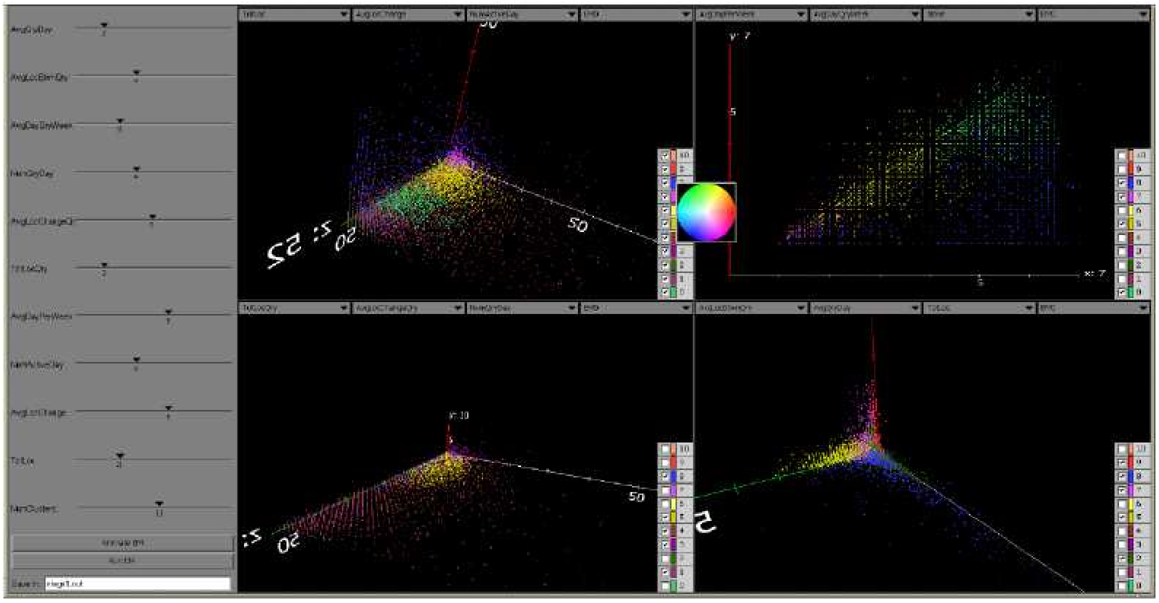


Figure 6.2: Visualization used for exploring the results of the first-stage clustering to find user mobility patterns.

This visualization has two parts: a control panel and the display, consisting of a matrix of three-dimensional scatterplots. This matrix display, however, does not correlate each matrix axis with a particular variable; instead, each pane is independent from the other panes.

Each pane in the matrix display contains a single three-dimensional scatterplot. Using pulldown menus, the user can interactively choose which variable to use for the X, Y, Z, and color axes for each individual scatterplot in a pane. If the user chooses to color-code the scatterplot, then a legend automatically appears on the right edge of the pane. Each entry in the legend has a checkbox, a colored rectangle, and a label indicating the value corresponding to that color. The user can use the checkbox to control whether the points corresponding to the label are displayed. Using the mouse, the user can directly rotate, zoom, or pan each scatterplot. A popup menu can reset the scatterplot to its initial position, as well as control whether or not the edges are shown.

The edges in the scatterplot correspond to clusters. If the user sets the color to be the results from the clustering algorithm, then the edges are also color-coded accordingly. An edge is drawn between a point and the center of the cluster it belongs to. These edges and cluster centers can be animated, using the control panel, to show the progression of the clustering algorithm. This animation is useful for debugging the clustering algorithm implementation, which is directly integrated into Rivet [8].

The user can use the control panel to animate the most recent clustering algorithm run, to run the

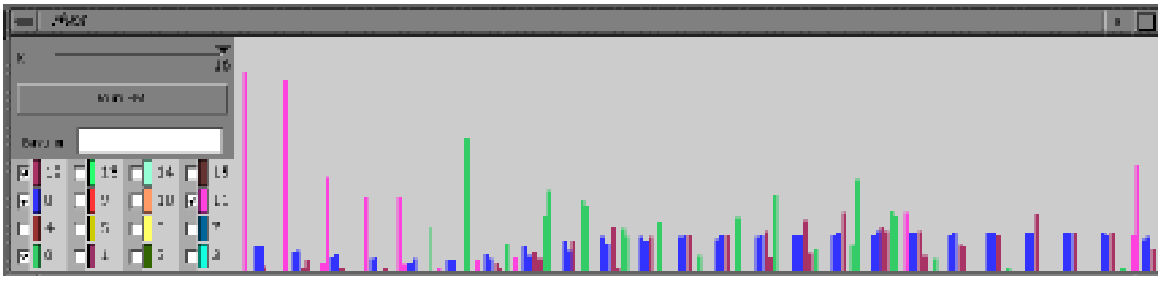


Figure 6.3: Visualization used for exploring the results of the second-stage clustering to find user mobility and time patterns.

clustering algorithm, to save the results of the clustering algorithm to a file, and to set the parameters of the clustering algorithm. The parameters are how many clusters the algorithm should create and how much each dimension should be weighted (the weight of a dimension is zero if the dimension should not be used in the clustering algorithm run at all).

The main advantage to this visualization is the direct integration of the clustering algorithm. Because we are using a less-popular clustering algorithm, EM, and because we have custom distance functions, other visualization packages, such as the SGI Mineset visualization package [57], do not provide these algorithms and do not provide for a way to link in user-supplied implementations. Without this capability, it becomes very frustrating to run the algorithm externally, bringing in the results manually for every single algorithm run. By directly integrating the algorithm and being able to configure each run of the algorithm directly from the visualization, the user gets quicker feedback and immediate visualization.

6.2.2 Second-stage Clustering

The second-stage Mars visualization, shown in Figure 6.3 is similar in that it also consists of a control panel and a display. The control panel in this visualization is much simpler, consisting only of a slider controlling the desired number of clusters, a save-to-file controller, and a legend. Because the second-stage clustering has a set of predetermined dimensions (the amount of activity for each hour of the day for weekdays and weekends), we do not need to set the weight for any of these dimensions. The legend controls what color each cluster is, and like the first-stage clustering, the user can use the checkbox to control whether or not that cluster is displayed.

The display consists of a clustered histogram, two bars per hour per cluster, indicating the average amount that the cluster is used during that hour either on the weekend or weekday. The bars are color-coded by cluster so that the user can directly compare different clusters.

6.2.3 Summary

We created both the first-stage and second-stage visualizations to explore the results from clustering the radio parameters to find user mobility and time patterns. However, both visualizations serve an additional purpose of illustrating the benefits of using a visualization environment such as Rivet. Rivet allows us not only to incorporate our custom clustering algorithms directly into the environment, but also to create visualizations most suited to the parameters quickly. This easy extensibility and rapid prototyping is especially useful when exploring data sets this large, since we can change the visualization as needed to correspond to our current needs. For example, we originally clustered the second-stage based on a different set of time parameters and we visualized the second-stage clustering also using a scatterplot matrix. When we changed the parameters to our current set, the visualization changed from a scatterplot matrix to the clustered histogram. Being able to make these changes rapidly is a key feature when exploring network traces.

6.3 DataGrove

The final visualization we develop, DataGrove, is for exploring the WaveLAN network trace. One of the main differences between the WaveLAN trace and the Metricom trace is the fact that we have additional knowledge, such as users' office locations and research groups that we want to leverage in our exploration. Our goal in developing DataGrove is not only to explore our network trace, but also to develop a conceptual framework for exploring this type of hierarchical multi-dimensional data, and an interface facilitating this type of exploration.

Because we are developing not only a visualization but also an underlying framework applicable to a wider range of data sets, this section is more in-depth than the previous visualizations. Specifically, Section 6.3.1 provides background information about hierarchical multi-dimensional structures in general. Related work is presented in Section 6.3.2, and Section 6.3.3 presents the virtual fact table conceptual model we use throughout the rest of this chapter. Section 6.3.4 presents the TierBar user interface, and Section 6.3.5 describes how virtual fact tables are generated. We then show how this visualization and underlying infrastructure is useful for exploring the network trace data.

6.3.1 Background

The database community has long recognized the importance of hierarchical multi-dimensional data, in which structured semantic knowledge is connected to a single main relational data table. This additional semantic knowledge is especially useful when analyzing large data sets.

For example, a store may keep transaction records in a simple relational table: each transaction record includes attributes such as product, customer, store location, and transaction date. Supplemental tables contain additional data relating to each attribute, such as product type, supplier, warehouse, etc. These separate data sets combine with the raw transaction data to form a multi-dimensional data hierarchy.

This combination of simple data augmented with hierarchically organized semantic knowledge, or meta-data, occurs in a wide range of problem domains. The meta-data hierarchy enables analysts to aggregate and organize the data set logically into different levels of detail, presenting the data in a more manageable, but still meaningful, form.

Unfortunately, most visualization packages do not take advantage of data structured in this fashion. Instead, they only support data in a single flat table, stored either internally or in a relational database. With a flat table, additional semantic knowledge must be explicitly included in the table, which can significantly increase its memory footprint. While a relational database can store the data more efficiently, in either case the hierarchical structure is typically unknown to the visualization and therefore not exposed to the user. Consequently, the user must have explicit knowledge of the hierarchy and manually incorporate this knowledge when using the visualization.

The database community describes hierarchical multi-dimensional structures through the use of star and snowflake schemas [22][59]. Both types of schemas have a root fact table, consisting of the basic data that the user wants to explore. Each field in the root fact table is either a dimension or a measure: dimensions are independent fields, either nominal or quantitative, whereas measures are dependent quantitative fields.

Additional semantic knowledge describing the dimensions is stored in associated existence tables. Each existence table represents semantic information for the dimension at a particular level of detail.

We can describe the WaveLAN trace data using a snowflake schema, as shown in Figure 6.4. In this schema, the dimensions are Time, User, Access Point, Application, Direction, and Remote Host, and the measures are Packets and Bytes. This schema is a snowflake schema because the User set of existence tables branches into both Research Group and Wing/Floor. Star schemas are a subset of snowflake schemas with only single, non-branching lines of existence tables.

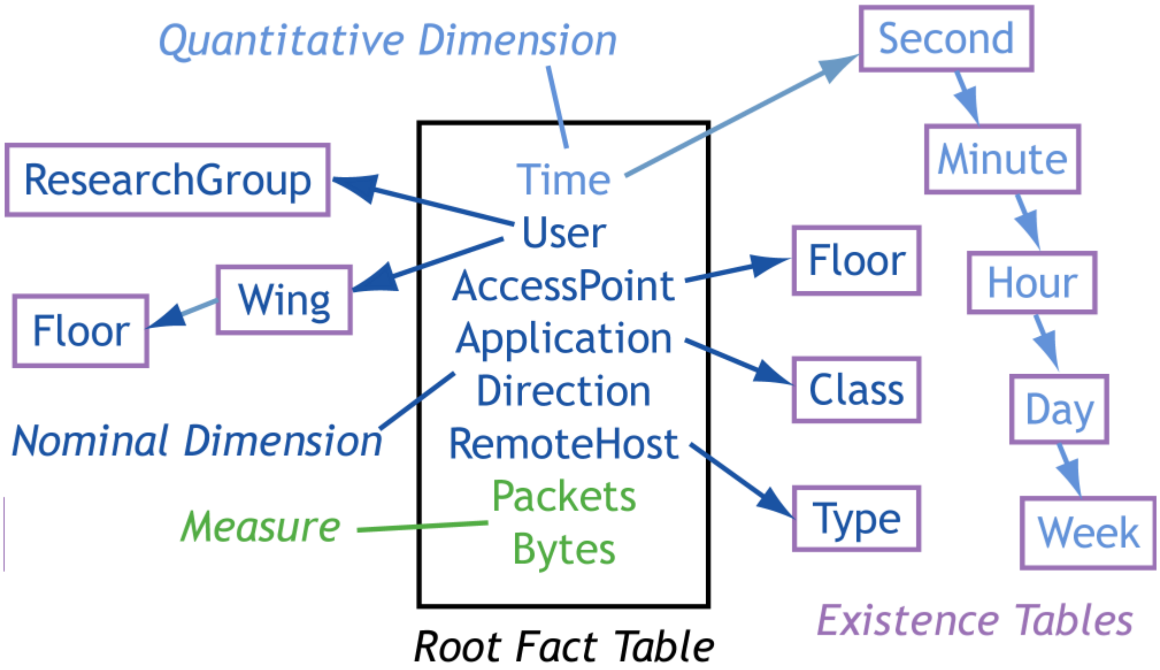


Figure 6.4: Snowflake schema of the wireless network trace data.

6.3.2 Related Work

There has been a great deal of work on visualizing trees, including techniques from tree maps [30] to cone trees [51] to hyperbolic browsers [44]. However, these techniques handle data that is itself a hierarchy, such as filesystems; our work is focused on visualizing data sets that are organized using hierarchical meta-data, such as the schema shown in Figure 6.4.

The Pad++ project [6] has explored interfaces for navigating large data spaces using semantic zooming and task-based filtering. While Pad++ has explored multiscale displays, they do not address data management schemes for hierarchical multi-dimensional data.

The Sage group at CMU has worked with hierarchical data, combining their Aggregate Manipulator with dynamic queries [21]. However, they restrict their discussion of data manipulation to desired operations such as on-the-fly aggregation and filtering.

A popular approach for handling hierarchical data in the database community is the datacube [22]. Users can query the datacube for slices using interfaces such as the Pivot Table [42]. Datacubes handle hierarchical data by essentially building a datacube of datacubes, which users can then query. While our conceptual model is similar, we normalize the resulting datacubes into tables and provide a focus-plus-context (showing the overview of the data as well as the more detailed area the user has focused in on) interface for user exploration.

6.3.3 Conceptual Model

Using hierarchical meta-data to display a data set at multiple levels of detail is similar to using techniques such as mipmaps [63], clipmaps [58], and r-sets [48] to render images at different resolutions with a constant amount of work. A mipmap is an image pyramid, where each level of the pyramid represents a different level of detail (i.e., the image at a different resolution). The display resolution available to the image determines which level of the pyramid is used. Clipmaps extend mipmaps to handle arbitrarily large textures by filtering, or clipping, the more detailed levels of the pyramid to the region being displayed. In both cases, zooming into and out of the image corresponds to moving up and down the pyramid, changing the level of detail. Another technique similar to mipmaps is r-sets, which vary the level of detail of the horizontal and vertical dimensions independently.

We can apply these ideas from the graphics community to hierarchical multi-dimensional data. However, rather than varying the level of detail over a single dimension (for mipmaps, the image resolution), we need to handle multiple levels over multiple dimensions. Changing the level for a dimension is equivalent to transforming how the root fact table is aggregated over that dimension. For example, changing the User dimension to the Research Group level means that all users in a particular research group are aggregated together. We then define a virtual fact table to be the root fact table transformed so that each dimension is at a specific level of detail. Examples of virtual fact tables given the schema presented in Figure 6.4 are shown in Figure 6.5.

Exploring a hierarchical multi-dimensional data set is then equivalent to exploring the space of virtual fact tables. One can think of navigating a graph of virtual fact tables, similar to the mipmap image pyramid, where a user can go from one virtual table to another by changing the level of detail of a single dimension. A portion of this graph for our data schema is shown in Figure 6.5.

We can also apply the clipmap extension to our virtual fact table model. Filtering the virtual fact table to show only a subset of the domain (e.g., showing only the graphics research group in our data) is analogous to clipping mipmaps. Changing this filter corresponds to panning the displayed image data.

We can divide the exploration of this type of data into two parts: the user interface to explore and specify which virtual fact table we want to display, presented in Section 6.3.4, and the generation of virtual fact tables, presented in Section 6.3.5.

Note that there is an additional stage in the exploration process: determining how to display the

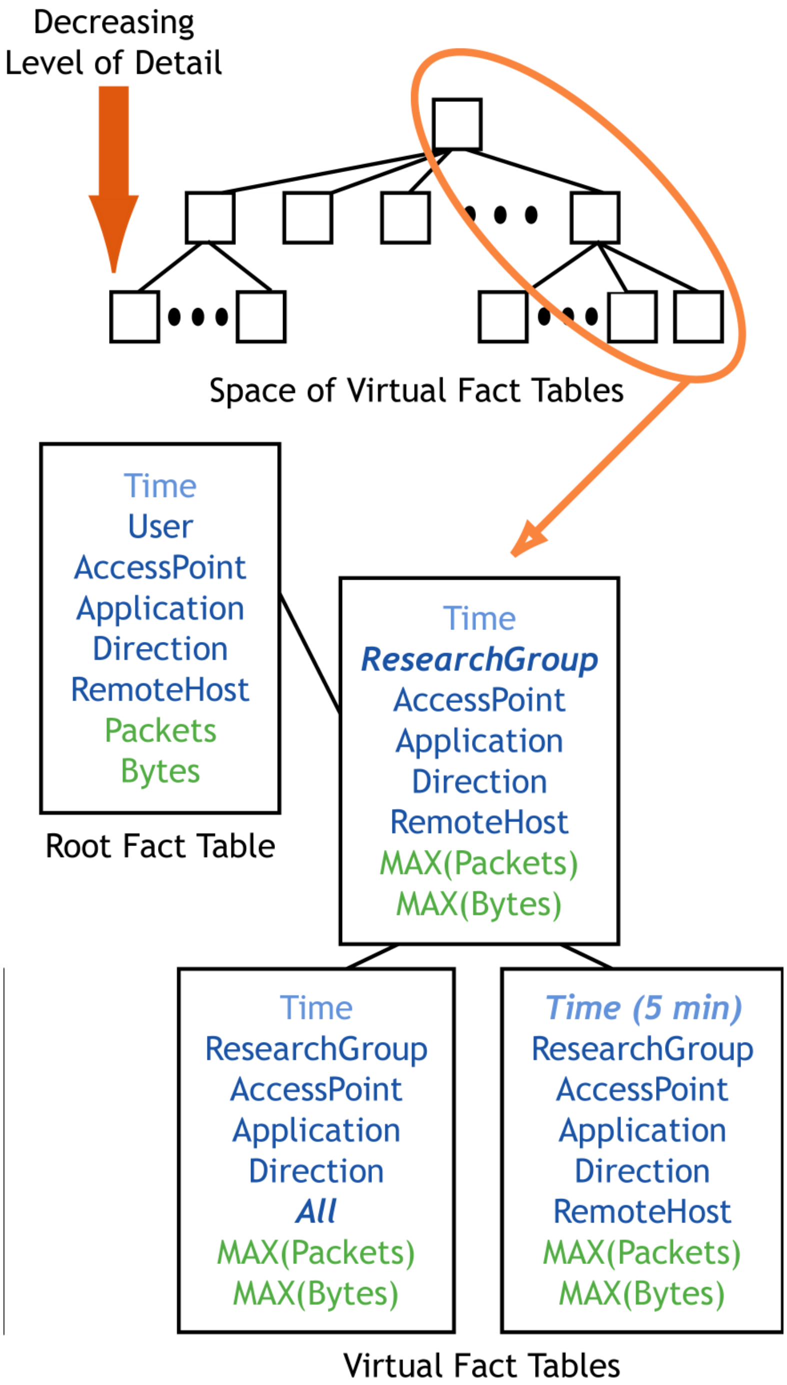


Figure 6.5: The space of virtual fact tables. The italicized entry in each virtual fact table indicates which dimension changed.

generated virtual fact table. However, in this chapter we focus on a single visual representation, specifically a collection of strip charts displaying data varying over time.

6.3.4 TierBar Interface

To enable easy exploration of hierarchical multi-dimensional data, we had several goals when designing the TierBar interface:

1. Enable the user to navigate the dimensional hierarchies easily and intuitively by zooming into and out of the data.

2. Allow the user to choose how to traverse dimensional hierarchies. This ability is especially important for exploring snowflake schemas, which can branch.

3. Provide a focus-plus-context view of the hierarchy.

4. Provide direct manipulation of elements in the hierarchy, enabling the user to scroll the data display and change the data filters.

Figure 6.6 shows the TierBar for both a nominal and a quantitative dimension. The interface is multi-tiered, with each tier corresponding to a different level of the dimensional hierarchy. The tiers are presented from lower levels of detail (shown on the outside of the TierBar) to higher (shown on the inside), corresponding to navigation from the leaves of the snowflake schema towards the dimension in the root fact table. In addition to the levels explicitly specified in the schema, we provide an "All" node, corresponding to the lowest possible level of detail. Conceptually, the "All" level is an implicit child of all leaf nodes in the meta-data hierarchy. Figure 6.7 shows how one dimension of the schema is mapped to the hierarchy traversed by the TierBar.

Each entry in a tier is color-coded: blue if the entry can be further expanded and green otherwise. Each entry also has faint lines indicating how many elements there would be if the user expanded that entry. The user zooms into and out of the hierarchy by clicking on a blue entry to expand or collapse it. Trapezoids connect an expanded entry with its children in the next tier. An optional final tier enables users to select a subset of the entries from the current last level in the hierarchy; this subset is drawn in red. To prevent disconcerting shifts in the interface during exploration, each tier maintains a constant size; the total amount of space allocated to the TierBar grows and shrinks as levels are expanded and collapsed.

The user can change the data filter by selectively expanding entries or scrolling the selected areas in the hierarchy. For example, if the user begins with the Research Group tier and expands

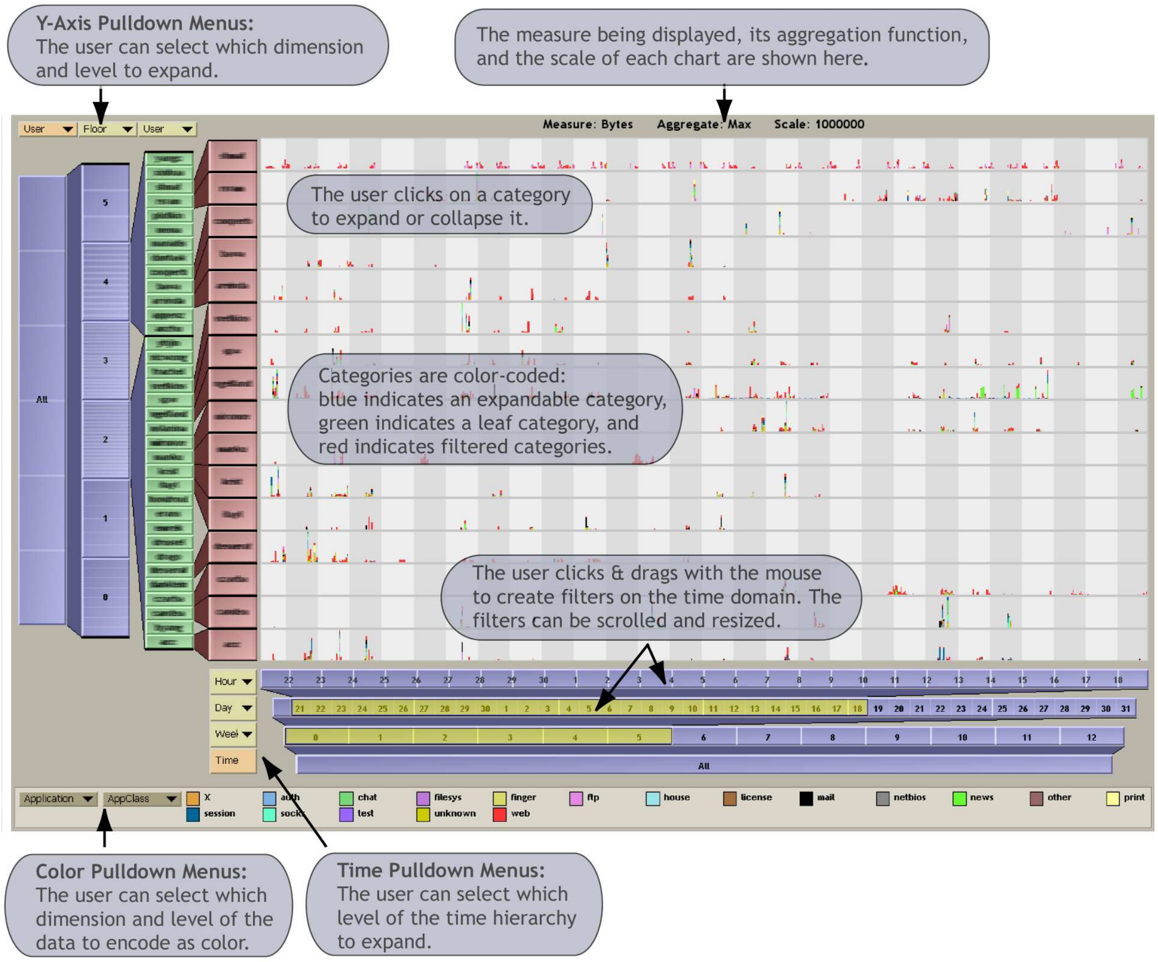


Figure 6.6: The TierBar user interface for exploring hierarchical structured data. The TierBar for User, a nominal dimension, is shown on the left, and the TierBar for Time, a quantitative dimension, is shown on the bottom. Note that the Hour labeling is lighter: we label the tier using Day granularity because there is insufficient space to label each hour. For publication, we intentionally blur the user names in the screen shot for privacy.

the graphics entry, only users in the graphics group will be displayed. If the user then collapses the graphics entry and expands the robotics entry, only robotics students will be displayed. The user can expand multiple entries: clicking on the theory entry will show both robotics and theory students. The user can also scroll a selected area by dragging it. In a quantitative dimension, for example, we can scroll the display from week to week. This functionality is also useful in a nominal dimension with too many entries (e.g., the application level has over 100 entries, and a strip chart per entry would be illegible, so scrolling becomes necessary at that point).

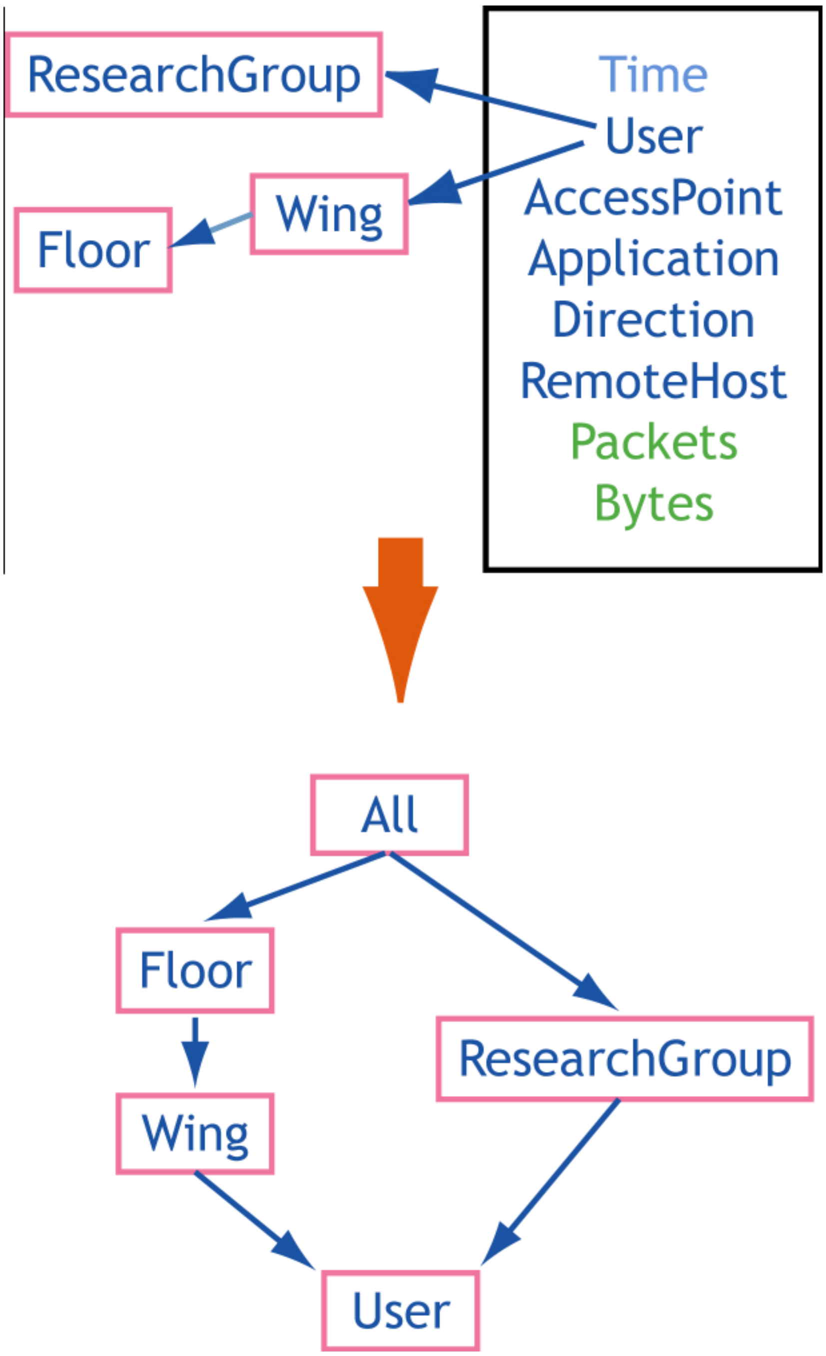


Figure 6.7: How a schema specification is converted into a dimensional hierarchy traversed by the TierBar. The interface always starts from the lowest level of detail, All, and successively introduces more detail.

If there is more than one choice on how to expand an entry, as is the case in a snowflake schema, a pulldown menu enables the user to choose the expansion path. For example, the user can choose to expand the User dimension by Floor or Research Group.

The interfaces to nominal and quantitative dimensions are very similar: both allow the user to specify the desired level of detail and the data filter. For a quantitative dimension with a continuous domain, the level of detail determines the discretization granularity of the data. For example, specifying the Time dimension at the Hour level of granularity means that all tuples within an hour are aggregated together.

Because each TierBar determines both the level of detail and a filter for a single dimension, several TierBars together uniquely specify the desired virtual fact table.

6.3.5 Virtual Fact Table Generation

Given a schema specification (described in Appendix A) and a virtual fact table specification from the TierBars described in the previous section, we now discuss the series of data transformations that must be performed on the root fact table to generate a virtual fact table.

The first step is to group together tuples that have the same values at the specified level of detail for each dimension. All tuples are initially in a single group, and are then subdivided according to the virtual table specification.

For every nominal dimension listed in the root fact table, there are three cases:

1. If the desired level is ”All,” then the dimension is not used when creating the virtual fact table.
2. If the desired level is the raw data in the root fact table, e.g., if the desired level for the User dimension is User, then the data is grouped by that field.
3. Otherwise, the data from the specified existence table is joined into the root table and then grouped by the field from the existence table. This may require a multi-level join, depending upon the depth of the existence table in the hierarchy. For example, specifying the User dimension at the Floor level means that Floor must be joined into Wing and Wing into User.

For every quantitative dimension listed in the root fact table, there are two cases:

1. If the desired level is ”All,” then the dimension is not used when creating the virtual fact table.
2. Otherwise, the tuples are grouped according to the specified granularity.

Once the data is grouped properly, the tuples in each group are aggregated into a single tuple: the dimension values are the common values of the group, and the measure values are computed using an aggregation function (e.g., sum, max, mean), specified either by the user or in the schema specification. All the aggregated tuples are then merged together to form a single virtual fact table.

Finally, any requested filters are applied. We choose to apply the filters last to provide better response time when the user modifies the filters, at the expense of slower response time when the user changes the level of detail.

We currently provide two optimizations for virtual fact table generation. First, we pre-calculate the data at different time granularities and incrementally operate on these pre-calculated tables instead of the root fact table. We presently only precalculate the Time dimension since it affects the data size the most. Second, all generated virtual fact tables are saved in files. We check to see if the virtual fact table file exists before calculating it.

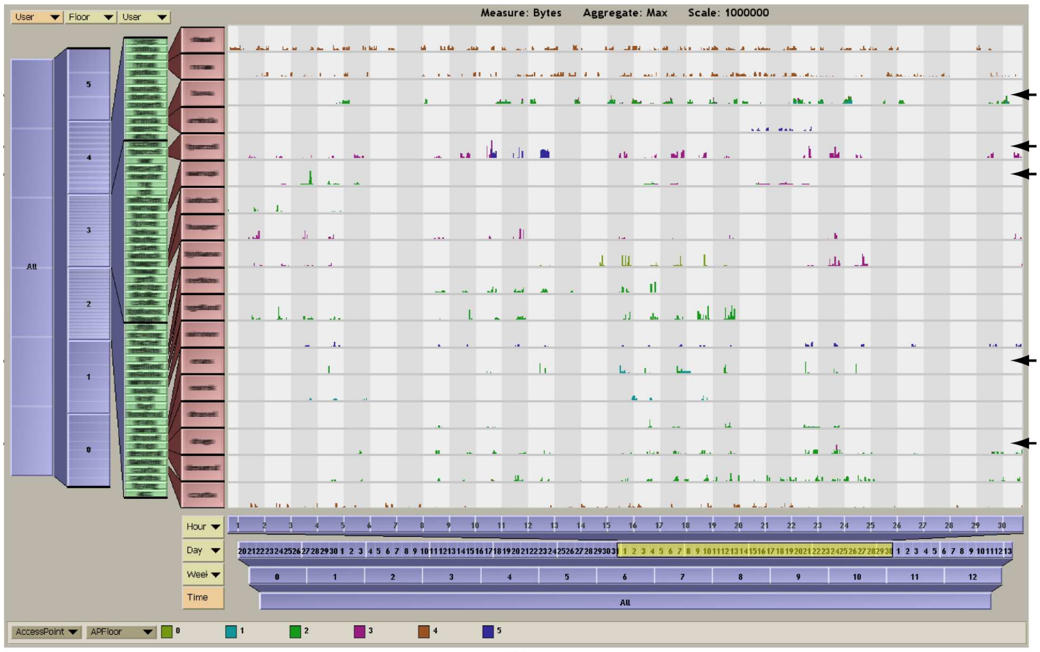
6.3.6 Results

Using the user interface and data infrastructure described above, we developed a visualization in the Rivet visualization environment [8] to explore traffic characteristics of the wireless network. We wished to answer questions such as how traffic is distributed between incoming and outgoing bytes and among users and applications. Given our focus on traffic characteristics as a function of time, We display the data as a collection of strip charts (categorized by one of the data dimensions) showing the data measures (packets and bytes) as a function of time.

The visualization consists of four parts: a TierBar controlling Time, a TierBar controlling the categorization, a color controller, and the strip chart display itself. The user uses the bottom TierBar to choose the time granularity and displayed range. The TierBar on the left is used to choose the dimension, level of detail, and data filter to be used for grouping the data into strip charts. The color controller consists of two pulldown menus, which enable the user to specify the dimension and level of detail for the color palette. The colors in the palette are automatically selected [60]. The state of the TierBars and the color selector are mapped directly to a virtual fact table specification. They specify the level of detail and filters for all visible dimensions; all other dimensions are set to the ”All” level of detail. When the user changes the state of the TierBar, the system computes the virtual table specification and generates a new virtual fact table; the resulting table is displayed in the strip charts.

Using the TierBars, we can focus our search in both nominal and quantitative dimensions. Figure 6.8 shows how to take advantage of this capability while trying to understand where users were

(a)



(b)

Figure 6.8: Two successive screen shots showing how often users change location while using the wireless network. (a) Overview showing that more users with offices on floors 2-4 move around. (b) Zoom on individual users, showing that only the users indicated by arrows move during a day. For publication, we intentionally blur the user names in the screen shots for privacy.

located when using their laptops. In Figure 6.8a (18,055 records after aggregation but before filtering), we group users by their office location and use color to encode their locations while using the network during the month of November. The resulting display shows that users on floors 2, 3, and 4 use their laptops from several different locations. For example, users with offices on the fourth floor tend to be spread out among the second, third, fourth, and fifth floors, especially in the second half of the month. In Figure 6.8b (11,312 records), we then zoom in on the floors of interest. This display then shows that most users use their laptops from a location other than their offices, while only five users actually visited multiple locations during the month.

A second example, shown in Figure 6.9, illustrates the logical grouping of data for ease of exploration and the progressive refinement of the time dimension's visible range and granularity. It also illustrates how the hierarchy can be used for color-coding the data display. Our goal is to understand how incoming and outgoing traffic differs among different types of applications. Because there are so many applications (129 in total), we logically group applications with similar characteristics (e.g., session applications such as telnet, ssh, rlogin, etc.) to find the overall trends. We can further expand a class to display individual applications if required. Figure 6.9a (53,297 records) shows incoming versus outgoing traffic for 14 different application classes at a one-hour granularity over a 22-day period. This display shows the unusual occurrence of several peaks where outgoing traffic dominates incoming traffic across several application classes.

We further refine the time range and granularity, zooming in on one of the peaks to show more detail on this unusual occurrence. Figure 6.9b (1,895,047 records) shows a 3.5 hour period at a 15-second granularity, showing that the single peak is in fact several bursts of traffic occurring over a period of 1.5 hours. We can then change the color mapping from Direction to Wing, as shown in Figure 6.9c (1,362,176 records), to determine that the 3b wing is responsible for the traffic.

Overall, we find that exposing the hierarchical structure to the user is effective for exploring and logically aggregating the data. By taking advantage of this structure, the analyst can more easily and intuitively understand the data.

6.3.7 Summary

Our goal in developing DataGrove was to create a visualization for exploring the WaveLAN network trace data. In the process, we have demonstrated the importance of leveraging hierarchical multi-dimensional meta-data for exploring data. We also present both a conceptual model used to support hierarchical multi-dimensional data within a visualization environment and an interface, TierBar, for exploring this type of data.

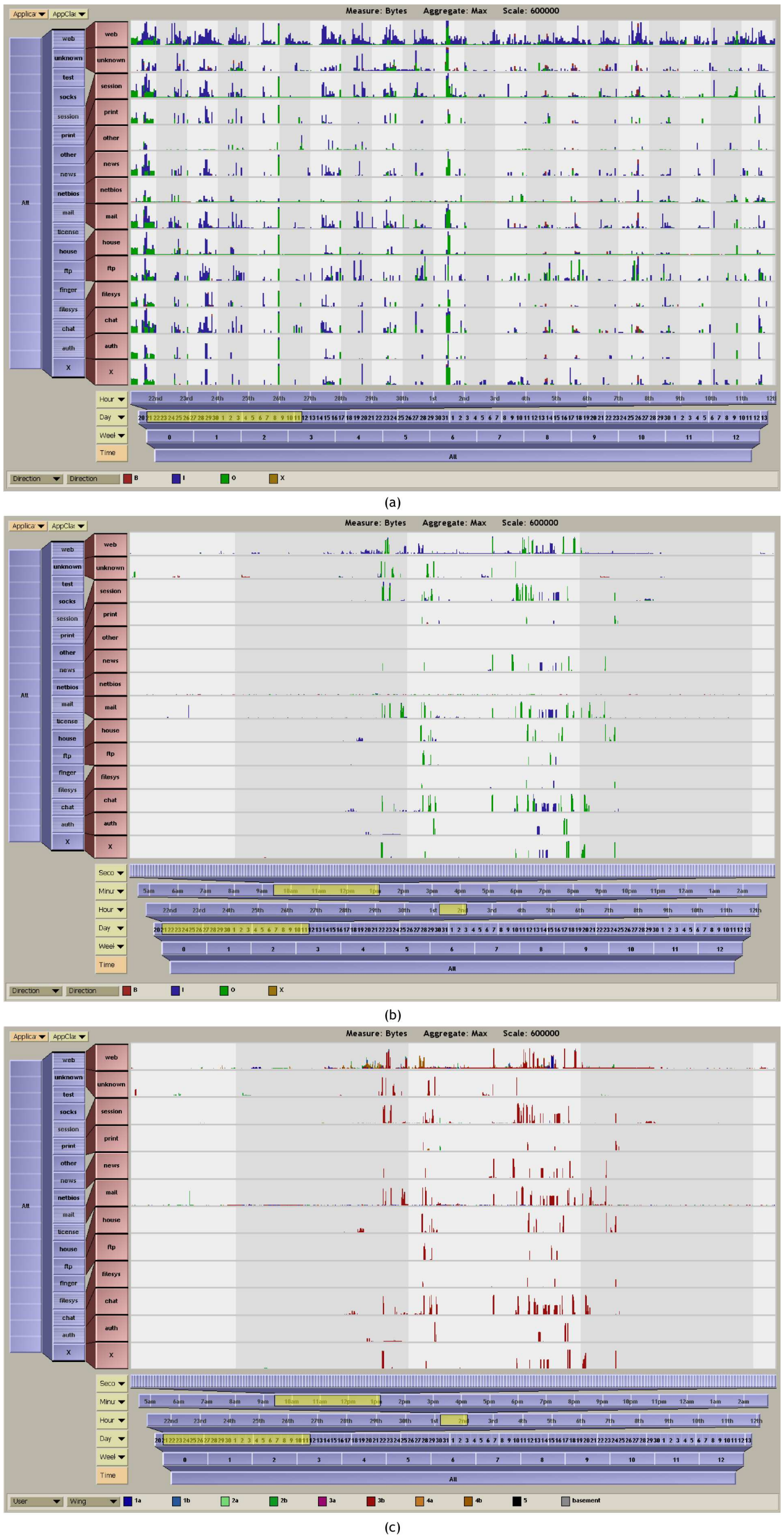


Figure 6.9: Three successive screen shots showing how application traffic is distributed among users and between incoming (I), outgoing (O), internal (B), and unknown (X) bytes. (a) Overview of 22 days showing an unusual amount of outgoing traffic across several different applications types. (b) Zoom into a 3.5 hour range showing that the outgoing traffic is distributed over a 1.5 hour period. (c) Same as (b), except that the user's office location is now encoded by color, showing that users on the 3b wing are responsible for the unusual behavior.

Allowing the user to explore hierarchical structured data rather than a flat file is useful when analyzing the wireless network trace data. For example, it enables us to aggregate applications logically by class, thus creating a more easily understood visualization. We believe these techniques are generally applicable to other data sets. Extending them will allow the system to handle even more complex data structures and generate a wider variety of visualizations.

Chapter 7

Conclusion

The goal of this dissertation is to increase our understanding of how users take advantage of wireless networks. Understanding mobile users is critical not only for choosing research and development topics that will have the most impact, but also in creating more realistic models of user behavior, especially important given the mobile networking community's dependence on simulation.

The secondary goal of this dissertation is to encourage others to do similar analyses of other wireless networks. Additional analyses are useful for further increasing our understanding of mobile users. We can compare analyses to see how users of different networks or in different environments behave. Also, if there are several analyses over a period of time, we can detect trends in user behavior to predict future user patterns.

To accomplish the primary goal of this dissertation, we have presented detailed analyses of two very different wireless networks: the Metricom Ricochet packet radio network and the WaveLAN local-area wireless network installed in the Gates Computer Science building at Stanford University. While these two analyses are not very comparable given the large difference in both the scale of the networks and the data collected for the analysis, the results do add to a growing understanding of how users take advantage of wireless networks.

In analyzing the Metricom Ricochet packet radio network, a metropolitan-area network used by almost 25,000 users, we find that users are more active during the day and on weekdays, and that most users move within a local area as opposed to longer distance movements. Network designers can take these results into account when planning the mobile network infrastructure, such as when to prepare for peaks in network usage. We also find that a majority of users are stationary on any given day, but that there is a small fraction of users that are highly mobile. This conclusion might lead network designers to conclude that optimizing handoffs in this type of metropolitan-area network is not a high priority since it does not affect a large percentage of users. We also find user mobility and time patterns that can be used to create more realistic models of user behavior when simulating mobile networks, a common technique used by the networking community to analyze the performance of mobile systems. For example, a mobility model based on the patterns we find in Chapter 2 might have a set of users corresponding to each pattern. The sizes of the set would be proportional to the sizes of the clusters. Each user would have a set of locations he visits, with the number of locations falling with the distribution specified by the pattern. Each user would also have his own distinct time pattern, as specified with the time pattern clustering.

When analyzing the WaveLAN local-area wireless network in the Gates Computer Science building at Stanford University, we found an unexpectedly high number users taking advantage of their laptops for synchronous communication with other users. This result indicates that synchronous communication between mobile users of data networks might be a developing trend and a fruitful area for future research. We also find that users fall into distinct location-based sub-communities as opposed to being a single, fairly homogeneous community, each with its own behavior regarding movement and periods of activity. This demarcation of users into sub-communities is a result that future network designers might need to take into account when allocating resources to various parts of the wireless network. Another unexpected result when doing the WaveLAN analysis concerns the symmetry of traffic: while incoming traffic dominates outgoing traffic overall (34 billion bytes compared to 12 billion bytes), the opposite tends to be true during periods of peak throughput, implying that significant fixed asymmetry in network capacity could be undesirable for the users. Given the current research on asymmetric networks, this result is especially relevant. In addition to these results, we also do an in-depth study of user traffic to find common user application mixes, which can be used to create more realistic user traffic models when simulating mobile networks.

While the results from these analyses are necessarily valid only for the particular networks and user communities studied, they do increase our understanding of mobile users, thus fulfilling the primary goal of this dissertation. These analyses also help accomplish the secondary goal by acting as a blueprint for others doing analyses of wireless networks.

In addition to providing these blueprints, we also present clustering algorithms and detailed, complex, and interactive visualizations as useful tools the networking community should use when doing such analyses. Although these tools are known to other communities in the computer science field, they are not commonly used in the networking community. To facilitate their use in analyzing wireless networks, we discuss several issues involved in using these tools, such as parameter choice in clustering and interactivity in visualization.

We hope the analyses presented in this dissertation will be used by the mobile networking community, not only to serve as blueprints for similar studies, but also to focus research on the topics that will impact users the most and to create more realistic models of user behavior when simulating mobile networks.

7.1 Future Work

There are several directions in which to expand this work. The most obvious direction is analyses of other wireless networks. Specifically, it would be interesting to compare the WaveLAN network analysis to an analysis of a larger academic user community with users not in the computer science department to see if any of our conclusions are specific to computer scientists (such as the late hours or the application mixes found). Another comparison would be to a corporate network. A larger local-area network in general would be interesting, since a mobility study would have more significance.

With regards to the Metricom analysis, it would be interesting to compare the Metricom network to another large-scale network such as a cellular phone or CDPD network to see how the mobility models differ. Another reason for studying other large-scale wireless networks would be to answer the questions we were unable to answer due to lack of data, such as the mobility of users while maintaining a connection, traffic mix analyses, or latency and bandwidth analyses.

Another interesting comparison would be to analyze wireless networks where the devices used are much smaller, such as cellular phones or personal digital assistants (PDA's), such as the Palm Pilot. We expect that in such a network we would see many more mobile users, since smaller devices are much easier to carry around and use in a variety of locations than laptops.

With regards to the clustering algorithms, an analysis comparing the effectiveness of multi-stage clustering versus single-stage clustering versus multi-stage clustering using the same set of parameters in all stages is needed. Such an analysis should pinpoint the advantages and disadvantages of each type of algorithm and use several different data sets to see if there are particular problem domains for which each type of clustering algorithm is suited.

There are also several directions to expand the visualization research. One area specific to the networking community would be to determine if any of the visualizations already developed, Mars, Metnet, or DataGrove, would be more generally applicable to exploring other network traces. We expect that Metnet is fairly specific to the Metricom or other wide-area networks spanning a large geographic area and Mars is useful only if clustering is being used in the analysis. However, given that most network traces have similar dimensions to the ones used in DataGrove, that visualization is the most likely to be generally applicable in the networking area.

Another area of future work is to explore DataGrove's effectiveness in exploring other hierarchical multi-dimensional data sets, such as transaction records. For DataGrove to be effective, it also needs to support graph types other than strip charts. We also plan on experimenting with alternative user interfaces, incorporating semantic zoom, modeling more sophisticated data hierarchies, and optimizing virtual fact table generation. These areas are now discussed in further detail.

First, we would like to demonstrate the broad applicability of the techniques described in this section by using them to analyze other hierarchical multi-dimensional data sets. We also want to expand the visualization to support other graph types in addition to strip charts.

In addition, we would like to explore other interfaces for generating the virtual fact table specification. One possible interface, similar to the Windows File Explorer [43], would allow each dimension to be expanded or collapsed like a directory. Another possible technique, similar to the TableLens [49], would equate zooming with space allocation. As more space is allocated to a data subset, the dimensions along that table axis are expanded; as less space is allocated, the dimensions are collapsed. One difficulty in designing these interfaces is deciding how to allow the user to choose between multiple expansion paths, an issue when exploring snowflake schemas. We solve this problem in TierBar with pulldowns, but other solutions should also be explored.

A further extension would be not only to change the level of detail being displayed when zooming, but also to change the visual representation of the data as well, as in Pad++ [6]. For example, when the level of detail of time gets fine enough, the graph would change from a utilization strip chart to a Gantt chart of individual events.

We also plan to extend the underlying data infrastructure. First, we would like to support ad hoc grouping and aggregation, such as grouping the robotics and theory research groups together. One possible interface for ad hoc grouping would be to allow dragging of entries within a tier; when one field is dragged on top of another, the two are aggregated. Another extension would be to allow cross-branch expansions of a dimension. For example, we would allow the User dimension to expand first by Wing, and then by Research Group. Other more complex extensions would be to link multiple fact tables together into a single, more complex hierarchy, and to allow selective aggregation and non-uniform expansion of data sets.

Finally, we would like to optimize the data infrastructure. One reason for generating the virtual fact tables by composing transformations is to allow for incremental generation. Rather than always transforming the root fact table, we can cache previously generated virtual fact tables, and then choose the closest one as the basis for generating the next table. This is a generalization of the optimization introduced in Section 6.3.5. Because virtual fact tables are typically smaller than the root fact table, this incremental generation should allow for quicker response, especially for large data sets.

7.2 Traces

While we cannot release the Metricom Ricochet packet radio traces, we can release the WaveLAN traces we collected. An anonymized version of =these traces will be available on our website: http://mosquitonet.stanford.edu.

Appendix A

Schema Specification

The schema specification encapsulates the domain-specific knowledge for a data set. It first describes the dimensions and measures of the root fact table. Then, for each dimension, it also specifies the hierarchy of existence tables characterizing that dimension. This structure can then be queried by the TierBar to determine how each level of the hierarchy can be expanded automatically.

The particular information needed for the root fact table is:

- the name of the file containing the table's records
- a list of dimensions; for each dimension, we need:
  - its name
  - its type (quantitative or nominal)
  - a list of existence tables describing the dimension
- a list of measures; for each measure, we need:
  - its name
  - the default aggregation function (e.g. sum, max, mean)

Similarly, for each existence table we need:

- the name of the file containing the table's records. For example, the UserWing.bin file contains a list of records, each of which maps a user's name to the wing in which his office is located.
- a list of levels; for each level, we need:

- its name
- its type (quantitative or nominal)
- for quantitative fields, we can optionally specify an interval giving the numeric relationship between the field and its parent (e.g., the interval between second and minute is 60)
- a list of existence tables further describing the level

We define the specification in XML [64] so that schemas can be easily read by humans as well as easily parsed. The document type description (DTD) and a partial XML specification of the schema in Figure 6.4 is given below. Note that while we currently just use a pointer to a file containing the existence table records, we could easily extend the specification to include this information directly.

XML Document Type Description:

```
<!ELEMENT table (datafile?,(dimension|measure)+)>
<!ELEMENT datafile (\#PCDATA)*>
<!ELEMENT dimension (name,type,interval?,table*)>
<!ELEMENT measure (name, type, aggfunc)>
<!ELEMENT type (nominal|quantitative)>
<!ELEMENT nominal EMPTY>
<!ELEMENT quantitative EMPTY>
<!ELEMENT name (\#PCDATA)*>
<!ELEMENT interval (\#PCDATA)*>
<!ELEMENT aggfunc (\#PCDATA)*>
```

Partial XML specification: we only specify the Time and User dimensions and the Packets measure here.

```
<table>
    <datafile>Root.bin</datafile>
    <dimension>
        <name>Time</name>
        <type><quantitative/></type>
        <interval>1</interval>
        <table>
```

```
            <dimension>
                <name>Second</name>
                <type><quantitative/></type>
                <interval>1</interval>
                <table>
                    <dimension>
                        <name>Minute</name>
                        <type><quantitative/></type>
                        <interval>60</interval>
                        <!--Hour, Day, Week, etc.-->
                    </dimension>
                </table>
            </dimension>
        </table>
    </dimension>

    <dimension>
        <name>User</name>
        <type><nominal/></type>
        <table>
            <datafile>UserWing.bin</datafile>
            <dimension>
                <name>Wing</name>
                <type><nominal/></type>
                <table>
                    <datafile>Floor.bin</datafile>
                    <dimension>
                        <name>Floor</name>
                        <type><nominal/></type>
                    </dimension>
                </table>
            </dimension>
        </table>
```

```
        <table>
            <datafile>UserResearchGroup.bin</datafile>
            <dimension>
                <name>ResearchGroup</name>
                <type><nominal/></type>
            </dimension>
        </table>
    </dimension>

    <measure>
        <name>Packets</name>
        <type><quantitative/></type>
        <aggfunc>MAX</aggfunc>
    </measure>
</table>
```

Bibliography

[1] Adhockey. www.monarch.cs.cmu.edu/ns-faq/faq.html

[2] Appenzeller, G., Roussopoulos, M., and Baker, M. User-Friendly Access Control for Public Network Ports." Proceedings of IEEE INFOCOM, 1999, March, 1999, p. 699-707.

[3] Bahl, P. and Padmanabhan, V.N. User Location and Tracking in an In-Building Radio Network. Microsoft Research Tech Report: MSR-TR-99-12 (Feb 1999).

[4] Becker, R.A., Cleveland, W.S., and Wilks, A.R. Dynamic Graphics for Data Analysis. Statistical Science, vol 2, p. 355-395, 1987.

[5] Becker, R.A., Eick, S.G., and Wilks, A.R. Visualizing Network Data. IEEE Transactions on Visualization and Computer Graphics, vol. 1 no. 1. p. 16-28, March 1995.

[6] Bederson, B., Hollan, J., Perlin, K., Bacon, D., and Furnas, G. Pad++: A Zoomable Graphical Sketchpad for Exploring Alternate Interface Physics. Journal of Visual Languages and Computing, 1996, pp. 3-31.

[7] Bennington, B.J. and Bartel, C.R. Wireless Andrew: Experience Building a High Speed, Campus-Wide Wireless Data Network. Proceedings of the Third Annual ACM/IEEE International Conference on Mobile Computing and Networking. August, 1997. p. 55-65.

[8] Bosch, R., Stolte, C., Tang, D., Gerth, J., Rosenblum, M., Hanrahan, P. Rivet: A Flexible Computer Systems Visualization Environment. Computer Graphics, 34, 1 (February, 2000), p. 68-73.

[9] Caceres, R., Measurements of Wide Area Internet Traffic, Technical Report 89/550 at the University of California, Berkeley, Computer Science Department, 1989.

[10] Case, J.D., Fedor, M., Schoffstall, M.L., and Davin, C. Simple Network Management Protocol (SNMP). RFC 1098. April, 1989.

[11] Casteneda, R. and Das, S. Query Localization Techniques for On-demand Routing Protocls in Ad Hoc Networks. Mobicom 1999, p. 186-194.

[12] Cheshire, S. and Baker, M. A Wireless Network in MosquitoNet. IEEE Micro, February 1996, 44-52.

[13] Chung, M., Kim, T., Cho, H., and Sung, D. Modeling of terminal mobility to evaluate the number of location updates. 1997 IEEE International Conference on Communications. Towards the Knowledge Millennium. ICC '97. Conference Record p. 1266-70.

[14] Crowcroft, J. and Wakeman, I. Traffic Analysis of some UK-US Academic Network Data, Proceedings of INET 1991, Copenhagen, June, 1991.

[15] Dempster, A.J., Laird, N.M., and Rubin, D.B. Maximum Likelihood from Incomplete Data via the EM Algorithm. J. Royal Statistical Society, B39:1-38, 1977.

[16] Droms, R. Dynamic Host Configuration Protocol (DHCP). RFC 2131. March, 1997.

[17] Eckardt, D. and Steenkiste, P. Measurement and Analysis of the Error Characteristics of an In-Building Wireless Network. Computer Communication Review 26, 4 (October 1996), 243-254.

[18] Feldmann, A., Gilbert, A., and Willinger, W. Data networks as cascades: investigating the multifractal nature of Internet WAN traffic. Computer Communication Review, vol.28, no.4, p. 42-55, October, 1998.

[19] Fong, C. and Lui, J. Transient performance analysis for location update protocols in cellular networks. Proceedings Sixth International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems, p. 282-7, 1998.

[20] Fox, A., Johanson, B., Hanrahan, P., and Winograd, T. Integrating Mobile Computing Devices into an Interactive Workspace. Submitted to Computer Graphics and Animation, special issue, Off the Desktop.

[21] Goldstein, J. and Roth, S. Using Aggregation and Dynamic Queries for Exploring Large Data Sets. Proceedings of SIGCHI 1994, September 1994, p. 23-29.

[22] Gray, J., Chaudhuri, S., Bosworth, A., Layman, A., Reichart, D., Venkatrao, M., Pirahesh, H., and Pellow, F. Data Cube: A Relational Aggregation Operator Generalizing Group-By, Cross-Tab, and Sub-Totals. Proceedings of the Twelfth International Conference on Data Engineering, February 1996, pp. 152-159.

[23] Hartigan, J.A. Printer graphics for Clustering. Journal of Statistical Computation and Simulation, 4, p. 187-213.

[24] Heimlich, S. Traffic Characterization of the NSFNET National Backbone. Proceedings SIGMETRICS Conference on Measurement and Modeling of Computer Systems, p. 257-258, May, 1990.

[25] Heinzelman, W., Kulik, J., and Balakrishnan, H. Adaptive Protocols for Information Dissemination in Wireless Sensor Networks. Mobicom 1999, p. 174-185.

[26] Holland, G. and Vaidya, N. Analysis of TCP Performance over Mobile Ad Hoc Networks. Mobicom 1999, p. 219-230.

[27] Howard, J.H., Kazar, M.L., Menees, S.G., Nichols, D.A., Satyanarayanan, M., Sidebotham, R.N., and West, M.J. Scale and Performance in a Distributed File System. ACM Transactions on Computer Systems, Vol. 6, No. 1, February 1988, p. 51-81.

[28] Jacobson, V., Leres, C., and McCanne, S. tcpdump. Available via anonymous ftp to ftp.ee.lbl.gov, June 1989.

[29] Johansson, P., Larsson, T., Hedman, N., Mielczarek, B., and Degermark, M. Scenario-based Performance Analysis of Routing Protocols for Mobile Ad-hoc Networks. Mobicom 1999, p. 195-206.

[30] Johnson, B. and Shneiderman, B. Treemaps: A Space-filling Approach to the Visualization of Hierarchies. Proceedings of IEEE Visualization 1991 Conference, October 1991, pp. 284-291.

[31] Lai, K., Roussopoulos, M., Tang, D., Zhao, X., and Baker, M. Experiences with a Mobile Testbed. Worldwide Computing and Its Applications, Lectures notes in Computer Science (1368). Berlin: Springer, 1998, 222-237.

[32] Lam, D., Cox, D., and Widom, J. Teletraffic Modeling for Personal Communications Services. IEEE Communications, 35(2), p. 79-87, February, 1997.

[33] Lee, W., Stolfo, S., and Mok, K.W. A data mining framework for building intrusion detection models. Proceedings of the 1999 IEEE Symposium on Security and Privacy, p. 120-32.

[34] Leland, W., Taqqu, M., Willinger, W., and Wilson, D. On the Self-Similar Nature of Ethernet Traffic. Computer Communications Review, 23, 4 (October, 1993), p. 183-193.

[35] Liu, G. and Maguire, G. A predictive mobility management algorithm for wireless mobile computing and communications. 1995 Fourth IEEE International Conference on Universal Personal Communications. Record. Gateway to the 21st Century, p. 268-72, 1995.

[36] Liu, G. and Maguire, G. A class of mobile motion prediction algorithms for wireless mobile computing and communications. Journal of Special Topics in Mobile Networks and Applications (MONET), vol.1, no.2, p. 113-21, October, 1996.

[37] Liu, T., Bahl, P., and Chlamtac, I. Mobility modeling, location tracking, and trajectory prediction in wireless ATM networks. IEEE Journal on Selected Areas in Communications, vol.16, no.6, p. 922-36, Aug 1998.

[38] Liu, T., Carruthers, J., and Chlamtac, I. Mobility tracking and prediction for wireless infrared LANs. Proceedings of the SPIE - The International Society for Optical Engineering, vol.3532, p. 50-60, 1999.

[39] MASH project. nam: Network Animator. mash.cs.berkeley.edu.

[40] Metricom User Manual for Ricochet Packet Radio Networks, 1998.

[41] Metricom, Inc. http://www.metricom.com.

[42] Microsoft Corporation. Microsoft Excel User's Guide, Microsoft, Redmond, WA, 1995.

[43] Microsoft Corporation. Microsoft Windows 95. Available: http://www.microsoft.com/windows95/, cited March 2000.

[44] Munzner, T. H3: Laying Out Large Directed Graphs in 3D Hyperbolic Space. Proceedings of the 1997 IEEE Symposium on Information Visualization, October 1997, pp. 2-10.

[45] Noble, B., Satyanarayanan, M., Nguyen, G., and Katz, R. Trace-Based Mobile Network Emulation. Computer Communication Review 27, 4 (October 1997), 51-61.

[46] Oh, S. and Wasserman, K. Optimality of greedy power control and variable spreading gain in multi-class CDMA mobile networks. Mobicom 1999, p. 102-112.

[47] Paxson, V. End-to-End Internet Packet Dynamics. Computer Communications Review 27, 4 (October 1997), 139-152.

[48] Peachey, D. Texture on demand, unpublished manuscript, 1990.

[49] Rao, R. and Card, S.K. The Table Lens: Merging Graphical and Symbolic Representations in an Interactive Focus+Context Visualization for Tabular Information. Proceedings of the Conference on Human Factors in Computing Systems (SIGCHI'94), pp. 318-322.

[50] Rasmussen, E. editors: Frakes, William B. and Baeza-Yates, R. Information Retrieval: Data Structures and Algorithms. Prentice Hall, Upper Saddle River, NJ, 1992, 419-442.

[51] Robertson, G., Mackinlay, J. and Card, S. Cone Trees: Animated 3D Visualizations of Hierarchical Information. Proceedings of the ACM SIGCHI Conference on Human Factors in Computing Systems, 1991, pp. 189-194.

[52] Ritter, M. Personal communique, October 1999.

[53] Royer, E. and Perkins, C. Multicast Operation of the Ad-hoc On-Demand Distance Vector Routing Protocol. Mobicom 1999, p. 207-218.

[54] Russell, S. and Norvig, P. Artificial Intelligence: A Modern Approach. Prentice-Hall, Upper Saddle River, NJ, 1995.

[55] Sandberg, R., Goldberg, D., Kleiman, S., Walsh, D., and Lyon, B. Design and Implementation of the Sun Network Filesystem. Proceedings of the Summer 1985 USENIX Conference, June, 1985, p. 119-130.

[56] Sasisekharan, R., Seshadri, V., and Weiss, S.M. Data mining and forecasting in large-scale telecommunication networks. IEEE Expert, vol.11, no.1, p. 37-43, February, 1996.

[57] SGI Mineset. http://www.mineset.sgi.com.

[58] Tanner, C., Migdal, C., and Jones, M. The Clipmap: A Virtual Mipmap. Proceedings of SIGGRAPH 1998, August 1998, pp. 151-158.

[59] Thomsen, E. OLAP Solutions: Building Multidimensional Information Systems. Wiley Computer Publishing, New York, 1997.

[60] Travis, D. Effective Color Displays: Theory and Practice. Academic Press, London, 1991.

[61] Walton, R., Wallace, M., and Howard, S. CDMA downlink performance issues. Ninth IEEE International Symposium on Personal, Indoor and Mobile Radio Communications, p. 308-312, 1998.

[62] WaveLAN. http://www.wavelan.com.

[63] Williams, L. Pyramidal Parametrics. Proceedings of SIGGRAPH 1983, July 1983, pp. 1-11.

[64] The XML Industry Portal. Available: http://www.xml.org, cited March 2000.