Analysis and Implications of Student Contact Patterns Derived from Campus Schedules

Vikram Srinivasan and Mehul Motani
Electrical & Computer Engineering
National University of Singapore
{elevs,motani}@nus.edu.sg

Wei Tsang Ooi
School of Computing
National University of Singapore
ooiwt@comp.nus.edu.sg

ABSTRACT

Characterizing mobility or contact patterns in a campus environment is of interest for a variety of reasons. Existing studies of these patterns can be classified into two basic approaches – model based and measurement based. The model based approach involves constructing a mathematical model to generate movement patterns while the measurement based approach measures locations and proximity of wireless devices to infer mobility patterns. In this paper, we take a completely different approach. First we obtain the class schedules and class rosters from a university-wide Intranet learning portal, and use this information to infer contacts made between students. The value of our approach is in the population size involved in the study, where contact patterns among 22341 students are analyzed. This paper presents the characteristics of these contact patterns, and explores how these patterns affect three scenarios. We first look at the characteristics from the DTN perspective, where we study inter-contact time and time distance between pairs of students. Next, we present how these characteristics impact the spread of mobile computer viruses, and show that viruses can spread to virtually the entire student population within a day. Finally, we consider aggregation of information from a large number of mobile, distributed sources, and demonstrate that the contact patterns can be exploited to design efficient aggregation algorithms, in which only a small number of nodes (less than 0.5%) is needed to aggregate a large fraction (over 90%) of the data.

Categories and Subject Descriptors

C.2.0 [**General**]: Wireless Networks

General Terms

Experimentation, Design, Performance

Keywords

Contact Patterns, Virus Spread, Delay Tolerant Networking, Mobile Social Software

MobiCom'06, September 23–26, 2006, Los Angeles, California, USA.
Copyright 2006 ACM 1-59593-286-0/06/0009 ...$5.00.

1. INTRODUCTION

Pervasiveness of mobile devices has lead to development of a new class of mobile applications which use proximity-based connectivity such as Bluetooth to exchange messages. Since such applications often exploit the social context of the users, the term MoSoSo, or mobile social software, has recently been coined to refer to this class of applications. Examples of MoSoSo include mobile applications that aim to expand one's social circle (so called BlueDating), by alerting a user if another user with similar interest profiles is nearby. Serendipity [7] and Nokia Sensor[1] are examples of BlueDating applications available currently. The academic community has proposed several other novel applications for Bluetooth connectivity on mobile phones too. Two examples are DTN and PeopleNet. DTN (e.g., Haggle [4]) studies how packets can be routed from a source to a destination using carry-and-forward strategy, allowing messages to be delivered in the absence of end-to-end connectivity. PeopleNet [14] allows a user to query for information provided by other users, by propagating the query to other mobile phones in the network to increase the probability that a matching answer can be found. These benevolent applications are not the only applications exploiting the Bluetooth connectivity between mobile phones. Malicious viruses can spread to nearby mobile phones through Bluetooth connectivity as well [6].

Whether malicious or benevolent, mobility and contact patterns of the mobile devices play a significant role in the performance, or even viability, of such applications. Such contact patterns not only help to accurately predict the performance of these applications, but also help to design superior algorithms based on the characteristics of these contact patterns.

Previous efforts to characterize mobility and contact patterns can be categorized into two basic approaches. The first approach involves constructing mathematical models (see reference [3] for a survey) to generate movement patterns of devices. The advantage of the model-based approach is its scalability – large number of devices can be modelled over a large time scale. These mobility models, however, have been shown to exhibit a different set of characteristics to those observed in the real world (as shown by Chaintreau et. al.[5]). The second approach measures locations and proximity of mobile devices to infer their mobility and contact patterns. Such measurements can be done through logging at wireless access points (e.g., traces from Dartmouth [9]), from which contact and mobility patterns can be inferred, or through mobile sensing devices carried around by participants of the measurement experiments (e.g., the iMotes experiment [4]). While the measure-

[1] http://www.nokia.com/sensor

ment approach allows real-world data to be collected, it is difficult to scale to large number of users. Often such efforts require cooperation from service providers [16] or large number of man-hours [9].

In this paper, we take a different approach to acquiring mobility and contact patterns, by exploiting a new data source – class time-tables and student rosters easily accessible from within our university. We extract time-tables and rosters from our university's Intranet learning portal, and infer mobility and contact patterns of the students on campus. Our approach not only gives us accurate contact patterns between students with relatively little effort, but it also allows contact patterns of a large population (the whole student population) to be studied over a long period (one academic semester). The population represents students from different colleges and schools, in different years of studies, exhibiting different behavioral patterns. Our trace complements measurement-based traces, providing long-term, periodic, contact patterns among students. Our approach augments existing trace driven studies because it allows us to ask hypothetical questions about the performance of DTN and similar applications. Measurements based studies represent a snapshot in time and are dependent on the penetration and usage behavior at the time of the study.

Based on our traces, we explore how these contact patterns impact mobile applications in three different scenarios. We first view these patterns from the perspective of delay-tolerant networks, using "store, carry, and forward" routing. Such routing algorithms have been studied in several contexts, including in wireless ad-hoc networks (e.g., [15]), and most recently in opportunistic networking [5]. Our analysis gives insights into the minimum achievable end-to-end delay in opportunistic routing and shows that a large percentage of students experience small inter-contact times with other students.

Our second scenario concerns the spread of mobile viruses in a campus setting. We investigate how fast a mobile virus spreads itself to other mobile devices through student contacts during classes. Assuming 100% probability of infecting a neighbor, our results show that a virus can spread at an alarming rate – 90% of the mobile devices belonging to the student population can be infected in 1-2 days, depending on the initial number of infected phones. If we consider contacts outside of classrooms, then even starting with a single infected phone and using a reasonable model of interactions outside classrooms, the virus can spread to 90% of the population in a day. We also identify the classes or sessions which contribute to the speed of spread of the virus.

Finally, we consider information gathering from a large number of distributed data sources and show how the characteristics of the contact patterns can be exploited to design efficient aggregation algorithms. We consider a scenario, where an aggregation query is issued to query information from a user population through mobile phones. For instance, consider a virtual used textbook marketplace running on top of PeopleNet [14]. One might want to find the average (or minimum) selling price of a book, before making buying decisions. We present an algorithm to perform the aggregation in a distributed and scalable manner. The characteristics of the contact patterns allow a small number of nodes to sample a large fraction (more than 90%) of the population for data. Thus only a small fraction of phones (less than 0.5%) need to respond with the aggregated information, avoiding implosion at the source.

We summarize our contributions as follows. First, we present the first large-scale, long term, mobility traces inferred from university class rosters and time tables. These traces are several orders of magnitude larger than any previously reported traces that directly measures contact patterns. We will make the anonymized traces publicly available to the research community. Second, we analyze the traces in the context of three mobile applications. In each of these applications, we present the impact of the contact patterns observed in our mobility traces on the applications. For delay-tolerant networking, we introduce the notion of *time distance* between students, and show that students have small average time distance between them, providing insights to the achievable end-to-end delay in routing. We utilize the mobility traces to ask *what if* questions and explore how fast a mobile virus can spread in a university campus under different conditions and infection parameters. We also investigate how a scalable aggregation algorithm can be designed exploiting the characteristics of the contact patterns, in the context of querying over distributed, mobile data sources.

The rest of this paper is organized as follows. Section 2 presents related work in the literature. Section 3 elaborates on the value of our data set, and presents the models and assumptions used in our study. The basic properties of the contact patterns are presented in Section 4. We study characteristics of contacts between students in the context of a DTN application in Section 5, and how malicious mobile virus can spread among the student population in Section 6. Section 7 describes a scalable, decentralized, aggregation algorithm to aggregate information stored on mobile devices of the students. Finally, we reflect on our findings and propose future directions of this study in Section 8.

2. RELATED WORK

In the past there have been two approaches adopted to understand contact patterns, a measurement-based approach and a model-based approach. This section reviews some related work in the literature on these approaches, and highlights the weaknesses of these existing approaches.

In the measurement-based approach, researchers conduct measurement studies and collect traces through logging access patterns and contact patterns. At UCSD [13] and Dartmouth [9], studies of client's association patterns with WiFi access points were conducted over several months. The UCSD study includes data from client-based logs. The study was conducted over 3 months. In the Dartmouth study, information was obtained from SNMP logs from access points. The study was conducted over four months. These traces have also been used to make inferences about general contact patterns (e.g. contact times, inter-contact times) between different users/people. However these inferences might be incorrect since these are based on the fact that two devices have simultaneously associated with the same access point. This association does not necessarily imply that the two devices are within range of each other. Jain et. al [11], derive an empirical model for user registration patterns in a campus wireless LAN derived from trace based studies.

To get more detailed proximity patterns, a study was conducted by Intel and Cambridge as part of the Haggle [4] project. They handed out iMotes to volunteers and asked them to carry these devices around for a few days. These devices were collected once the battery expired to read off data regarding proximity information to other users. In the largest experiment, around 40 devices were handed out to volunteers at INFOCOM 2005. No experiment was longer than 5 days. The time granularity of the iMotes study was 120 seconds.

These measurement-based studies are certainly useful in that they give us insight into current trends and evolution in how these technologies are used. However, these trace based studies are based on current penetration of wireless technologies and usage behaviors. Moreover, they might not be an accurate reflection of real contact patterns since they are based on the interactions between only a small subset of the population. They also do not allow us to ask *what if* questions. For example, if a new application such as DTN or PeopleNet is deployed, what would be the performance when the penetration rate is ρ? There is clearly a huge gap in the knowledge of real world contact patterns.

In the model based approach, researchers have developed simple and reasonable mobility models which have been used to characterize the performance of wireless ad hoc networks (see [3] and references therein). Mobility models such as the Random Way-point and the Random Direction model are commonly studied in ad hoc networking research. In these models however, nodes are typically assumed to move independently. This assumption results in inter-contact times that are exponential in nature. However as the trace driven studies [13, 9, 4] have shown, inter-contact times typically obey the power law distribution. Therefore these simple mobility models do not reflect real world mobility patterns. There have been attempts to model realistic campus mobility models by Jardosh et. al [12]. However it is not clear if these are valid models.

3. OUR DATA SETS AND MODEL

Our insight is that accurate information of human contact patterns is available in several special scenarios such as university campuses. If one knows the class schedules and enrollment of students for each class on a campus, it gives us extremely accurate information about contact patterns between students over large time scales. In this paper, we obtain this information about student enrollment and class schedules from our university. We first characterize the properties of these contact patterns and analyze their impact on the following three applications: *(i)* DTN, *(ii)* spread of mobile viruses and *(iii)* data aggregation from mobile data sources.

3.1 Data Collection

We collected information on class schedules and class rosters for the Spring 2006 semester in which there were 22341 enrolled. Our university, the National University of Singapore, has different colleges (e.g., Engineering, Science, Law) and within each college there are departments (e.g., Electrical and Computer Engineering, Computer Science). Every department offers graduate and undergraduate degrees, and face to face classes are an integral part of these programs. Many classes also have labs and recitations associated with them. For large classes, there are several recitation sessions offered and students sign up for the recitation session which is most convenient to them. The same goes for the labs. At this time of writing, all lessons are conducted on the main campus of the university at Kent Ridge, spanning an area of 146 hectares.

Our university has a central Intranet portal for teaching, called Integrated Virtual Learning Environment (IVLE). The Intranet portal hosts a web site for every class that is taught on campus. Professors manage the web site for their respective classes and post lecture notes, quizzes, solutions etc. on their class web site. Information about students enrolled and the schedule for the class is posted on the web site for each class. We wrote a Perl script to harvest this data . We anonymize the identity of the students using MD5. For each student we stored information about the classes he was registered for, the start and end time of the class and its venue.

3.2 Data Processing

How accurate and reliable is our data? Are there pieces of information missing? The answer is yes. For a few classes, there are inconsistencies in the way data is stored on the class web sites. For example the schedule information is not available. Large classes (e.g., > 500 students) have different lecture sessions and we do not have information on which lecture sessions these students have signed up for. Also, for a given class, we do not have information on which students have signed up for which recitation and laboratory. In the rest of this section, we will describe how we dealt with these issues.

As mentioned previously, for each class, we obtained the sessions associated with the class, and the students enrolled in the class. A session can be of a certain *type*, for instance, a lecture session, a recitation session or a laboratory session. A class can have multiple sessions of each type. Sessions of the same type can be grouped into a *session group*. For instance, a class may hold two lecture sessions (delivering different content) in a week for the same set of students. Both these lecture sessions are said to belong to the same session group. On the other hand, a class with large number of students, may hold two lecture sessions (delivering the same content) in a week for different batches of students. These lecture sessions are considered to be in different session groups. A student signs up for a session group for each type of session in a class he is enrolled in, and is expected to attend all sessions within that session group.

Our Intranet portal does not provide detailed information about which session group a student has signed up for. To fill in these details, we randomly assign a student to a session group. To be more specific, given a student s, for each class c that s has enrolled in, for each session type t of c, s randomly and independently signs up for a session group of type t, and attends all sessions of that session group.

Our random assignment of students to session groups might result in conflicts – that is, a student might have signed up for two sessions which are held at the same time. We adopt a simple approach to deal with such conflicts. If a session group assigned to a student leads to a conflict, the student is randomly assigned to another session group of the same type. If it is impossible to resolve a conflict, the student will not be attending any session group of that type. In our trace, only 3% of all assignments resulted in unresolved conflicts.

After both screen scraping [2] and session assignment, we have a view of which student is attending which session at what time. This data provides us with in-class activity of a student for a week. We further simplify the model in several ways. Firstly, most sessions start on the hour and end on the hour. For the few sessions which are not, we round up the starting time and ending time of the sessions to the nearest hour. This simplification allows us to use one hour as one unit time. Secondly, we “compress” the time by removing any idle time slots without any active sessions. For example, suppose the last session of Monday ends at 9pm, and the first session of Tuesday starts at 8am. If Monday 8pm to 9pm corresponds to the 10th hour, then Tuesday 8am to 9am is the 11th hour in our model. This concept is similar to business days, which counts the number of days excluding weekends and public holidays. We refer to our compressed time unit as a business hour. By compressing the time, we can remove any effects introduced by idle hours during the night and during weekends. For the rest of this paper, when we use the unit hours, we are referring to business hours. Finally, class activities which are held every fortnight are assumed to be held weekly for simplicity.

3.3 Models

The data we obtained from the Intranet portal gives us the session schedule of students, from which we can infer the contact patterns of students inside the classrooms. Students, however, are likely to come into contact with each other outside of class as well. For

[2] Screen scraping refers to the action of parsing data from a format usually meant for human consumption.

instance, at dining halls or libraries. The class schedules and rosters do not provide us with such information. To model such out-of-class contacts among students, we introduce *random hubs* into our model.

We assume that there are N_{hubs} random hubs on campus. At a given hour, if a student is not attending any session, with probability p_{hub} the student is in one of the random hubs. This probability p_{hub} is used to model the fact that, a student who is not in class might not be in contact with other students at all. If a student is in a random hub, he chooses one of the N_{hubs} hubs with uniform probability and stays in the hub for an hour. After an hour, the student, if still not attending any classes, decides again if he will be in a random hub, and if so, which one, independently from the previous hour. In our analysis, we set p_{hub} to 0.1 and N_{hubs} to 10. One could argue that the actual student population which is on campus on any given day is a small fraction of the total population and hence the random hubs will cause an artificial increase in the number of contacts. We found, however, that over 80% of the students had at least one lecture, lab or recitation scheduled on any given day. Our choice of N_{hubs} also closely mirror the large hubs on our campus. There are 11 large venues (5 dining halls and 6 libraries) on our campus where students can gather.

We can now describe how we infer the contact patterns among students inside classrooms and random hubs. The rule is simple – *two students are in contact with each other if and only if they are in the same venue at the same time*. In other words, we assume that as long as two students are in the same classroom, they are within Bluetooth range of each other. This assumption has been validated inside large classrooms on our campus. We also assume that two students who are in different classrooms are out of range of each other, even if one classroom is just next door to the other. We further assume that contacts take place only during business hours, and ignore that fact that students hang around campus for various activities after hours. We note that the last two assumptions are conservative – the number of contacts we obtained is a lower bound of the actual contacts that take place on campus.

3.4 Student Attendance Survey

The in-class contacts obtained through the rosters and schedules are on the assumption that students attend lessons all the time. This assumption is, of course, idealistic. Can we then easily take absentees into consideration when inferring the contact patterns? We conducted an online survey among the general student population in our university on class attendance patterns. The students were asked how regularly (at least 75% of the time) they attended:

1. sessions in the beginning, middle and the end of the semester.
2. early morning, mid morning and afternoon sessions.
3. small ($<$ 50 students), medium (50-100 students) and large ($>$100 students) sessions.
4. sessions with online webcasts.

The survey, gathered 2317 responses from the students, which is over 10% of the student population. The survey indicates that, in general, at least 64% attended their lessons regularly (except for early morning classes, where only 48% of the respondents attend regularly). The survey also found that, not surprisingly, the answer depends on the period during the semester, session size, and time of the day when the session is held. About 91% of respondents attend classes regularly at the beginning of semester. The number drops to 71% for the middle of semester and decreases further to 64% near the end of semester. For sessions with size less than 50, 84% of the respondents attend the lesson regularly, but only 64% do so for session larger than 100 students. For sessions with online webcasts, the attendance rate drops to 38%. This irregularity in attendance rate shows that, without attendance logs of individual sessions, we cannot easily model student attendance. Based on the traces we collected, one can, however, plug in different attendance model to study the effects under different what-if scenarios. We only present contact-patterns based on the assumption of full attendance in this paper, which gives us a bound on the actual in-class contact patterns.

The contact patterns among students that we obtained through the procedure above, give us *human contact patterns*. From these contact patterns, we can infer contact patterns between mobile devices and explore hypothetical questions about the performance of algorithms. We investigate these contact patterns in the rest of this paper.

4. FIRST LOOK AT THE DATA

To give the readers a feel of the data collected, we first present some basic properties of our data in this section. We will defer discussions on the impact of some of these properties to later sections.

Our data pulled from our Intranet portal gave us 4885 sessions, with an average number of 40.25 students in a session. The largest session has a session size of 615, and smallest session size is 1. 90% of the sessions have less than 85 students. Each student attends between 1 to 22 sessions a week, averaging 8.8 sessions a week. Figure 1 and Figure 2 show the cumulative distributions of the number of sessions attended by a student and the number of students in a session respectively. The figures show that the number of students attending a session has a highly skewed distribution, while the number of sessions attended by students is more evenly distributed.

We say a session is *active* at a time t if the session is being conducted at t. A student is active at time t if he is attending a session at t. Figures 3 and 4 show the distributions of the number of active sessions and active students at a given hour. The figures show that both the number of active sessions and active students are almost uniformly distributed, with an average of about 109 sessions (2%) and 4900 students (22%) active at a time.

One of the properties we are interested in is whether students tend to come into contact with the same group of students in the sessions they are attending, or whether they interact with different sets of students over time. To explore the interactivity among students, we construct two graphs and investigate the basic graph properties of these graphs.

Let the *session graph* G_{ses} be an undirected graph where the vertices are the sessions, and two sessions are connected by an edge if and only if they share at least one common student. Figure 5 shows the CDF for the degree of sessions in the G_{ses} constructed with our data. On average, a session shares common students with 155.4 other sessions. The most connected sessions shares students with 1602 other sessions. This distribution indicates that there is quite a bit of "mixing" occurring on our campus. Not only do there exist sessions with large size, there are sessions with students who attended many other sessions as well.

To see how well connected our students are, we construct another undirected graph called the *student graph* G_{stu} where the vertices are students, and two students are connected by an edge if and only if they attended at least one common session. The distribution of the degrees in student graph is shown in Figure 5. On average, a student shares a common session with 560.5 other unique students. The most connected student shares a common session with 2303 other unique students, almost 10% of the whole student population!

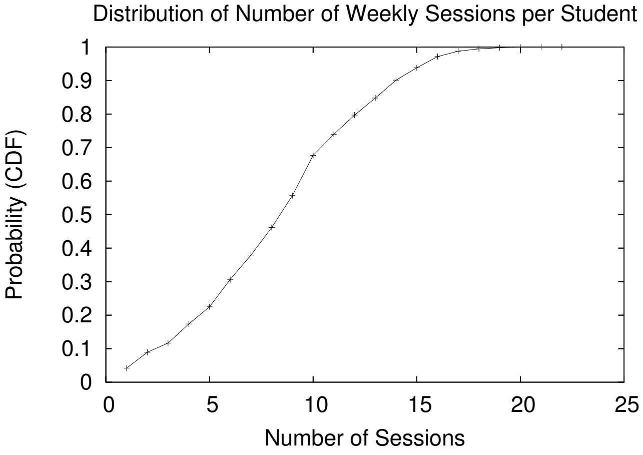

Figure 1: Number of Weekly Sessions per Student.

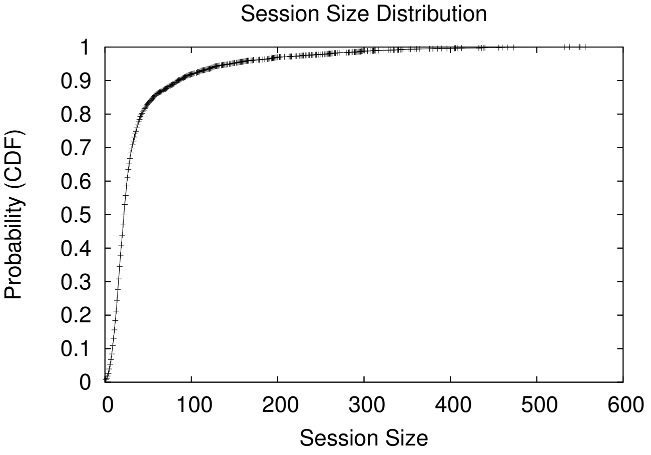

Figure 2: Sessions Size.

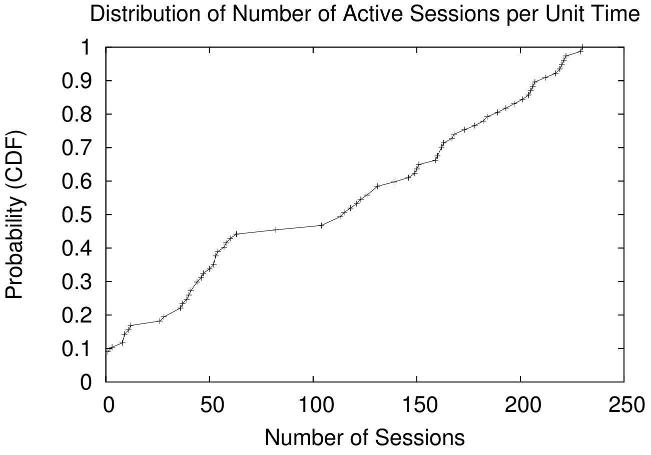

Figure 3: Number of Active Sessions per Unit Time.

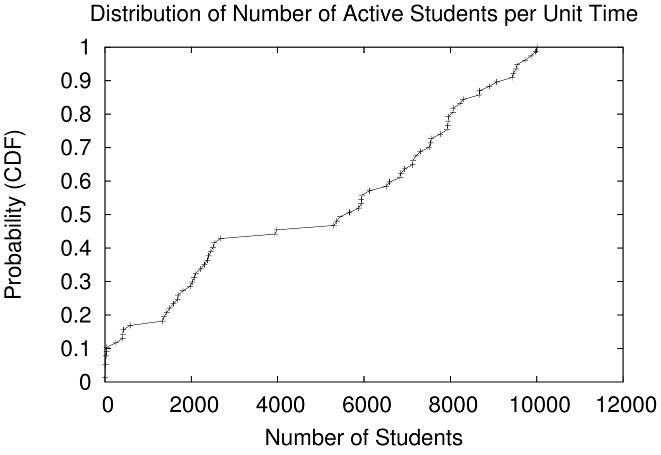

Figure 4: Number of Active Students per Unit Time.

Another interesting statistic we looked at is the *degree of separation* between students. We compute the length of the shortest path (called *hop distance*) between any two students in the student graph, and plot the distributions in Figure 6. We found that the diameter of the student graph is 8, *i.e.,* two students are connected by at most 8 hops, with an average hop distance of 2.45, a surprisingly small number. More than 90% of pairs of students are reachable within 3 hops of each other.

Our observation of small degrees of separation between students lead us to the question: do the student graph and session graph exhibit small world properties? Before we answer this question, let us first explain the notion of cluster coefficient and clustering index.

The *clustering coefficient* of node i measures how well the neighbors of i are connected between themselves. Let k_i be the degree of node i and let $G_i(v_i, e_i)$ be the subgraph of the neighbors of i. Then the clustering coefficient of node i is

$$CC_i = \frac{|e_i|}{(k_i)(k_i-1)/2}$$

The *clustering index* of the graph CC is the average clustering coefficient over all the nodes.

We compute the average path length $L(G)$ of a graph G as the average length of the shortest path between all pairs of nodes in G. Using average path length, and the clustering index of the graphs, we can identify small world characteristics. Like random graphs, small world networks have small average path length but unlike random graphs, they have a relatively large clustering index. Small world networks mirror social networks, in which tightly coupled cliques of friends are connected to each other by fairly few outgoing popular persons.

We compare our session graph to a random graph, in which we have a set of N nodes and each pair of nodes is connected with a certain probability. It is easy to show that the degree distribution of such a graph is binomial (and Poisson in the large graph limit). Let D be the average degree of nodes in the graph. Then average path length L_{rand} is given by

$$L_{rand} \approx \frac{ln(N)}{ln(D)}$$

and the clustering index CC_{rand} is $CC_{rand} \approx \frac{D}{N}$ [2].

Table 1 presents and compares the graph metrics of our session graph and student graph with random graphs with the same number of vertices and average number of edges. For both comparisons, we see that both our network and the random network have small av-

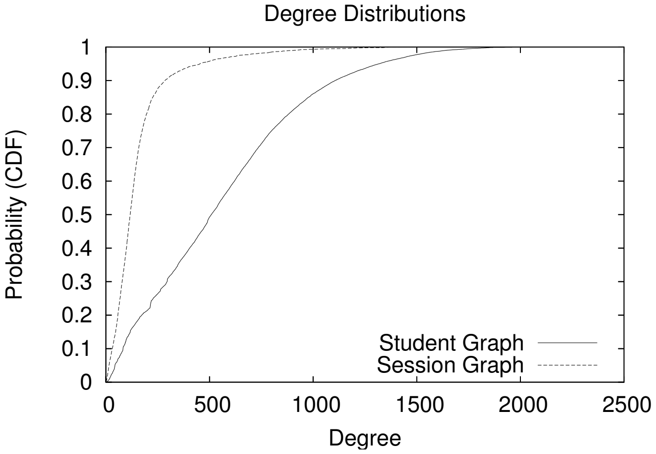

Figure 5: Degree Distributions in Session Graph and Student Graph.

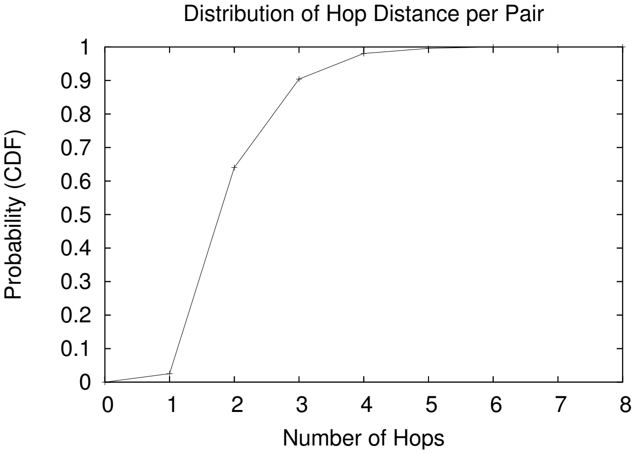

Figure 6: Distribution of Hop Distance Between Pair of Students.

	G_{ses}	G_{rand}	G_{stu}	G_{rand}
L	2.25	1.66	2.45	1.57
CC	0.34	0.03	0.53	0.026

Table 1: Small world characteristics of the session and student graphs.

erage path lengths but our graph has much larger clustering index. This result confirms that both the session and student graphs are small world networks.

This preliminary study on sessions and students leads to several general observations. Firstly, even when we consider contacts within classrooms, our student population is well mixed, with small degree of separation between them. This observation is good news for mobile applications that rely on proximity-based connectivity to disseminate messages (or viruses), as a potentially large number of contacts can be made. Secondly, our data shows that creating a mathematical model to model contact patterns might not be trivial, as complex interactions exist between time, sessions and students. These observations will be further elaborated in the subsequent sections, where the impact of the contact patterns from our traces on several mobile application scenarios will be studied.

5. DELAY TOLERANT NETWORKING

The first scenario that we look at is delay tolerant networks (DTN). DTN in general refers to networking applications that can tolerate large delays, operating without end-to-end connectivity. Examples of DTN include inter-planetary networks and vehicular-based networks in remote villages. The most relevant DTN in our context is what is termed *pocket switched network (PSN)* [4], in which opportunistic data transfer between two devices (PDAs, cellphones etc.) in close proximity is exploited to route packets from one user to another, in the absence of Internet connectivity. PSN allows store-carry-and-forward routing between devices, where a device might buffer and carry a packet on behalf of a source device, and forward the packet to the destination when the carrying device and destination device move within contact range of each other.

The mobility patterns of the devices, which in turn affect the contact patterns, have huge impact on the design of forwarding algorithms in PSN. Chaintreau et. al. [5] pointed out the weaknesses of using model-based mobility approaches (e.g. random way-point) to design forwarding algorithms, and showed that the characteristics found in six mobility traces they studied are different from those exhibited by the mobility models. Their mobility traces show that the tail distribution of inter-contact time between pairs of devices is heavy tailed, not light tailed as found in common mobility models.

In this section, we analyze the inter-contact time of our campus traces. First, we look at the case without random hubs. We measure the inter-contact time for a particular pair of students as follows. Let s and s' be two students. Suppose s and s' are in contact k number of times in a week, at starting time $t_0, t_1, ..., t_{k-1}$, where $t_{i+1} > t_i$. We let t_k be the time of the first contact in the following week, that is, $t_k = t_0 + T$ where T is the number of hours in a week. The sequence of k inter-contact times between s and s' is given by $\langle t_1 - t_0, t_2 - t_1, ..., t_k - t_{k-1}\rangle$. Inter-contact time tells us how frequently packets forwarding opportunities arise between these two students.

There are three significant differences between the inter-contact time inferred from our campus schedule and rosters, and those collected through mobility traces. Firstly, our contact durations are in the order of hours, since a contact is maintained for the whole duration of a session. Secondly, our contact patterns span over the whole academic semester (13 weeks), and involve slightly more than 22,000 students. This data is therefore several orders of magnitude larger than that reported by Chaintreau et. al [5], both in terms of duration of traces and population size. More importantly, we do not have the issues, where large inter-contact time close to the trace durations are likely not to be observed, and small inter-contact time smaller than granularity of measurements cannot be observed. Finally, the distribution of inter-contact time from our traces does not follow the power law distribution, as the inter-contact time for in-class contacts is bounded – students are expected to see each other at least once every week[3]!

Figure 7 shows the distribution for inter-contact time for all pairs of students, conditioned on contacts being made. The figure shows two curves. The solid curve (labelled "Time") plots the CDF of inter-contact time. This curve shows that more than 20% of the student pairs have an inter-contact time of 77 hours. This inter-contact time corresponds to the case where students meet each other only

[3] Assuming they have not overslept and missed class.

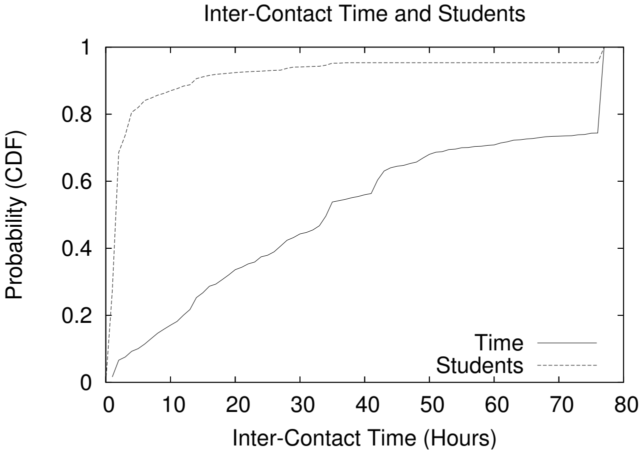

Figure 7: Distribution of Inter-Contact Time Between Pair of Students and Distribution of Students Experiencing at least One Contact of that Inter-Contact Time.

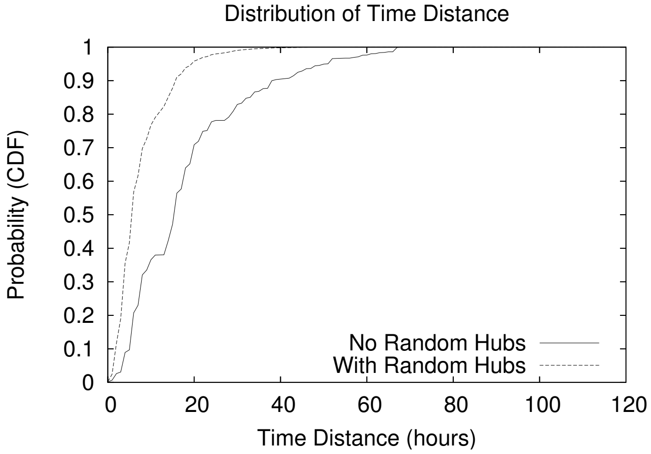

Figure 9: Distribution of Time Distance between Pair of Students with t = 0.

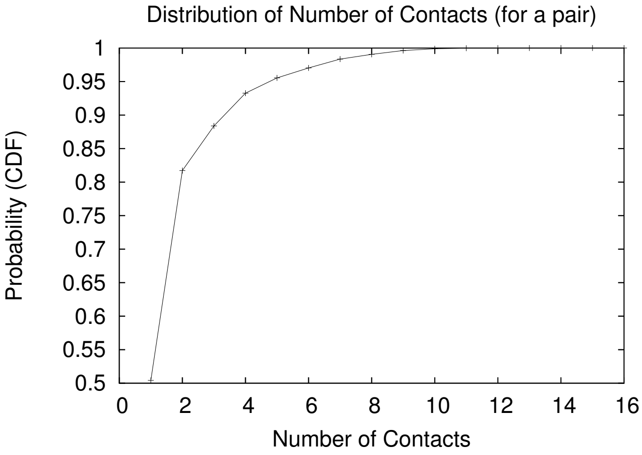

Figure 8: Distribution of Number of Contacts per Week.

once a week, most likely during a weekly lecture. There is also a slight increase in the number of inter-contact times around 35 hours and 42 hours. These bumps are due to the time-tabling policy in our university, in which lectures belonging to the same session group are typically arranged to be three days or four days apart. The average inter-contact time between any pair of students is 39.1 hours. While this average value seems large, our data shows that many of the students experience at least one inter-contact time that has much smaller values. The dotted line (labelled "Students") in Figure 7 plots the CDF of the number of students who have experienced at least one inter-contact time of a given value. From the figure, we can see that about 90% of the students have an inter-contact time of at most 10 hours with at least one other student.

Figure 8 shows the CDF for the number of contacts in a week, between any pair of students. The figure shows that less than half of all pairs never come into contact with each other and about 90% of all pairs come into contact with each other at most 3 times a week. The average number of weekly contacts between a pair of students is 1.9.

Another metric that is of interest to delay tolerant networking is the *time distance* between any pair of students. The time distance between two students s and s', with respect to a start time t_0, is the shortest possible time s can pass a message to s' (possibly through some intermediate students), given that s received (or created) the message at t_0. Time distance is interesting to look at, as it is a fundamental property of the contact patterns. For a given starting time and a given trace, time distance gives us a lower-bound on the delay between s and s', regardless of the forwarding algorithm used.

To compute the distribution of time distance, we first fix the source s and starting time t, and simulate flooding starting from s. At every hour, any student that has received the message originating from s, forwards it to all other students who are attending the same session but have not received the message. We record the time a message first reaches a student. After every student receives the message, or when no more new recipients are possible, we stop the current simulation. We repeat these steps for all possible sources s. This flooding procedure is similar to simulation of epidemic spread with infection probability of 1 (see Section 6).

Before we present the results, we note that, unlike hop distance, time distance cannot be computed by running all-pairs shortest path algorithm on a weighted graph G, where the weight of an edge is the shortest time distance between two neighboring students. The reason is that, the time that elapses before a student s can forward a message to a neighbor, depends on the time s receives the message.

The distribution of time distance for starting time $t = 0$ is shown as solid line in Figure 9. We found that starting time t does not have a significant effect on the distribution of time distance. Thus, only one value of t is shown here. The average time distance, between all pairs of students, is 18.8 hours, which is equivalent to about one and a half days.

Recall that up to now, we are only considering in-class contacts among students. Thus, the time distance computed is an upper bound to the actual time distance between two students, as students might interact outside of classrooms and speed up the flooding process. We introduce random hubs as described in Section 3 into our model, and recompute the time distance between all pairs of students. The dotted line in Figure 9 shows the new distribution, with average time distance of 7.9 hours. The random hubs have a significant effect – without the hubs, about 30% of all pairs have time distance of 20 hours or more. With random hubs, only 3% of all pairs of students have time distance of 20 hours or more.

6. DATA DISSEMINATION AND EPIDEMIC PROPAGATION

In this section we explore the epidemic spread of information and computer viruses. Recall that we assume that every student follows his/her assigned schedule and is present for every lecture, recitation and lab. We assume that every student has a mobile device that is capable of communicating in peer-to-peer mode and any two mobile devices that are in the same venue can communicate with each other directly with some probability p. We then analyze the spread dynamics.

We first study the spread of the epidemic by ignoring recovery. If at some time there is an infected node I which is co-located with several other nodes in a class, we assume that each of these other nodes *catches* the infection from I with some probability p. p could model a variety of factors such as attendance rate, or the probability of two devices in the same lecture theater being in communication range of each other. In fact it is reasonable to assume that $p = 1$. Even in our largest lecture halls we found that it was possible for Bluetooth phones to communicate with each other even when they are at the opposite ends of the lecture hall. In our simulation, we first pick a random time at which to seed S nodes with the infection. At each time step all infected nodes spread the infection to other nodes in their vicinity with some probability p. The results are then averaged over 100 simulation runs. We noticed that the results were independent of both the choice of seeds and the start time of the infection. First in Figure 10 we show the results when probability of infection is 1 as the number of initial seeds is varied. We also investigate how random hubs speed the dynamics of the spread. We do not see much difference in performance between 50 and 100 seeds. We see that even without random hubs, and with as little as one seed, more than 50% of the population gets infected within 15 hours. This time is only slight longer than on business day! With random hubs, the situation is much worse, 90% of the nodes get infected within 10 hours, irrespective of the number of seeds! In Figures 11 and 12, we keep the number of seeds fixed at 50 and investigate the effect of probabilistic infection. We see that the rate of spread slows down only below $p = 0.5$. Relating this back to our class attendance survey, this implies that as long as attendance rate is above 50%, the epidemic will spread quickly.

In our simulations, we chose nodes at random to seed and a random time to seed the nodes with an infection. We noticed that typically the speed of spread was almost independent of the nodes and times chosen. We now investigate what really contributes to the speed of spread of the epidemic. We use the metrics of centrality viz. closeness, betweenness and degree to investigate whether these measures affect the spread of epidemic.

In social networks terminology, the *closeness* measure of a node i is defined as $(\sum_j d_{ij})^{-1}$, where d_{ij} is the length of shortest path between node i and node j. Note that a larger closeness implies a more important node since it is on average closer to all other nodes. *Betweenness centrality* measures the extent to which a node is in between other sets of nodes. In other words, it measures how likely a node is in the shortest path between any pairs of nodes in the network. Let g_{ij} be the number of shortest paths from node i to j and $g_{ij}(k)$ be the number of shortest paths from i to j passing through k. Then the betweenness measure of node k is $\sum_{i \neq k \neq j} \frac{g_{ij}(k)}{g_{ij}}$. Finally the degree centrality of a node k is just the number of neighbors of k.

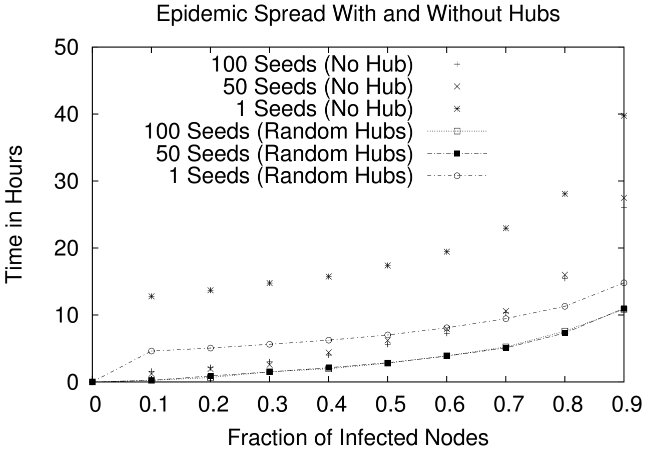

Figure 10: Spread of epidemic with and without random hubs as the number of seeds is varied from 1 to 50 to 100. For 50 and 100 seeds, we see no difference in the speed of spread. Clearly random hubs contribute significantly to the speed of spread.

We first compared the spread dynamics (we assume probability of infection = 1) for the three measures of centrality above to study whether these measures affect the spread of the epidemic. We did this by removing the top x most central sessions for each measure of centrality. We set x to be $20, 50$ and 100. These numbers represent around $0.5\%, 1\%$ and 2% of the total number of sessions. First we observed that irrespective of the centrality measure, the spread dynamics were almost identical conditioned on the top x central sessions being removed. Moreover we observed that removing the 20 most or 50 most central sessions did not affect the dynamics of spread, i.e., the spread dynamics was almost identical to having no sessions removed. A significant slow down in the spread dynamics was observed only when the 100 most central sessions were removed. To verify that it was the removal of the 100 most central sessions which contributed to the slow down in spread dynamics, we also removed 100 sessions at random and compared the dynamics. There was no difference in the spread dynamics with all sessions present and with 100 random sessions removed. The results can be seen in Figure 13. We only show results for the closeness measure, since the other measures result in almost identical performance.

We next model the epidemic spread via the standard SIR model available in literature [1]. In the SIR model, nodes can be in one of three states, S, I or R. A node which has never been infected, is said to be susceptible and is in state S. A node which is currently infected is in state I and a node which has recovered from the infection is in state R. A node which is in state R never goes back to states S or I. If every infected node has a finite probability of transiting to state R, then the epidemic eventually dies out. This standard SIR model has also been used to model the spread of Internet and mobile computer viruses [10]. In our model, in any time slot, a susceptible node which is co-located with an infected node can get infected with probability p_i. At the end of the time slot, a node which is in state I can transit to state R with probability p_r. We investigate the spread of virus under three scenarios[4], *(i)* $\frac{p_i}{p_r} = \frac{2}{3}$ *(ii)* $\frac{p_i}{p_r} = 1$ and *(iii)* $\frac{p_i}{p_r} = \frac{3}{2}$. In Table 2, we show

[4] There are of course several other possibilities, but we found these three cases representative of the dynamics of the system.

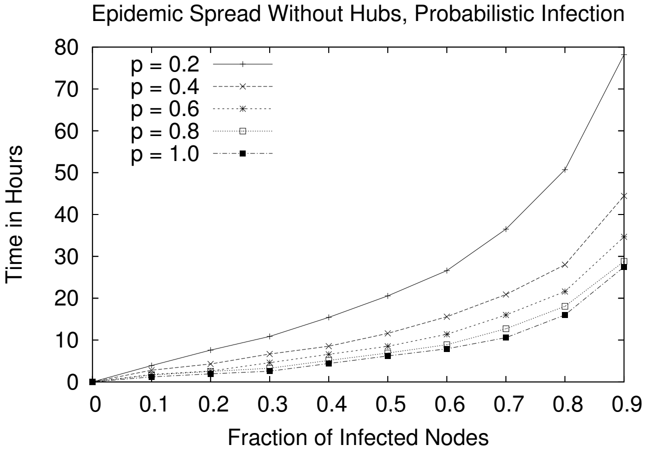

Figure 11: Dynamics of epidemics spread as the probability of infection is varied. We see that the spread dynamics slows down significantly for values of p less than 0.5.

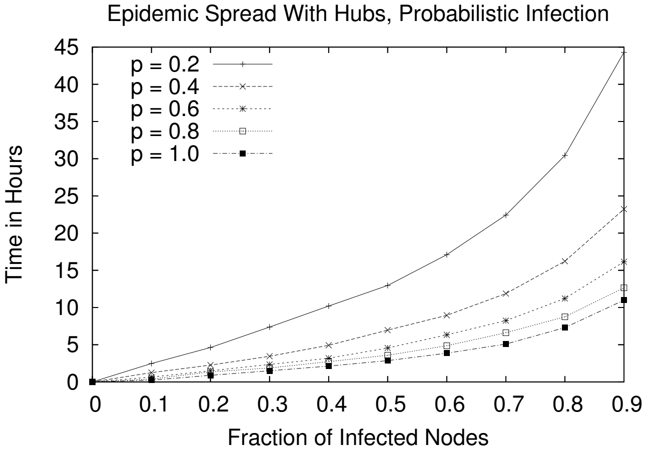

Figure 12: The effect of random hubs when the infection spread is probabilistic.

	No Hubs			With Hubs		
	$\rho = \frac{2}{3}$	$\rho = 1$	$\rho = \frac{3}{2}$	$\rho = \frac{2}{3}$	$\rho = 1$	$\rho = \frac{3}{2}$
Lifetime of Infection	58	73	84	178	180	183
Percentage Infected	54.55	66.91	80.23	99.62	99.95	100

Table 2: Infection Dynamics With the SIR Model.

how long it takes for the infection to die out and the total number of people infected. We see that the random hubs not only cause the infection to last much longer, but also cause almost 100% of the population to get infected.

7. DATA AGGREGATION

In this section we address the issue of how data can be aggregated from a large set of mobile phones. To motivate this problem, we present two scenarios where aggregation is useful.

The first scenario concerns virtual marketplace built on-top of PeopleNet [14], in which buyers and sellers enter their information on the phones, and let the information propagates through other phones, until a match is found. One could use PeopleNet to buy and sell textbooks. An aggregation query is useful, when say, a student wants to know the minimum selling price of a particular book. Another scenario could be a survey/opinion poll application. For instance, university management could send SMS messages to students polling for information (e.g. have you seen a certain suspicious character? or how long have you been waiting for the shuttle bus?) for security and planning reasons[5]. An aggregation query would be useful, if only aggregated information is needed (e.g. average waiting time) rather than individual answers from students.

Running aggregation query over large number of mobile nodes is non-trivial. There are three key mechanisms that need to be worked out. First, how do we get the queries to the mobile nodes? Second, how do the mobile nodes aggregate information for the query? Third, how do the mobile nodes relay this information back to a central data collection agency? For the first and third options, there are two alternatives. If we exploit the fact that mobile nodes have the ability to communicate over the infrastructure (e.g., cellular), then the query and aggregated information can be sent via the infrastructure. On the other hand, if we wish to exploit only the peer-to-peer capabilities of these devices, then we can use DTN like ideas, to opportunistically hand off data to conveniently located access-points whenever a mobile node is near one. We believe that the first solution is more robust and reliable. Therefore for sending queries and returning aggregated information, we exploit the infrastructure capabilities of the mobile devices.

Given our choice for transmitting and receiving aggregated information, we turn our attention to how mobile nodes should aggregate data. Notice that in our system model, there are energy and bandwidth constraints [6]. Clearly, the quickest method to query the database and receive responses is to broadcast the query over the infrastructure to all the nodes in the system and have them respond over the infrastructure. The bandwidth costs required for this is enormous. If one was tolerant to some delay in receiving responses however, a better strategy would be to transmit the query to a few nodes in the system and have them opportunistically aggregate information from neighboring nodes and relay the aggregated information after some fixed interval T. Notice that in addition to delay, we are also trading-off accuracy for bandwidth cost. A small subset of the nodes might only be able to aggregate information from a small fraction of the population. In what follows, we will see how the nature of our trace data can be exploited to efficiently aggregate information from a large fraction of the population in the shortest possible time, while still keeping the infrastructure costs low.

Assume that the query is initially sent over the infrastructure to some randomly chosen nodes. The quickest way (on an average) in which this query can be spread to all the nodes in the system is via an epidemic spreading model. In other words, a node with the query periodically broadcasts and spreads the query to other nodes in its vicinity. This will allow a large fraction of the population to be sampled very quickly. The key issue however, is how this

[5] This application is not as far fetched as it sounds, as our Intranet learning portal already allows a professor to send class announcements through SMS to all students in his class with a click of a button.

[6] Memory constraints are not so severe given the increasing memory capabilities of mobile devices such as cellular phones.

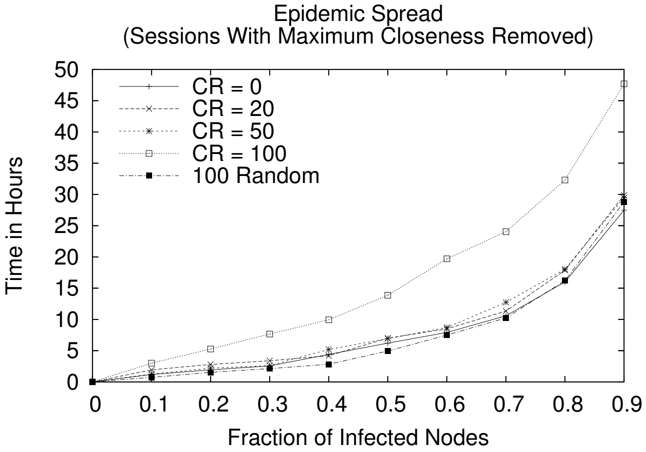

Figure 13: We look at the dynamics of epidemic spread with the top 20, 50 and 100 most close sessions removed. We compare the performance with no sessions removed and with 100 random sessions removed. In the legend CR represents the number of sessions removed.

information can be aggregated so that only a few nodes transmit aggregated information over the infrastructure with most of the sampled data. This is where the observation that in the campus model, nodes frequently aggregate in large clusters can be exploited. For simplicity, we consider only classes of queries where the size of aggregated data does not grow, with the amount of data aggregated (e.g. queries which wish to compute quantities such as AVERAGE, MIN and MAX). This assumption implies that the infrastructure bandwidth cost is only proportional to the number of aggregators responding and not the amount of data which is transmitted.

The idea behind our algorithm is simple. Every node which hears the query inherits the responsibility to aggregate. Next we ensure that in each aggregation center (e.g. in a class room, lab or random hub) only one node takes the responsibility of aggregating information. We show that this simple idea has some very nice properties.

Formally, our algorithm is as follows. At any time t, there are three sets A_t, R_t and S_t, such that $A_t \cup R_t \cup S_t = N$, where N is the set of mobile nodes. A_t is the set of nodes which have aggregated data at time t. R_t is the set of nodes which have received the query and taken the responsibility to aggregate data, but do not have any aggregated data at time t. S_t is the set of nodes which have not received the query up to time t. Let $a_t^i \in A_t$, $r_t^i \in R_t$ and $s_t^i \in S_t$ be the nodes in aggregation center (class room, lab etc.) i at time t. At time step t all nodes in $A_t \cup R_t$ broadcast their desire to aggregate information for query q. At aggregation center i, one of the nodes $n(i) \in a_t^i \cup r_t^i$ is elected [7] as the aggregator for that center and all other nodes in a_t^i relinquish the data that they have aggregated so far to $n(i)$. All nodes in s_t^i transmit their data to $n(i)$. Therefore if C is the set of aggregation centers, then, at time $t+1$, we have,

$$\begin{aligned}
a_{t+1^-}^i &= n(i) \quad \forall i \in C \qquad (1)\\
r_{t+1^-}^i &= \{a_t^i \cup r_t^i \cup s_t^i\} \setminus n(i) \quad \forall i \in C\\
s_{t+1^-}^i &= \phi \quad \forall i \in C\\
A_{t+1} &= \cup_{i\in C} a_{t+1^-}^i\\
R_{t+1} &= \cup_{i\in C} r_{t+1^-}^i\\
S_{t+1} &= N \setminus \{A_{t+1} \cup R_{t+1}\}
\end{aligned}$$

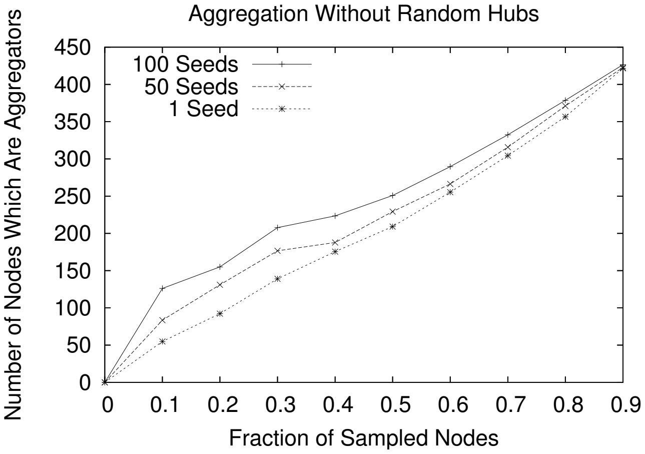

Figure 14: The figure shows the number of nodes which are aggregators, when a certain fraction of the node population has been sampled. The number of initial seeds is varied. We see that when over 90% of the population has been sampled, the number of aggregators is the same irrespective of the number of initial seeds.

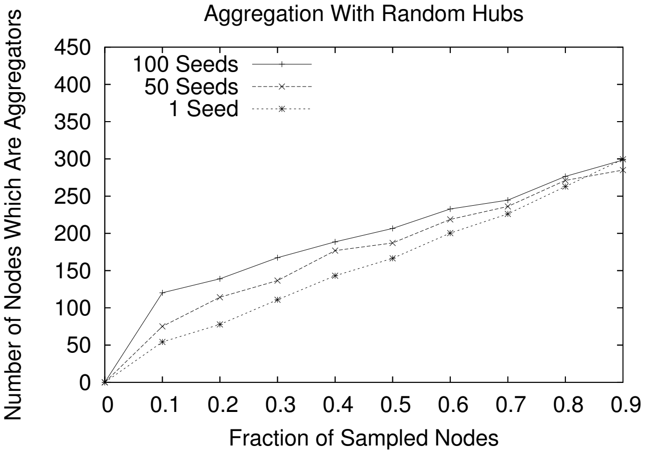

Figure 15: Same as Fig. 14, but with random hubs.

For simplicity, in our algorithm, we choose $n(i)$ uniformly at random from $a_t^i \cup r_t^i$. At the end of the deadline T for query q, nodes from A_T will transmit their aggregated information over the infrastructure.

[7] Dealing with the intricacy of distributed election protocol is out of the scope of this paper.

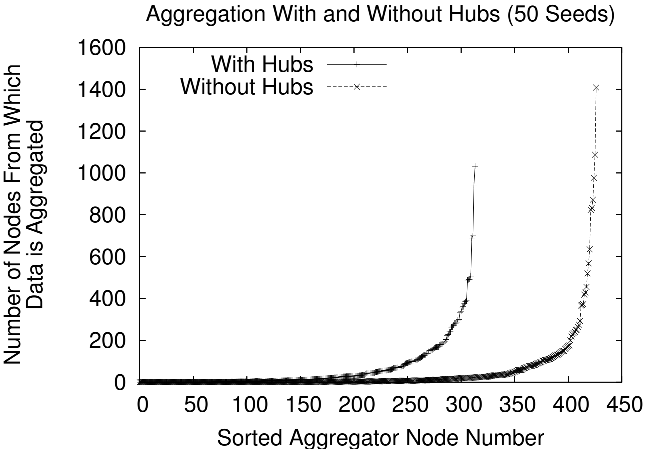

Figure 16: The figure shows the amount of information aggregated by each aggregator with 50 Seeds. The figure shows that a small number of the nodes have aggregated most of the information.

Let us look at some of the properties of the algorithm. First of all we see that the set of nodes from which each node in A_t has aggregated information is disjoint, i.e., if $n_i \in N$ is the set of nodes over which node $i \in A_t$ has aggregated data, then $n_i \cap n_j = \phi, \forall i \neq j$. Since we have only one aggregator at each aggregation center, and new information only comes from the set S_t, there is no duplication of information gathered by different aggregators. In other words there is no double counting. Nodes essentially store the query id q and transmit information for query q only once. Second, we see that given an initial number of seeds, any algorithm which relies on proximity based aggregation, cannot aggregate information from an equivalent number of nodes in shorter time. In other words, for any other algorithm, if the initial number of seeds is fixed, for any t, $|A_t \cup R_t|$ is maximized in our algorithm. This is due to the fact that the query spreads in an epidemic fashion in our algorithm.

The above properties are certainly useful. However, the key question is, how does the set A_t grow as a function of t? If it grows arbitrarily large, then every one of these nodes will respond with information over the infrastructure and bandwidth costs will be enormous. In Figures 14 and 15, we plot the number of nodes in the set A_t versus the fraction of sampled nodes in the system. We observe that irrespective of the number of seeds, the number of aggregators required to sample 90% of the nodes, is approximately the same. We also notice that the number of aggregators required to sample 90% of the nodes, without the random hub is about 430, while with random hubs, it drops to about 250. These numbers are not very large, but can they be made smaller? In order to gain some insight, we look at how much data each of these aggregators has, i.e., we compute $n_i \quad \forall i \in A_{t}^{*}$, where $t^* = \min\{t : \frac{|A_t \cup R_t|}{N} > 0.9\}$. We sort and plot n_i in Figure 16. We see that a few of the nodes have aggregated most of the data. Remarkably, we found that irrespective of the number of seeds, the largest 100 aggregators had aggregated over 90% of the data and the largest 50 aggregators had aggregated over 72% of the data. This points to a simple thresholding principle, whereby a node $i \in A_t$ transmits information only if $n_i > \tau$, where τ is some predefined threshold.

In addition, we also looked at the amount of overhead involved when nodes in a_i^t have to relinquish their data to node $n(i)$. In our simulations, we found that when 90% of the nodes have been sampled, the number of overhead transmissions is about 900 without hubs and 380 with hubs. We contend that this overhead is quite small.

To do a simple verification that it was this clustering model of students which resulted in a small number of nodes aggregating most of the data, we conducted the following control experiment. Given our trace session schedule, at each time slot, all students who are supposed to belong to one of the sessions are distributed uniformly between the sessions which are active at that time. This retains most of the dynamics of the system and only skews the distribution of students in sessions. When we ran this experiment with 50 seeds, we found that around 525 nodes were required to aggregate information from 90% of the nodes. Moreover, if we considered only the top 100 aggregators, they had aggregated only 70% of the data – a decrease of 25%.

It is clear that we would not see these properties in a system where nodes did not cluster and disperse at frequent intervals. We have been able to exploit this clustering phenomenon to obtain good performance from a simple yet intuitive algorithm. This algorithm can be further formalized into a protocol with several design parameters. However in this paper we only focus our attention on the key ideas of the algorithm. The main motivation was to show that the properties of our traces can result in simple yet powerful algorithms. The specific protocol design issues are beyond the scope of this paper.

8. REFLECTIONS AND CONCLUSIONS

In this section we reflect about the paper, its contributions, weaknesses and future directions. One of the main contributions of this paper is the source of data. In the past most attempts at deriving mobility patterns of users have been from analyzing trace measurements (typically in a campus environment) of user interactions with WiFi access points and other users. There are two possible reasons for this heavy reliance on measurement based data. One is the entrenched mind set in the research community which has been used to making measurements over the Internet to infer user behavior and system usage patterns. The other possible reason is that university campuses, out of privacy concerns, have been reluctant to part with the kind of information we have collected. Whatever the case, compared to existing literature in the mobile network research community, our approach and data are significantly different. In retrospect of course this method is a natural approach to obtaining such data. We highlight that our data is complementary to measurement based studies, which are dependent on existing applications and existing penetration of wireless devices. Measurement based studies are useful in understanding current usage patterns and trends in user behavior. However, they do not allow you to ask *what if* questions about new applications and algorithms. The research community has relied on simple, intuitive, but unrealistic synthetic mobility models to ask such questions. Our study presents a rich data source for the research community which can be used to study all kinds of applications and usage scenarios.

In addition to being an entirely new and complete data set, the other key aspect of our data is its size and time scale. We have collected data about the entire student population of $22,341$ students on campus. Measurement based studies, especially ones which depend on users carrying logging devices are necessarily limited in scope due to cost and manpower considerations.

Given the data set, we have first thoroughly characterized all its properties. We further show that it exhibits small world behavior.

We then focused on three aspects which we believe will have significant import on future mobile networking applications. The first is DTN. Terrestrially, DTN is a key technology which can not only bridge the digital divide, but will also be extremely useful in disaster recovery scenarios. We show that in a campus environment, arbitrary pairs of students can communicate with each other in less than two business days on average. This represents an upper bound on the performance of the system. We show that with random hubs present, this number reduces to 7.6 hours. Next, we consider the spread of mobile computer viruses on our trace data and show that a virus can spread to almost the entire campus remarkably quickly. We then identify the sessions which actually contribute to the speed with which these viruses spread. We then consider a new application. We view mobile devices such as phones as a large dynamic distributed database. We consider how data can be efficiently aggregated over such a database. We show that due to how students cluster on campus, data can be aggregated extremely efficiently with a few nodes aggregating data from almost the entire population. All our results represent upper bounds on what will happen in a real world scenario.

One observation we make is that generating an accurate synthetic mobility model derived from our trace data will be a very challenging task. Generating a synthetic model, all of whose properties are identical to the actual campus trace (even in terms of graph properties) is not very simple. A recent approach by Ghosh et. al [8] is possibly a step in the right direction. Their notion of sociological orbits and hubs, seems to reflect the campus traces at a high level. It remains to be seen, if algorithms which run on this model result in identical performance to our traces.

There are some weaknesses in our approach which we can identify. First, knowledge gaps exist during the times in which students are not active, i.e., in some session or the other. We use a simple random hub model to model how students behave when not in classes. In our model, students independently choose which hubs they will go to. This assumption is not true in real life. In the real world, people will probably go to a specific set of random hubs. The choice of random hubs could also be a function of the venue of the last session attended. For example, after a lecture which ends at noon, a student will probably head to the nearest cafeteria. These dependencies are not captured in our model. One could invest a lot time and effort concocting several other random hub models. However, in the absence of any other data to guide such choices, we chose to use the simplest reasonable model. The other place where our actions have influenced the model is that of assigning students to recitations and sub-sessions in large classes. We have assumed once again that students independently pick a recitation. However, in our experience with students on our campus, we found that students try to attend the same recitation groups as their friends or study partners. This dynamic is not captured by our data.

Finally, we realize that it may be difficult for other researchers to obtain access to similar data from other universities. Therefore we have made our complete and anonymized traces available at *http://www.comp.nus.edu.sg/ ooiwt/papers/contact-mobicom06-data/*. The traces are also available from CRAWDAD (*http://crawdad.cs.dartmouth.edu /*). We believe that there is much more that can be learnt from these traces.

Acknowledgements

We would like to thank Anirudh Natarajan and Arumugam Pillai Venkatesh for writing scripts to parse the data. We would like to thank the anonymous reviewers and our shepherd Jason Nieh for valuable comments which helped in improving the quality of the paper. This research was funded by grant R-263-000-331-112 from the National University of Singapore.

9. REFERENCES

[1] N. Bailey. *The Mathematical Model of Infectious Diseases and its Applications*. Hafner Press, February 1975.

[2] B. Bollobas. *Random Graphs*. Number 73 in Cambridge Studies in Advanced Mathematics. Cambridge University Press, Cambridge, UK, 2nd edition, January 2001.

[3] T. Camp, J. Boleng, and V. Davies. A survey of mobility models for ad hoc network research. *Wireless Communications and Mobile Computing (WCMC): Special issue on Mobile Ad Hoc Networking: Research, Trends and Applications*, 2(5):483–502, March 2002.

[4] A. Chaintreau, P. Hui, J. Crowcroft, C. Diot, R. Gass, and J. Scott. Pocket switched networks: Real world mobility and its consequences for opportunistic forwarding. In *Technical Report Number 617*, February 2005.

[5] A. Chaintreau, P. Hui, J. Crowcroft, C. Diot, R. Gass, and J. Scott. Impact of human mobility on the design of opportunistic forwarding algorithms. In *Proceedings of IEEE INFOCOM 2006*, Barcelona, Spain, April 2006.

[6] D. Dagon, T. Martin, and T. Starner. Mobile phones as computing devices: The viruses are coming! *IEEE Pervasive Computing*, 3(4):11–15, October 2004.

[7] N. Eagle and A. Pentland. Social serendipity: Mobilizing social software. *IEEE Pervasive Computing*, 4(2):28–34, 2005.

[8] J. Ghosh, S. J. Phillip, and C. Qiao. Sociological orbit aware location approximation and routing in MANETs. In *Proceedings of Broadnets 2005*, Boston, U.S.A, October 2005.

[9] T. Henderson, D. Kotz, and I. Abyzov. The changing usage of a mature campus-wide network. In *Proceedings of MobiCom 2004*, pages 22–31, September 2004.

[10] B. D. Noble J. W. Mickens. Modeling epidemic spreading in mobile environments. In *Fourth ACM Workshop on Wireless Security*, August 2005.

[11] R. Jain, D. Lelescu, and M. Balakrishnan. Model t: an empirical model for user registration patterns in a campus wireless lan. In *Proceedings of Mobicom 2005*, Cologne, Germany, August 2005.

[12] A. Jardosh, K.C. Almeroth E. M. B. Royer, and S. Suri. Towards realistic mobility models for mobile ad hoc networks. In *Proceedings of Mobicom 2003*, San Diego, U.S.A, September 2003.

[13] M. McNett and G. M. Voelker. Access and mobility of wireless PDA users. *Mobile Computing and Communications Review*, 9(2):40–55, 2005.

[14] M. Motani, V. Srinivasan, and P. Nuggenhalli. PeopleNet: Engineering a wireless virtual social network. In *Proceedings of MobiCom 2005*, Cologne, Germany, August 2005.

[15] G. Sharma and R. Mazumdar. Scaling laws for capacity and delay in wireless ad hoc networks with random mobility. In *Proceedings of IEEE ICC 2004*, volume 7, pages 3869–3873, Paris, France, June 2004.

[16] D. Tang and M. Baker. Analysis of a metropolitan-area wireless network. *Wireless Network*, 8(2/3):107–120, March 2002.