Bubble Rap: Forwarding in small world DTNs in ever decreasing circles

Pan Hui Jon Crowcroft

Abstract

In this paper we seek to improve understanding of the structure of human mobility, and to use this in the design of forwarding algorithms for Delay Tolerant Networks for the dissemination of data amongst mobile users.

Cooperation binds but also divides human society into communities. Members of the same community interact with each other preferentially. There is structure in human society. Within society and its communities, individuals have varying popularity. Some people are more popular and interact with more people than others; we may call them hubs. Popularity ranking is one facet of the population. In many physical networks, some nodes are more highly connected to each other than to the rest of the network. The set of such nodes are usually called clusters, communities, cohesive groups or modules. There is structure to social networking. Different metrics can be used such as information flow, Freeman betweenness, closeness and inference power, but for all of them, each node in the network can be assigned a global centrality value.

What can be inferred about individual popularity, and the structure of human society from measurements within a network? How can the local and global characteristics of the network be used practically for information dissemination? We present and evaluate a sequence of designs for forwarding algorithms for Pocket Switched Networks, culminating in Bubble, which exploit increasing levels of information about mobility and interaction.

1 Introduction

The first generation of human network models were probably the Erdős-Rényi random graphs [2]. More recently, heterogeneity has been introduced into models through the use of power-law and small-world graphs, especially in analysis of the AS-level of the Internet, for example in [4] [5]. This is the second generation of modeling. It is well known that some nodes may be more highly connected to each other than to the rest of the network. The set of such nodes are usually called clusters, communities, cohesive groups or modules. Many different approaches to community detection in complex networks have been proposed such as k-clique [28], betweenness [26], modularity [25] and more recently information theory [32]. Other kind of methods can be found in the survey paper [24]. Community detection can help us understand the local structure in mobility traces, and therefore help us design good strategies for information dissemination. It may be that communities detected from mobility data do not actually match well to real social communities, but still help with improved forwarding. [1]

[1] We will find out later that they actually match quite well.

The first goal of this research is to move to a third generation of human mobility models, understanding heterogeneity at multiple levels of detail.

Wireless networking has moved from a first generation of wireless access provided by 802.11 LANs and cellular services, through a second generation of Mobile Ad Hoc Networking, now on to a third generation: Pocket Switched Networks(PSN) [13] are a category of Delay Tolerant Network [8] aimed at supporting applications for human-to-human communications, through the so-called ferrying paradigm. Previous work [3] established the inter-contact intervals, and contact durations for a wide range of typical human mobility patterns and for a variety of today's radio devices. Critically, it was shown that stateless forwarding schemes would not provide a bounded expected mean delivery latency across such systems. On the other hand, flooding packets has a very high cost, not just in link-utilisation, but for other resources such as node storage and battery life, which are likely to be highly valued by users.

The second goal of this research is to devise efficient forwarding algorithms for PSNs which take advantage of both **a priori** *and learned knowledge of the structure of human mobility, to provide improved performance trade-off between delivery probability, latency and cost.*

Society naturally divides into communities according to needs for cooperation or selection. In sociology, the idea of "correlated interaction" is that an organism of a given type is be more likely to interact with another organism of a same type than with a randomly chosen member of the population [27]. If the correlated interaction concept applies, then our intuition is that using this community information to influence forwarding paths may be advantageous. To date, though, there have been few results to support this conjecture that we are aware of, except a very preliminary analysis by Hui et al. [14] on the use of as users' affiliation.

Searching using node degree rank was first introduced for peer-to-peer networks. Adamic et al. [1] describe a method for searching in networks, where the node degrees follow a power-law distribution, when the power law coefficient is sufficiently close to 2. Their strategy is to choose a node at each step with highest degree among all neighbors of the current node, quickly finding the highest degree node. Once the highest degree node has been visited, it will be avoided, and a node of approximately second highest degree will be chosen. Effectively, after a short initial climb, the search descends the degree sequence. The claim is that this is the most efficient way to do this kind of sequential search. This is a good incentive for us to look at this approach in PSNs as well. However, as we know, a PSN is very different from the Internet, which is largely fixed in structure. A PSN is a dynamic temporally varying network [17]; nodes move, connect and depart from time to time; the concept of degree is not simple to define. Is the degree of a node in a PSN the number of other nodes it has met in one second, one minute, one hour or one day? Why not 6 hours?

Freeman [10] defined several centrality metrics to measure the importance of a node to the network. "Betweenness" centrality measures the number of times a node falls on the shortest path between two other nodes. This concept is also valid in a temporal network. In a PSN, it can represent the importance of a node for relaying traffic for others in the system. Hence, we will look at whether the hierarchical search works with this centrality metric, and how to acquire the metric in a practical, decentralised way.

There are six specific contributions in this paper that progress towards our two top-level goals. First, we use the correlation of contact duration and number of contacts to classify human relationships in a PSN into four categories. Second, we use k-clique community detection algorithms on several real traces, to explore the nature of human community in different mobile environments. Third, we show empirically that identifying nodes according to their centrality or

ranking can improve delivery cost-effectiveness over a greedy approach. Fourth, we reconfirm the result of Hui et al. [14] that labelling increases the delivery cost-effectiveness, by using more reliable node selection. Fifth, we combine community and ranking together, making use of both local and global structures. This reduces the dead-end effect caused by global ranking, by forming a hybrid forwarding strategy, which improves over the delivery performance of naive multiple-copy-multiple-hop flooding schemes, but with much lower cost. Sixth, we use average unit-time degree to approximate centrality, and show that this achieves nearly the same performance as greedy ranking.

The rest of this paper is structured as follows. We briefly introduce the data-centric architecture and forwarding in Section 2, followed by a summary of the experimental data sets in in Section 3. Then in Section 4, we analysis contact graph distributions and use the correlation of contact duration and number of contacts to classify human relationships in a PSN into four categories. In Section 5, we shows the human heterogeneity from all the data sets. Next we use k-clique community detection algorithms on several real traces, to explore the nature of human community in different mobile environments. Subsequently, we show empirically that identifying nodes according to their centrality or ranking can improve delivery cost-effectiveness over a greedy approach in Section 8. We shows the result of direct "labeling" in Section 9 and the Bubble algorithm in Section 10. After that we present some early results of human predictability in Section 11. Finally we conclude the paper with a brief discussion.

2 Data-centric architecture and forwarding

Before moving into the main contributions of this paper, we want to first give a brief introduction about the data-centric architecture and forwarding paradigms for Pocket Switched Networks, which are related to this work.

Haggle architecture [36] is a data-centric clean slate designed for Pocket Switched Networks, where applications do not have to concern themselves with the mechanisms of transporting data to the right place, since that is what has made them infrastructure-dependent. By delegating to Haggle the task of propagating data, applications can automatically take advantage of any connection opportunities that arise, both local neighbourhoood opportunities and connectivity with servers on the Internet when available. Haggle is at a macro-scale comprised of six *Managers*, the Data, Name, Forwarding, Protocol, Connectivity and Resource Managers.

The data-centric principle of Haggle is that the data on each node in Haggle must be visible to and searchable for by other nodes (with appropriate security/access restrictions applied). In other words, relationships between application data units (e.g. a webpage and its embedded images) should be representable in Haggle, and applications should be able to search both locally and remotely for data objects matching particular useful characteristics. Haggle uses message switching, instead of package switching, in term of application-level data unit called Data Object (DO). A Data Object (DO) comprises many *attributes*, each of which is a pair consisting of a *type* and *value*. DOs can be linked into a directed graph to provides applications with a way to structure data, akin to the way that some applications use the placement of files in a common directory but more explicit and also for applications to link to the DOs which they require for their operation, which can be regarded as an "ownership claim." In the second way, many applications can claim the same DO, e.g. a photo gallery application can claim a photo that is linked to by a message (which brought it into the node) which is in turn claimed by the messaging application. Linking and claiming are accomplished using the same mechanism, we

use the two terms to differentiate between the parent being another DO or a different entity.

Considering forwarding in Pocket Switched Networks, while different applications have different network demands, we can summerize them into two categories: (a) *known-sender* where one node needs to transfer data to a userdefined destination. The destination may be another user (who may own many nodes), all users in a certain place, users with a certain role (e.g. police), etc. (b) *known-recipient* in which a device requires data of some sort, e.g. the current news. The source for this data can be any node which is reachable using any of the three connectivity types, including via infrastructure (e.g. a news webpage), neighbours (e.g. a recent cache of a news webpage) or mobility (e.g. the arrival of a mobile node carrying suitable data). We can see now that the communication paradigm is not only one-to-one(point-to-point), but also one-to-many, many-to-one, and many-to-many. The location doesn't matter but the name matters. In this sense, PSN is more than one kind of DTN to solve intermittently connection problem. It also represents a fundamental shift in the paradigm of networking, as fundamental as that was from telephony to IP. It tells us two points here: 1) communication is about data, not connection, not endpoints, and not path, 2) the killer application is multiple communication and sharing data, not one-to-one talk. This is similar to Van Jacobson's content-centric networking concept in his Google Tech talk 2006 [15].

We can see that we need a completely new paradigm to consider forwarding in this new communication model. In this paper, we look at two human social structures, community and centrality, which are very important for the data-centric forwarding. For example, the community concept would cover all the both communication paradigms, from one community member to another community member is one-to-one, from one member to a whole community is one-to-many, from one whole community to one member is many-to-one, and from a community to another community is many-to-many. And because we don't know the location of the recipient or even we don't know who is the recipient, then we need some other ways, instead of measuring topological distance, to help us to move the data outward, hubs or high centrality nodes are good choices. But for better focus, we will not mention about data-centric concepts in further texts and will only focus on one-to-one communication in this paper as a starting point and foundation for more advanced data sharing.

3 Experimental data sets

We use 4 experimental data sets gathered by the Haggle project for a period of 2 years referred to as *Hong Kong*, *Cambridge*, *Infocom05*, *Infocom06*, and one other dataset from the MIT Reality Mining Project [7], referred to as *Reality*. Previously the characteristics of these datasets such as inter-contact and contact distribution, have been explored in several studies [3] [13] [19], to which we refer the reader for further background information.

- In *Hong Kong*, the people carrying the wireless devices were chosen independently in a Hong Kong bar, to avoid any particular social relationship between them. These people have been invited to come back to the same bar after a week. They are unlikely to see each other during the experiment.

- In *Cambridge*, the iMotes were distributed mainly to two groups of students from University of Cambridge Computer Laboratory, specifically undergraduate year1 and year2 students, and also some PhD and Masters students. In addition to this, a number of stationary nodes were deployed in various locations that is expected many people to visit,

such as grocery stores, pubs, market places, and shopping centers in and around the city of Cambridge, UK. However, the data of these stationary iMotes will not be used in this paper. This dataset covers 11 days.

- In *Infocom05*, the devices were distributed to approximately fifty students attending the Infocom student workshop. Participants belong to different social communities (depending on their country of origin, research topic, etc.). However, they all attended the same event for 4 consecutive days and most of them stayed in the same hotel and attended the same sections (note, though, that Infocom is a multi-track conference).

- In *Infocom06*, the scenario was very similar to *Infocom05* except that the scale is larger, with 80 participants. Participants were selected so that 34 out of 80 form 4 subgroups by academic affiliations. In addition, 20 more long range iMotes were deployed at several places in the conference site to act as access points. However, the data from these fixed nodes is also not used in this paper.

- In *Reality*, 100 smart phones were deployed to students and staff at MIT over a period of 9 months. These phones were running software that logged contacts with other Bluetooth enabled devices by doing Bluetooth device discovery every five minutes, as well as logging information about the cellular tower they are associated with (a total of 31545 different towers were logged).

The five experiments are summarised in Table 1.

Experimental data set	Infocom05	Hong Kong	Cambridge	Infocom06	RealityMining
Device	iMote	iMote	iMote	iMote	Phone
Network type	Bluetooth	Bluetooth	Bluetooth	Bluetooth	Bluetooth
Duration (days)	3	5	11	3	246
Granularity (seconds)	120	120	600	120	300
Number of Experimental Devices	41	37	54	98	97
Number of internal contacts	22,459	560	10,873	191,336	54,667
Average # Contacts/pair/day	4.6	0.084	0.345	6.7	0.024
Number of External Devices	264	868	11,357	14,036	NA
Number of external contacts	1,173	2,507	30,714	63,244	NA

Table 1: Characteristics of the five experimental data sets

4 Contact graphs

Our first contribution is to introduce the notion of "contact graph" as a way to help represent the mobility traces, and to choose a threshold for community detection. The way we convert human mobility traces into weighted contact graphs is based on the number of contacts and the contact duration, although we could use other metrics. The nodes of the graphs are the physical nodes from the traces, and the edges are the contacts. The weight of the edges are the values based on the metrics specified such as the number of contacts during the experiment.

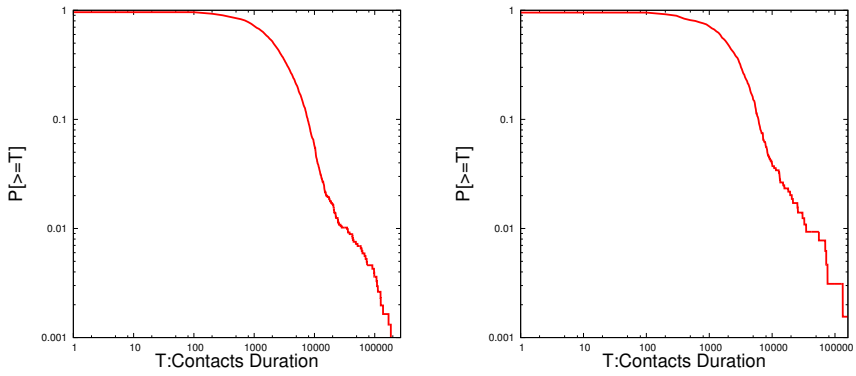
We measure the relationship between two people by how many times they meet each other and also how long they stay with each. We naturally think that if two people spend more time

together or see each other more often, they are in closer relationship. In this work we are not going to provide a specific threshold to infer actual social context: we just use these two metrics to produce some maps which may prove useful to guide forwarding.

Here we explore further properties of the experimental scenarios, and present statistics concerning the contact graphs for each dataset.

4.1 Weight distribution of contact graphs

First we would show that the statistical properties for the two conference scenario are quite similar. Figure 1(a) and 1(b) show the contact duration distribution for Infocom06 and Infocom05



(a) Infocom06 (b) Infocom05

Figure 1: Contact duration distribution for Infocom06 and Infocom05

respectively. We can see that their distributions are quite similar, with a mean different as small as 0.0003(0, 0.0633). More similarities will be seen in the next section as well. Because of space limitation, and these similarities, the later sections we only selectively show one as example, in most cases Infocom06, since it contains more participants, We show more results in a separate technical report.

Figure 2 and Figure 3 show the contact duration and number of contacts distribution for each pair in four experiments. For the HongKong experiment we include the external devices, but for other three experiments we use only the internal devices. We show later that for HongKong experiments we need to use the external devices to help to forward the data because of network sparseness.

4.2 Correlation between regularity and familiarity

We assume contact duration indicates familiarity. Two people sharing the same office might hate each other, and not talk, but we will ignore this kind of extreme situation here. The number of times two people meet each other implicitly reveals the pattern with which they meet. In this work, we infer regularity of meetings from the number of contacts. Two people might meet a lot of times in a short period (e.g. a day), and then not at all. However, short periods with many contacts are less likely to contribute to the upper quarters of the distribution, and here we will ignore these too as outliers.

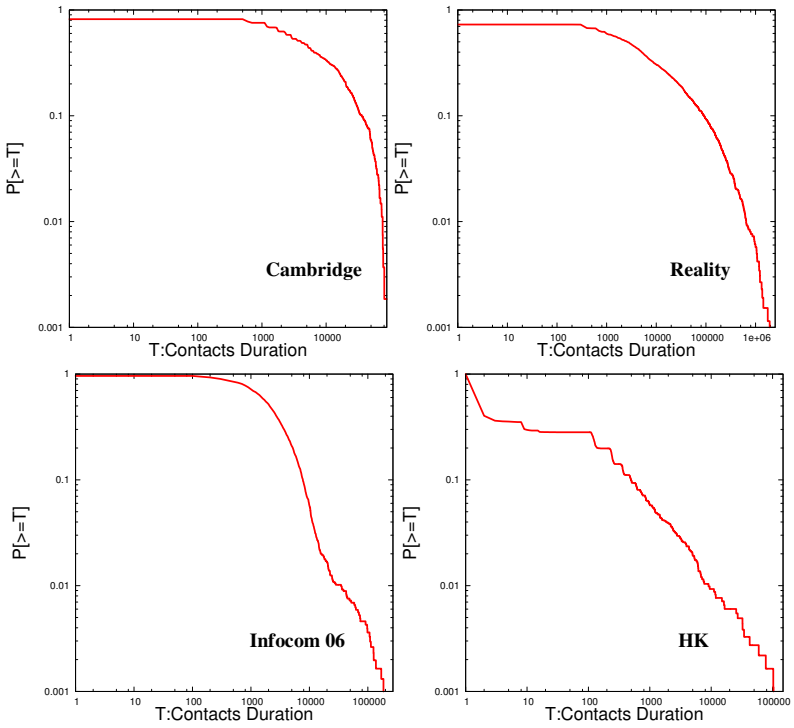


Figure 2: The contact duration distribution for each pair in four experiments.

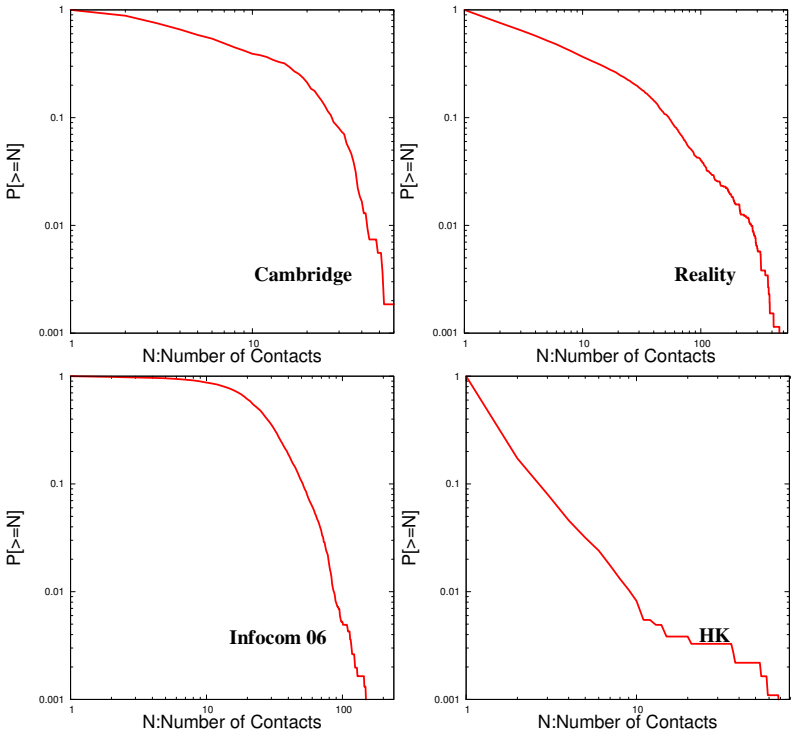


Figure 3: The number of contacts distribution for each pair in four experiments.

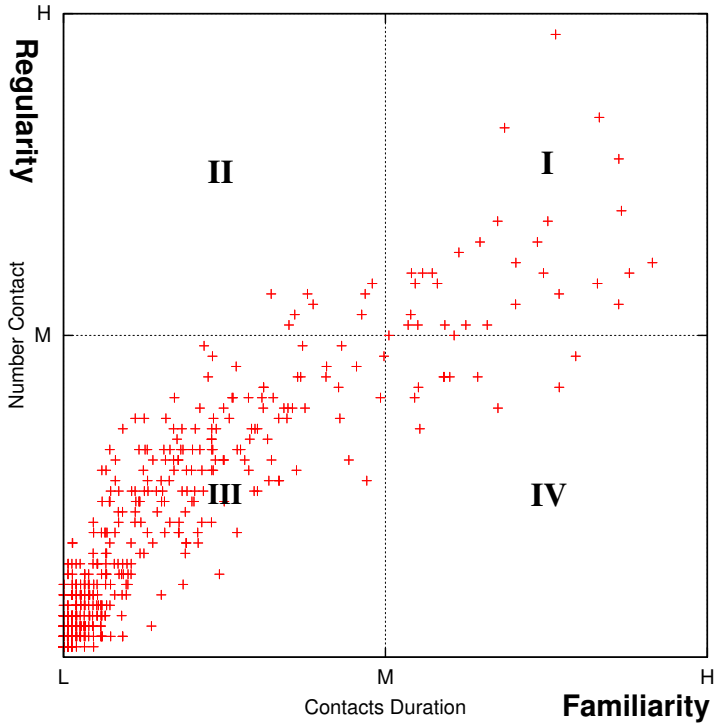


Figure 4: Number of contacts versus the contact durations for pairs of Cambridge Students.

Figure 4 shows the correlation between regularity and familiarity in the Cambridge data set. Here the regularity is positively correlated to the familiarity with a correlation coefficient of 0.9026. We define four kinds of relationships between a pair of nodes: Community, Familiar Strangers, Strangers, and Friends. A pair of nodes which has long contact duration (high familiarity) and large number of contacts (high regularity) is likely to belong to the same community. A pair of nodes which meet regularly but don't spend time with each other, could be familiar strangers [29] meeting everyday. People who don't meet regularly and don't spend time with each other would be in the category of strangers. Finally, for node pairs which don't meet very frequently but spend quite a lot of time together for each meeting, we count as friends. It is not necessary that the division of the four quarters are exactly at the middle. It is here acting as a reference or example. A clear cut division may need more empirical experimental results. But here we provide the methodology to classify these four kind of relationship based on pure contact duration and frequency. Additional difficulties faced by empirical social network research are well described in work by Watts [39].

Figure 5 shows the correlation between the number of contacts and contact durations for the other four experiments. We can see that conference environments are quite similar, both with a narrow stripe in the left bottom quarter. This stripe shows that people in the conference tend to meet each other more often than spend long time together. That is typical conference scenario, since people may meet each other many times in coffee breaks, corridors, registration desk etc. They may stand together and chat for a while, and then shift to chat with others instead of spending all the times together. *Infocom06* contains double the number of participants, and hence more data points. The *Reality* set is similar to the *Cambridge* one, with most of the points

lying on or above the diagonal line. However, it also seems that people have more contacts instead of spending times together. In the *HongKong* figure, we can find two pairs of friends, two pairs of close community members, and two pairs of familiar strangers. All the other pairs lie in the strangers quarter. This is in line with our expectations for an experiment designed to contain little social correlation.

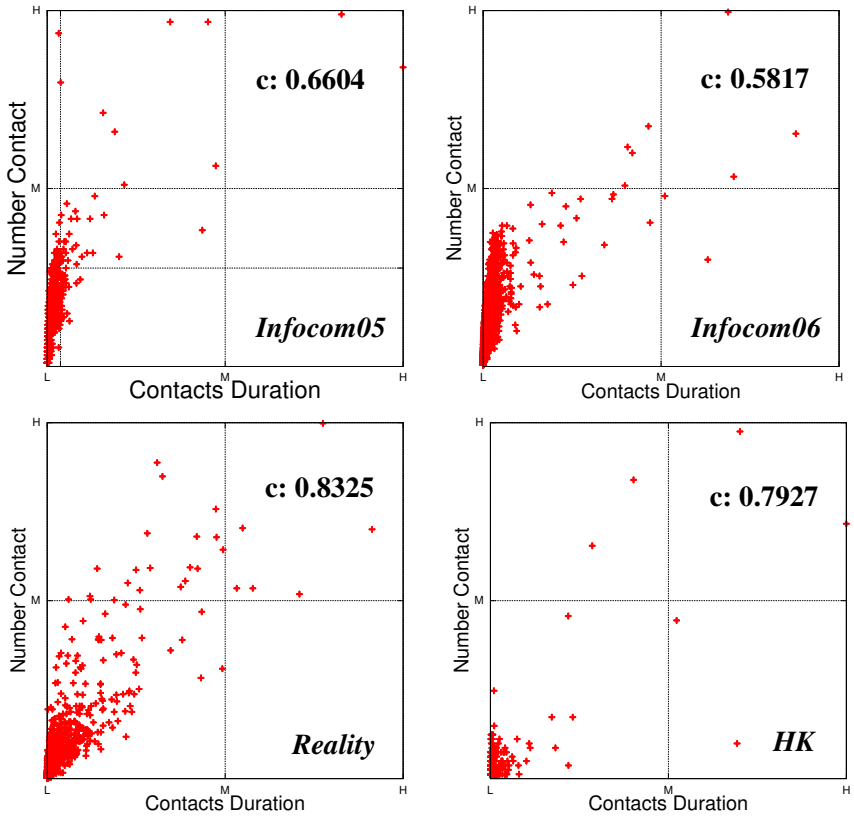


Figure 5: Number of contacts against contact durations for all pairs in the four datasets, with correlation coefficient.

5 On human heterogeneity

In many mobility models such as the random way-point, nodes are assumed, explicitly or implicitly, to have homogeneous speed distributions, importance and popularity. Our intuition is that the last two assumptions, at least, are not true. People have different levels of popularity: salesmen and politicians meet customers frequently, whereas computer scientists may only meet a few of their colleagues once a year. Homogeneity might favour different forwarding strategies for PSNs. In contrast, we want to employ heterogeneous popularity to help design more efficient forwarding strategies: we prefer to choose popular hubs as relays rather than unpopular ones. To date we are not aware of any empirical evidence for using human popularity or node centrality for information dissemination in mobile networks.

A temporal network is a kind of weighted network. The centrality measure in the traditional weighted network may not work here since the edges are not necessary concurrent. Hence we

need a different way to calculate the centrality of each node in the system. Our approach is as follows: First we carried out a large number of emulations of unlimited flooding with different uniformly distributed traffic patterns created using the *HaggleSim* emulator.

Then we count the number of times a node acts as a relay for other nodes on all the shortest delay deliveries. Here the shortest delay delivery refers to the case when a same message is delivered to the destination through different paths, where we only count the delivery with the shortest delay. We call this number the "betweenness centrality" of this node in this temporal graph[2]. Of course, we can normalize it to the highest value found. Here we use unlimited flooding since it can explore the largest range of delivery alternatives with the shortest delay. We believe that this definition is similar in spirit to the definition of the Freeman centrality [10].

Initially, we only consider a homogeneous communications pattern, in the sense that every destination is equality likely, and we do not weight the traffic matrix by locality. We then calculate the global centrality value for the whole homogeneous system. Later, we will analyze the heterogeneous system, once we have understood the community structure.

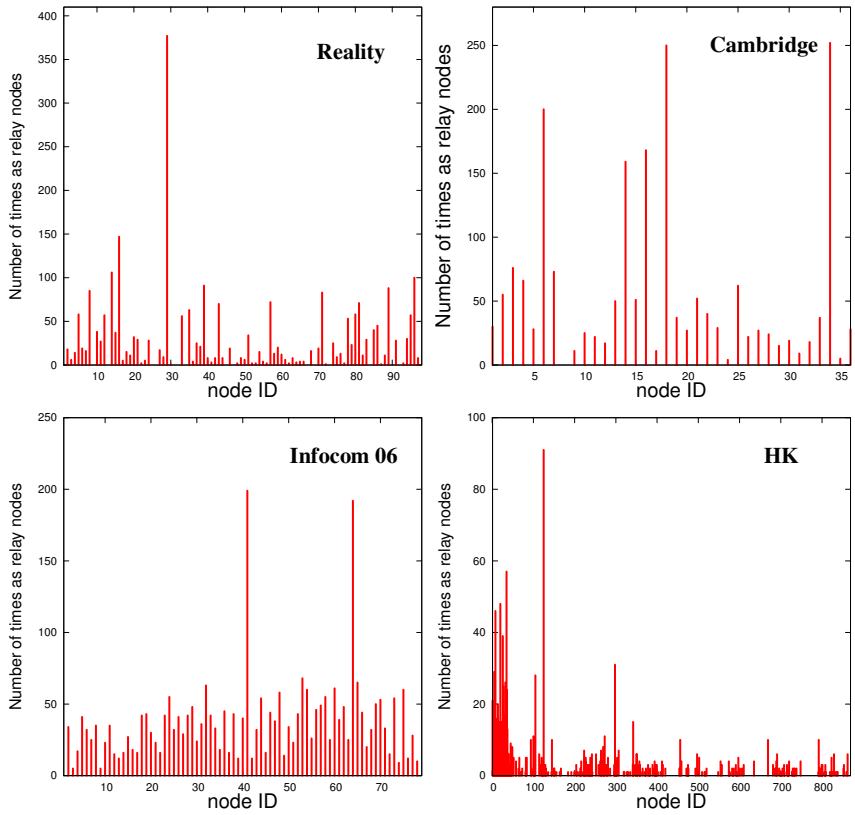


Figure 6: Number of times a node as relays for others on four datasets.

Figure 6 shows the number of times a node fall on the shortest paths between all other node pairs. We can simply treat this as the centrality of a node in the system. We observed a very wide heterogeneity in each experiment. This clearly shows that there is a small number of nodes which have extremely high relaying ability , and a large number of nodes have moderate

[2]We have calculated the weighted node centrality for each node, but found out that the weighted centrality is not well correlated to the centrality on the temporal graph. Nodes have very high weighted centrality may have very low temporal centrality.

or low centrality values, across all experiments. One interesting point from the HK data is that the node showing highest delivery power in the figure is actually an external node. This node could be some very popular hub for the whole city, i.e. postman or a newspaper man in a popular underground station, which relayed a certain amount of cross city traffic. The 30, 70 percentiles and the means of normalized individual node centrality are shown in Table 2 and the distributions are show in Figure 7.

Experimental data set	30 percentile	Mean	70 percentile
Cambridge	0.052	0.220	0.194
Reality	0.005	0.070	0.050
Infocom06	0.121	0.188	0.221
Hong Kong	0	0.017	0

Table 2: Statics about normalized node centrality in 4 experiments

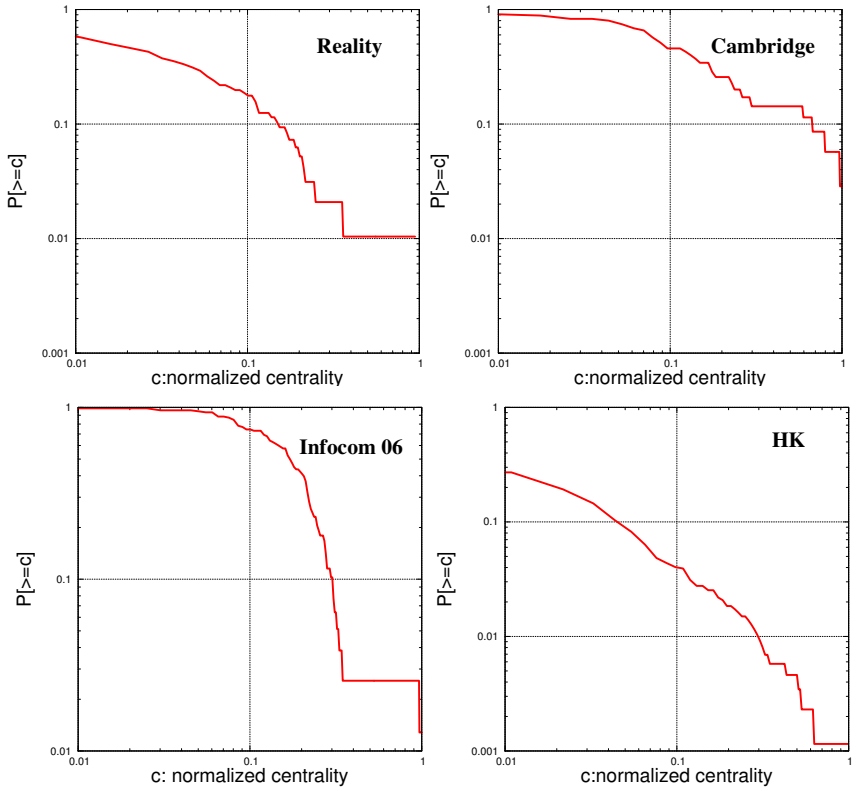


Figure 7: Distribution of normalized node centrality on four datasets.

6 Finding k-clique communities

Our second contribution is the identification of community structures using k-cliques. We have calculated all the results by using both contact duration and number of contacts on all five

experiments but because of space limitations we just show two cases of contact duration and two cases of number of contacts.

6.1 k-clique community detection

We use the k-clique community algorithm proposed by Palla et al. [28] in their work, since overlapping of communities are allowed, and we believe that in human society one person may belong to multiple communities. They define a k-clique community as a union of all k-cliques (complete subgraphs of size k) that can be reached from each other through a series of adjacent k-cliques, where two k-cliques are said to be adjacent if they share k-1 nodes. Their definition is based on their observation that an essential feature of a community is that its members can be reached through well-connected subsets of nodes, and that there could be other parts of the whole network that are not reachable from a particular k-clique, but they potentially contain further k-clique communities.

To illustrate this further, the k-clique-communities of a network at $k = 2$ are equivalent to the connected components, since a 2-clique is simply an edge and a 2-clique-community is the union of those edges that can be reached from each other through a series of shared nodes. Similarly, a 3-clique-community is given by the union of triangles that can be reached from one another through a series of shared edges. As k is increased, the k-clique-communities shrink, but on the other hand become more cohesive since their member nodes have to be part of at least one k-clique. The method is used for a binary network, and a weighted network is turned into binary network by setting a threshold.

6.2 k-clique university communities

In the visualization, an edge is added between two nodes if they are direct neighbors to each other in the community. The length of the edges is not proportional to any property of either the communities or the nodes. However the width of the edges is proportional to the link-weight that is the number of shared nodes between the two communities.

Figure 8 shows the k-clique communities detected from the Cambridge student data using number of contacts.

The duration of the experiment is 11 days. For the number of contacts, we used a threshold of 29 contacts, which represents an average of 3 contacts per day.[3] In this case, around 8.5% of all the edges are taken into account. We observe that the nodes mainly split into two communities of size 11 respectively with k as high as 10. Next we examine lower values of k. We can see also from Figure 8, when $k = 3$ there is a big community of 31 nodes, and when $k = 4$ the big community splits into two overlapping communities of sizes 14 and 17 with overlapping size = 1, and when $k = 5$ the two overlapping communities split into two disjoint communities of size 14 and 16 respectively. The two disjoint community structures stay visible until $k = 11$, with a gradual decrease in the community size. For the contact duration metric, we set the contact duration threshold to be 10 hours for the whole 11 days of experiment. We also observe mainly two communities when using this metric. The membership of these two communities is more or less the same as that when using the number of contacts metric. This agrees with Figure 4 that the contact duration and number of contacts for Cambridge data is highly correlated.

[3]Considering some students may be taking the same courses, be in the same supervision group, and live in the same College, and hence using same dining hall, this value is reasonable.

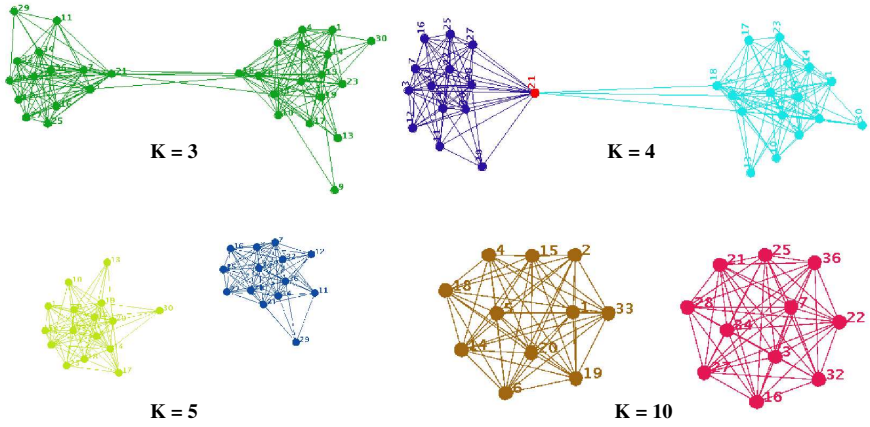


Figure 8: Communities based on number of contacts with weight threshold =29, k=3,4,5, and 10 (Cambridge).

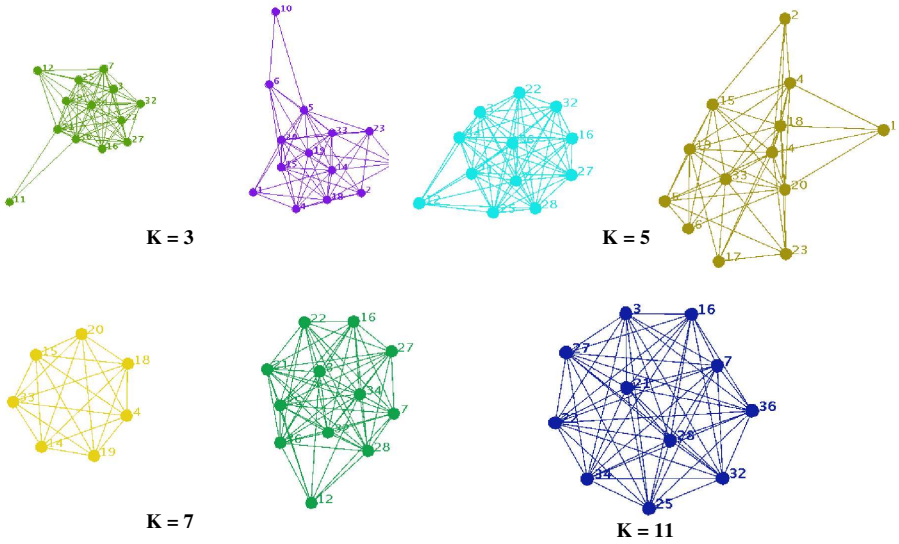


Figure 9: Communities based on contact durations with weight threshold = 10 hours, k=3,5,7, and 11 (Cambridge).

The output from the algorithm clearly illustrates that the participants can be seen as two communities in this case. When we look at the experimental data, the two communities classified by this algorithm match well with the two groups of Year1 and Year2 students selected for the experiment. Of course, in each group of students tend to know each other and meet each other, and hence the clique size can be as large as 10.

6.3 k-clique communities in Reality Mining

This is another campus environment but the environment is more diverse than the Cambridge one. Out of 100 participants, 75 are either students or faculty in the MIT Media Laboratory, while the remaining 25 are incoming students at the adjacent MIT Sloan business school. Of the 75 users at the Media lab, 20 are incoming masters students and 5 are incoming MIT freshmen. So we can see unlike the Cambridge data consisting mainly of two classes of students, this dataset consists of more groups.

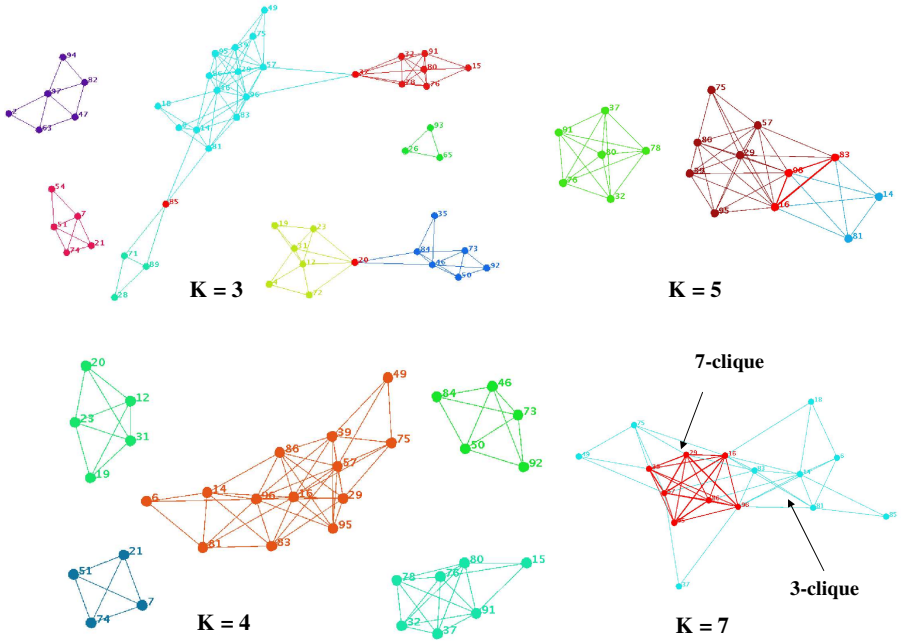


Figure 10: Communities based on contact durations with weight threshold = 388800 seconds, k=3,4,5, and 7 (Reality).

First we look at communities detected by using threshold of 388800 seconds or 108 hours on the 9 months Reality Mining dataset. Here we assume 3 lectures per week and 4 weeks per month and for a total of 9 months, we get this threshold value (2% of the total links are taken into consideration). Research students in the same office may stay together all the time a day so their contact duration threshold could be very large. For students attending lectures, this estimation can be reasonable. A looser threshold still detects the links with much stronger fit. We observe 8 communities of size (6,3,7,7,16,5,4,7) when $k = 3$ in this case. The 4-size one overlap at one node with the 16-size which also overlap with another 7-size community at another nodes. Two other 7-size nodes overlap each other with overlapping size 1. The other three communities are disjoint. When $k = 4$, the 3-clique community is eliminated and other

communities shrink or are eliminated, and only 5 communities of size (4,13,5,5,7) left. All of these 5 communities are disjoint. When $k = 5$, 3 communities of size (9,6,5) remains, the 9-size one and the 5-size one are split from the 13-size one in the 4-clique case. Moving to $k = 6$ and $k = 7$, there are 2 communities and 1 community respectively.

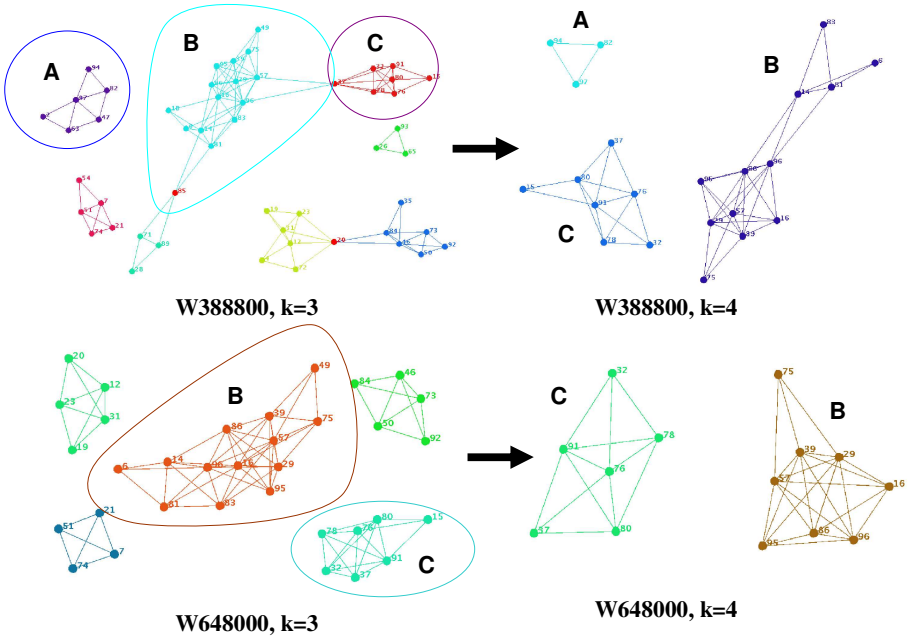


Figure 11: Communities based on contact durations with weight threshold = 648000 seconds, k=3,4 (Reality).

We are also interested in knowing about small groups which are tightly knit. We set a strict threshold of 648000 seconds, that is on average 1 hour per weekday, 4 weeks per month, and for a total of 9 months. Around 1% of the links are taken into account for the community detection. When $k = 3$, there are three disjoint communities of size (12,7,3). When $k = 4$, there are only two communities left of size (8,6). Figure 11 shows the 3-cliques and 4-clique communities of 648000 seconds threshold with its counter parts of 388800 seconds. A single 7-size community remains in $k = 5$ and $k = 6$ cases, this 7-clique community is the same as in the 388800 second case. These 7 people could be people from a same research group, they know each other and have long contact with each.

6.4 k-clique conference communities

In this section, we will show the community structures in a conference environment. Here we take Infocom06 as an example since it contains more participants than Infocom05 and we have more participants information. Infocom is a multiple-track conference with several programs running at the same time. We don't expect all our 80 experimental participants to attend the same sessions, so will not expect the clique size to be as big as the Cambridge data. The total dataset only covers 3 days, hence we will not expect the threshold to be very big. People usually socialise during conferences in a small groups so we expect clique sizes of 3, 4 or 5 to be reasonable. And for Infocom06, the participants were specially selected so that 34 out of 80 form four subgroups according to academic affiliations. Out of these four groups, there were

two groups from institutes in Paris with size of four and ten respectively(named Paris Group A and Paris Group B), and there is one group from Lausanne Switzerland of five people, and another, larger group of 15 people from the local organization in Barcelona. But for this local organization group, the volunteers are from different local institutions and also responsible for different sessions in the conference so we will not expect them to be all together. After collecting the data, for privacy purpose, all the personal information about the participants are deleted except the Node ID, the affiliation and the nationality.

Figure 12 shows the 3-clique communities with threshold 20000 seconds, that is approximately 1.85 hours per day. 1.68% of all edges are taken into account for the community calculation. We observe 6 communities of size (25,11,6,6,5,3) in this case. The 25-size one overlap at one node with a 6-size one which also overlap with the 11-size community at another node and the 3-size one at another node. The 2nd 6-size community also overlap the 3-size and 11-size at another two nodes. The 5-size community stands alone. Although we know that during a conference where the people from different sub-communities tend to mix together and hence the boundary of affiliation communities would become less clear. We still find the hints of the original affiliation communities from the figure. The algorithm correctly classified the nodes belonging to the local organizers into a community, see the Barcelona Group at the right hand side of the figure, and also the members of the Lausanne Group into another community. There are several nodes which not belonging to these affiliations are also "false positively" classified into the same communities, but this also truly reflects the nature of a conference, to socialize with people in other institutions. The two Paris groups are also clearly identified, they tend to socialize with each other. Nodes 47 is belonging to both groups, from the same figure, it is important to link this two groups together. There are many members in the 25-size group not belonging to a common institution but they are here linked together by different small groups of mixing together in conference.

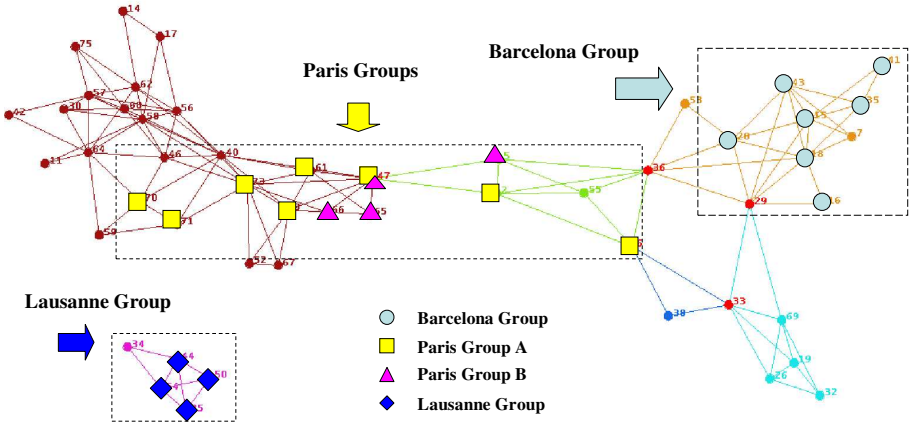


Figure 12: 3-clique communities based on contact durations with weight threshold equals 20000 seconds (Infocom06).

When we increase k from 3 to 4, it splits into 8 communities of size (8,6,6,5,5,4,4,4). The number of nodes decrease a lot, but we can also see that the tight of the affiliation communities are quite strong. The Barcelona Group and the Lausanne group are still there, just the number change from 7 to 5 and 5 to 4 respectively. The links from node 47 linking two detected communities containing Paris Groups members disappear, but we still observe a mixing of five

Paris Group A and Group B nodes together to form a community structure.

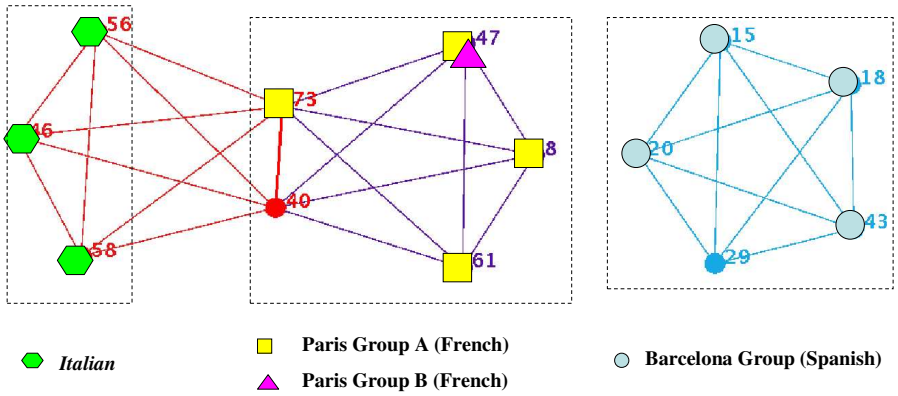


Figure 13: 5-clique communities based on contact durations with weight threshold equals 20000 seconds (Infocom06).

Figure 13 shows the communities when k is equal to 5. There are now only 3 communities of size (5,5,5). All small communities size less than 5 in k = 4 case are eliminated. We can observe that the Barcelona Group and a Paris Group are still there. Another group mainly consists of Italian speaking people overlaps with the French group. We do not want to claim that the division by the k-clique community algorithm matches perfectly to real social groups, but at least it gives us rich information about the underlying human interaction. A preliminary conclusion here is that, affiliation or even nationality have a very strong tie to human contacts, even in the conference, a highly mixed environment.

6.5 k-clique metropolitan communities

As we can see from Figure 5, most pairs have low number of contacts and contact duration. We didn't expect to discover a rich social structure from this data. However in this case, we can see how some internal nodes without much social correlation are nevertheless connected together by external Bluetooth devices, by considering all of the 869 nodes detected, including 37 iMotes and 832 external devices.

The experiment lasted 6 days. First we set the threshold to be 3 encounters which is equal to an average of one encounter per 2 days, around 8% of the total links will are taken into consideration. In this case we observed 10 communities sized (8,4,3,18,3,10,6,5,6,3) respectively when $k = 3$, which is shown on the Figure 14.

From the same figure we also see that the internal nodes are usually joined together by external nodes. They themselves may not have social correlation at all, but are connected together by these unknown external devices which may belong to colleagues or friends or familiar strangers of the iMote owners. This gives us optimism about the possibility of city-wide PSN data communication.

When $k = 4$ communities shrink to only two small communities of size 4 and 5 respectively. It seems that $k = 4$ is too strong in this case. We tried to increase the number of contacts to be 6, on average one contact per day; in this case on 2.4% of the links are taken into consideration. There are only 6 small communities of size (4,3,3,3,6,4) respectively, with only two overlapping with each other at a single node. This again confirms the very sparse social cohesion in the experiment.

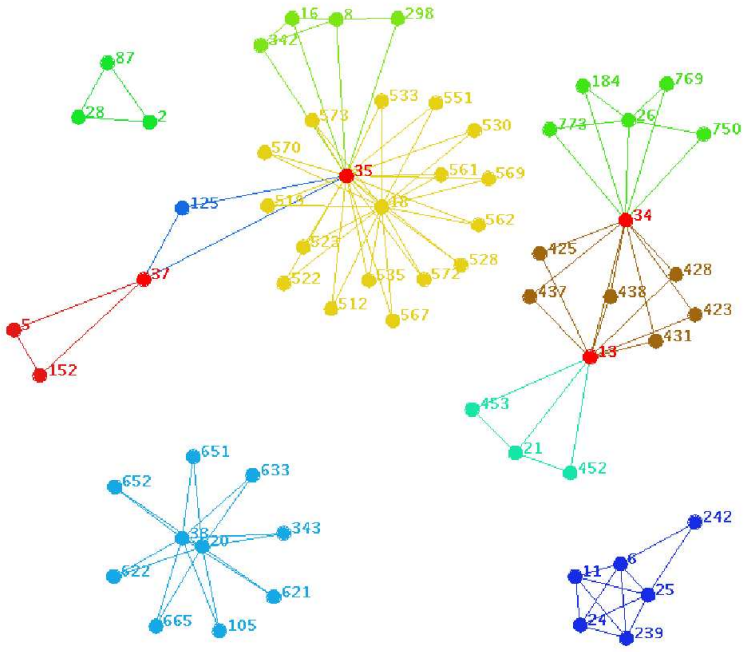


Figure 14: Communities based on number of contacts with weight threshold = 3 and k=3 (HK).

7 Interaction and Forwarding

In the first half of this paper we have shown the existence of heterogeneity at the level of individuals and groups, in all the mobility traces. This motivates us to consider a new heterogeneous model of human interaction and mobility.

Categories of human contact patterns Human relationships can be modelled by using correlation of contact duration and number of contacts. We defined four types of human relationship based on the correlation of contact duration and number of contact.

Cliques and Community We explored the community structures inside different social environments, and found these community structures match quite well to the real underlying social structures.

Popularity Ranking We shall see that popular hubs are as useful in the PSN context as they are in the wireline Internet and in the Web.

We also provide details of the statistics of interactions in the experiments so that they can be used by other researchers in future modeling, or to bootstrap larger experiments consisting of composites of these.

In the second half of this paper we look at how can we use this information to make smart forwarding decisions. The following three pre-existing schemes provide lower and upper bounds in terms of cost and delivery success. All of these schemes are inefficient because they assume a homogeneous environment. If the environment is homogeneous then every node is *statistically equivalent*, and every node has the same likelihood of delivering the messages to the destination. As we showed in the first half of this paper, the environments and nodes are

diverse, and hence all these naive schemes are doomed to have poor performance. We need to design algorithms which make use of this rich heterogeneity.

WAIT Hold on to a message until the sender encounters the recipient directly. Cheap, but unbounded expected mean delay.

FLOOD Messages are flooded throughout the entire system.

MCP Multiple-Copy-Multiple-Hop.Multiple Copies are sent subject to a time-to-live hop count limit on the propagation of messages. By exhausted emulations, 4-copy-4-hop MCP scheme is found to be most cost effective scheme in term of delivery ratio and cost for all naive schemes among all the datasets except the HK data. Hence for fair comparison, we would like to evaluate our algorithms against the 4-copy-4-hop MCP scheme in most of the cases.

The Mobile network has a dual nature: it is both a physical network and at the same time it is also a social network. A node in the network is a mobile device, and also associated with a mobile human.

Figure 15 shows the design space for the forwarding algorithms in this paper. The vertical axis represents the explicit social structure, that is facets of nodes that can specifically identified such as affiliation, organization or other social context. This is the social or human dimension. The two horizontal axes represent the network structural plane, which can be inferred purely from observed contact patterns. The Structure-in-Cohesive Group axis indicates the use of localized cohesive structure, and the Structure-in-Degree axis indicates the use of hub structure. These are observable physical characteristics. In our design framework, is not necessary that physical dimensions are orthogonal to the social dimension, but since they are represent two different design parameters, we would like to separate them. The design philosophy here is to include both the social and physical aspects of mobility into considerations.

LABEL Explicit labels are used to identify forwarding nodes that belong to the same organization. Optimizations are examined by comparing label of the potential relay nodes and the label of the destination node.This is in the human dimension, although an analogous version can be done by labelling a k-clique community in the physical domain.

RANK This is analogous to the degree of a node in a fixed network; we use a modified ranking scheme, namely the node centrality in a temporal network. It is based on observations in the network plane, although it also reflects the hub popularity in the human dimension.

DEGREE A heuristic based on the observed average of the degree of a node over some longer interval. Either the last interval window (S-Window), or a long term accumulative estimate, (A-Window)) is used to provide a fully decentralized approximation for each node's centrality, and then that is used to select forwarding nodes.

BUBBLE The Bubble family of protocols combines the observed hierarchy of centrality of nodes with explicit labels, to decide on the best forwarding nodes. Bubble is an example algorithm which uses information from both the human aspects and also the physically observable aspects of mobility.

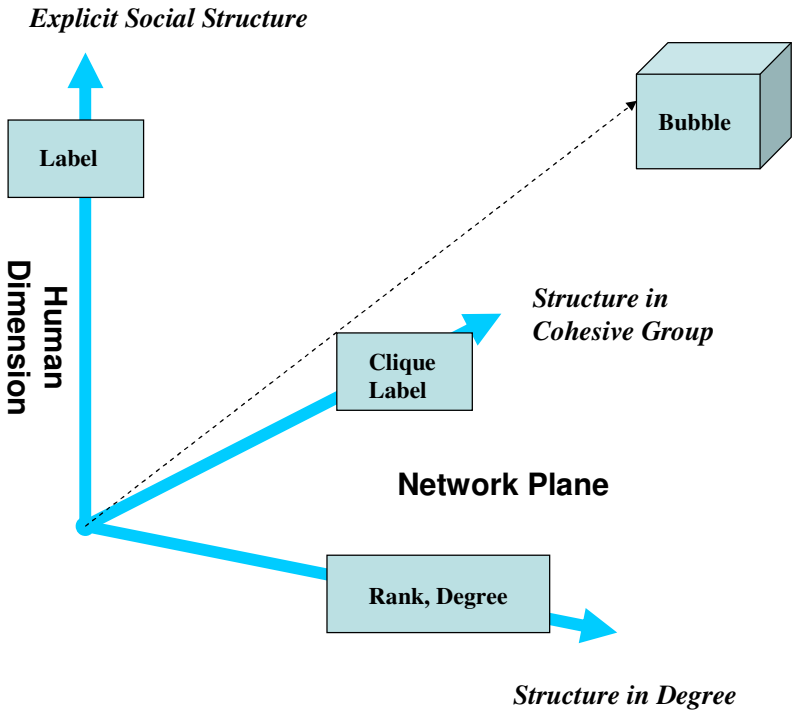


Figure 15: Design space for forwarding algorithms.

In the following sections, we will show how can we make use of these different metrics to improve forwarding performance in a heterogeneous system and also when they will fail. We focus on empirical analysis; that is what our mobile network research communities most lack; we do not consider abstracting a mathematical model in this work, but evaluate the forwarding schemes directly on the mobility traces.

8 Greedy ranking algorithm

The third contribution of this paper is to modify the greedy ranking search scheme over power law networks to apply to our temporal graphs, and evaluate the resulting algorithm.

8.1 The Power of Greedy Ranking

Here we use a similar greedy strategy to the one Adamic et al. introduced in [1]. A PSN is not like Internet: we do not know when a global or local maximum is reached since the next encounter is unexpected. We cannot employ precisely the same strategy as they propose, of traversing up the hierarchy until reaching the maximum, and then down a step. Here we also assume each node knows only its own ranking and the rankings of those it encounters, but does not know the ranking of other nodes it does not encounter, and does not even know which node has the highest rank in the system. Our strategy, which we call RANK, is very simple: we keep pushing traffic on all paths to nodes which have a higher ranking than the current node, until either the destinations are reached, or the messages expire.

If a system is small enough, the global ranking of each node is actually the local ranking.

If we consider only the Rummidge Computer Laboratory System Research group, this is the the ranking of each node inside the group. If we consider the whole Computer Laboratory, we are considering a larger system of many groups, but they all still use the same building. A homogeneous ranking can also work. But when we consider the whole city of Rummidge, a homogeneous ranking would exclude many small scale structures. In this section we show that in relative small and homogeneous systems, a simple greedy ranking algorithm can achieve good performance.

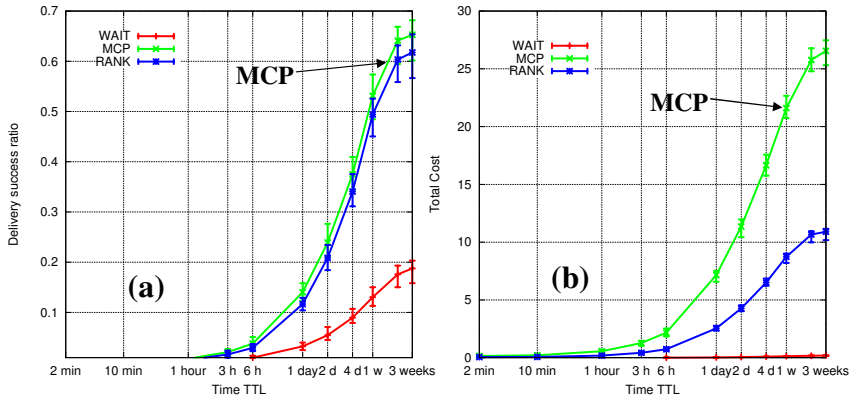


Figure 16: Comparison of delivery ratio (left) and cost(right) of MCP and greedy RANK on 4 copies and 4 hops case (Reality).

Figure 16(a) shows that the simple greedy ranking perform almost as well as MCP for delivery. Figure 16(b) also shows that the cost is only around 40% that of MCP, which represents a marked improvement.

Hierarchical organization is a common feature of many complex systems. The defining feature of a hierarchical organization is the existence of a hierarchical path connecting any two of its nodes. Trusina et al. [37] address how to detect and measure the extent of the hierarchy manifested in the topology of a given complex network. They defined the hierarchical path based on node degrees, a path between two nodes in a network is called hierarchical if it consists of an "up path" where one is allowed to step from node i to node j only if their degrees k_i, k_j satisfy $k_i u \leq k_j$, followed by a "down path" where only steps to nodes of lower or equal degree are allowed. Either the up or down path is allowed to have zero length. Because of the good achievement from the greedy ranking algorithm, we are going to analyse the percentage of hierarchical paths inside all the shortest paths. Table 3 summarises the results.

Experimental data set	% hierarchical paths
Rummidge	87.2 (-2.4,+4.3)
Reality	81.9 (-3.1,+3.3)
Infocom05	62.3 (-2.5,+2.5)
Infocom06	69.5 (-4.1,+2.4)
Hong Kong	33.5 (-4.0,+4.0)

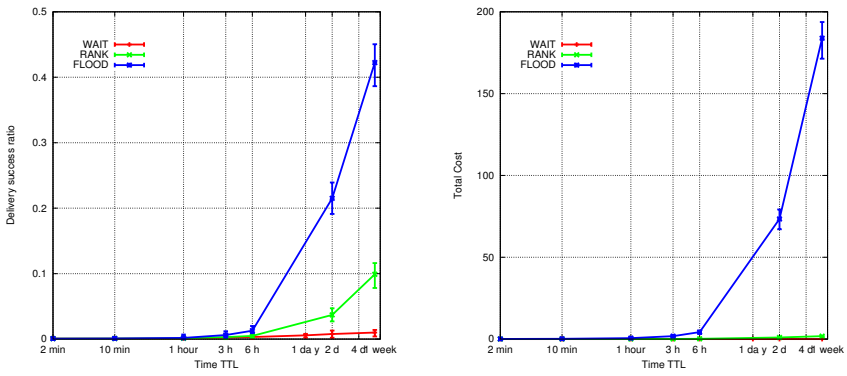
Table 3: Hierarchical Paths analysis of all shortest paths

The percentage of hierarchical paths is calculated as the number of hierarchical paths divided by the number of non-direct transfer deliveries. We can see that for Rummidge data and

Reality Mining, the percentage of hierarchical paths is very high, so our strategy of pushing the messages up the ranking tree can probably find a lot of these paths, and the performance of the ranking strategy here is not much different from the MCP. For Infocom06 and Infocom05, the percentages of hierarchical paths is also high, so the greedy RANK strategy can also discover many of the shortest paths. However, for Hong Kong experiment, the network is too sparse and a lot of shortest paths are hidden, because we could not know the devices detected by the external devices, and most of the resulting paths used for delivery are actually not the shortest . We can see that percentage of hierarchical paths controls the delivery success that is achieved by the greedy RANK algorithm. We conclude from this that a very high percentage of the shortest paths are actually hierarchical paths.

8.2 Where the Greedy Ranking Fails

For the Hong Kong dataset, the 37 participants are intentionally selected without any social correlation. They live and work distributively throughout the whole city. Relying on direct contact, less than 4% of the messages can be delivered. Unlike all the previous datasets, here all the external Bluetooth devices detected need to be used for constructing the paths. But because we don't know the devices detected by all these external devices so a lot of potential paths not found.



(a) Delivery

(b) Cost

Figure 17: Comparison of delivery ratio and cost of MCP and Greedy RANK on no constraints case (HK)

Figure 17(a) and Figure 17(b) show the delivery ratio and delivery cost using flooding, and using unlimited greedy ranking. We can see that using flooding, we can deliver more than 40% of the total traffic across the whole city by using only the 37 iMotes and the external devices detected by these iMotes without knowing the devices detected by the external devices, that will be a huge number of paths out of these 869 devices. However the cost is also very high: to deliver one message, we need to make around 180 copies. But in this case, greedy ranking can only deliver 10% of the messages, although the cost is much lower as well. In terms of delivery and cost, greedy ranking is still more cost-effective than flooding, but clearly the delivery success rate is still too low. One explanation for this low performance is that since the participants have no social correlation, and belong to different social communities, high global ranking of a node may not represent a good choice of relay for some local communities.

Messages keep being pushed up to some globally higher ranking nodes, and getting stuck at some maxima, rather than then trickling down to some local community. Figure 18(a) shows that the maximum number of hops for greedy Rank is 4 hops and after that the messages get stuck. Figure 18(b) shows the rank distribution of the source, destination and dead-end of all the undelivered messages, we can see that these "dead-end nodes" have relatively high ranking, and this supports our hypothesis concerning messages stuck at maxima.

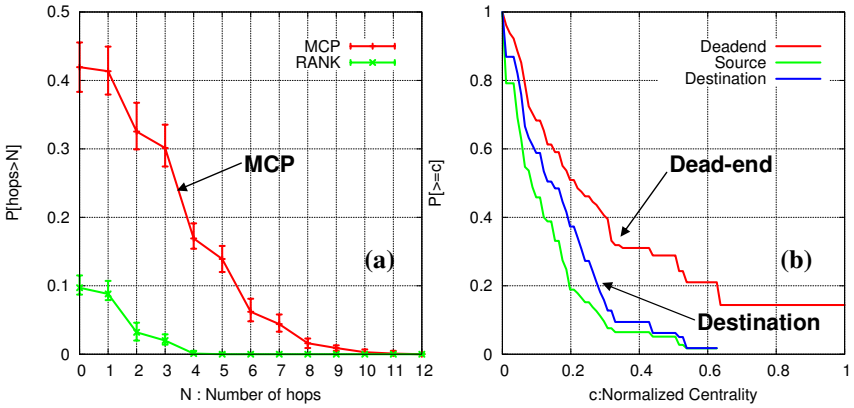


Figure 18: The hop distribution of the delivered(left) and the rank distribution of undelivered(right) on HK data.

9 Direct Labelling Strategy

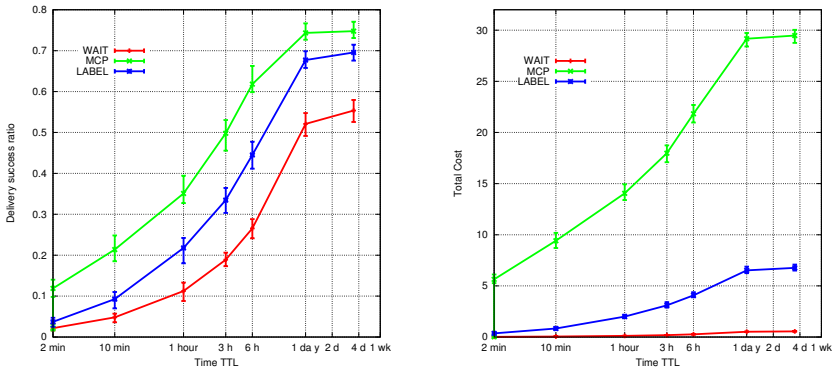
In the "labelling strategy" [14], each node is assumed to have a label that tells others its affiliation, just like a name badge in a conference. The "direct labelling strategy" refers to the exclusive of labels to forward messages to destinations: next-hop nodes are selected if they belong to the same group (same label) as the destination. Our fourth contribution is to evaluate the improvements to forwarding possible using this *a priori* affiliation label data.

9.1 The Power of Labelling

The direct labelling strategy is evaluated on the Infocom06 data. Since this is a conference scenario, where people meet frequently, direct labelling strategy works quite well as we might expect. In Figure 19(a) we see that, as expected, LABEL has a delivery ratio between MCP and WAIT, and the trend is for it to approach closer to the performance of MCP, as we increase the lifetime (TTL) of message. In terms of cost, in Figure 19(b) we can see that MCP costs much more than LABEL, especially when TTL is increased to 1 day, where MCP has less than a 10% improvement over LABEL, but has around 6 times the cost. Of course, WAIT has the lowest cost: since we are in a conference scenario, we do not expect to wait long to meet the destination, hence the delivery ratio is not too low.

9.2 The Problem with Direct Labeling

A human community represents one type of long term, stable relationship. An outside observer of human society would not know at first to which group each person belongs. As time goes by,



(a) Delivery (b) Cost

Figure 19: Comparison of delivery ratio and cost of MCP and LABEL on 4 copies and 4 hops case (Infocom06)

we gain higher confidence concerning who usually socialises with whom. In this part of analysis, we use the communities detected from the 9 month Reality Mining traces. Nine months is a long enough period for us to have high confidence to believe that the communities extracted from the dataset truly reflect the social communities existing between the participants. We think it is accurate, then, to evaluate the labelling strategy on this dataset.

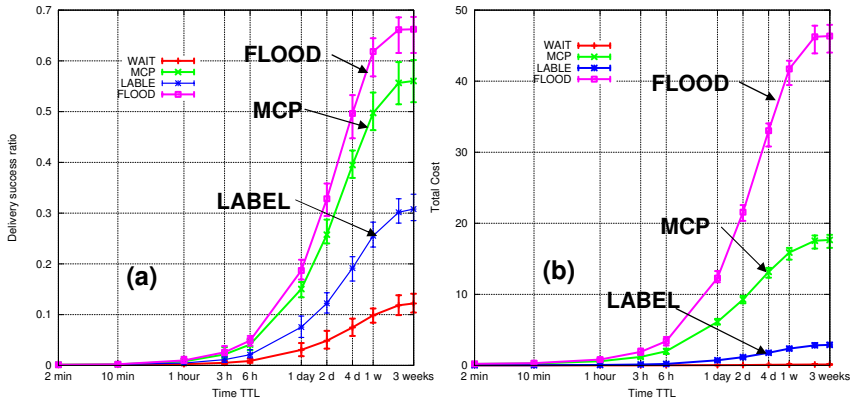


Figure 20: Comparison of delivery ratio(left) and cost(right) of MCP and LABEL on 4 copies and 4 hops case (Reality).

We can see from Figure 20 that "labelling strategy" only achieves around 55% of the delivery ratio of the MCP strategy and only 45% of the flooding delivery although the cost is also much lower. However it is not an ideal scenario for LABEL. In this environment, people do not mix as well as in a conference. A person in one group may not meet members in another group so often, so waiting until the member of the another group appear to do the transmission is not effective here.

Figure 21 shows the correlation of the nth-hop relay nodes to the source and destination groups for the messages on all the shortest paths, that is the percentage of the nth-hop relay nodes that are still in the same group as the source or already in the same group as the destination. We can see that more than 50% of the nodes on the first hops (from the S-Group plot)

are still in the same group as the source group of the message and only around 5% of the first hop nodes (from the D-Group plot) are in the same group as the destination. This explains why direct labelling is not effective, since it is far from discovering the shortest path. We can also see that on going to the 2nd hop, S-Group correlation drops to slightly less than 30%, and when going to 4th-hops, almost all (90%) messages have escaped from this source group. To calculate the percentage for each hop we just divide the count of messages which belong to that group by the total count of messages destined beyond that node, but not the total messages created. In the 4-hop case, there are perhaps only 100 messages to forward further, and only 10 out of these 100 relay nodes belong to the source group.

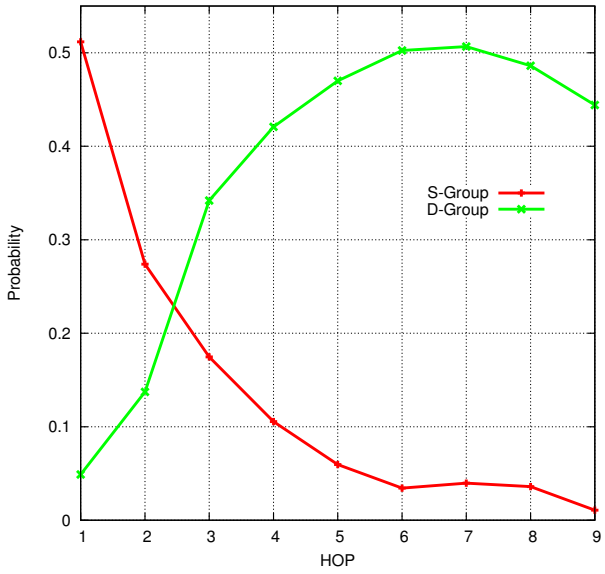


Figure 21: Correlation of nth-hop nodes with the source group and destination group (Reality).

10 Centrality Meets Community

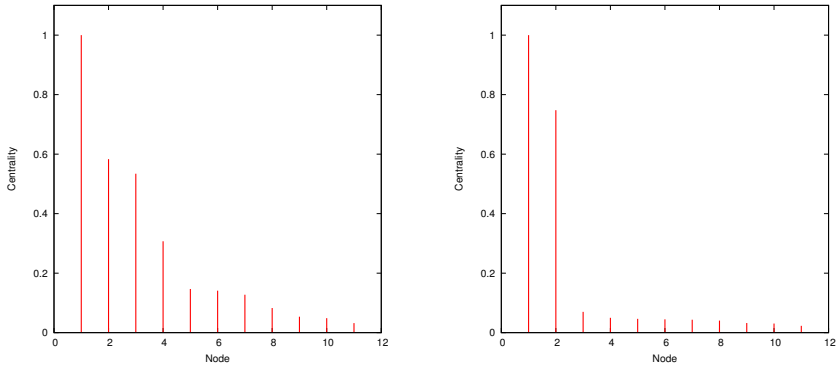
The fifth contribution in this paper is to combine the knowledge of both the centrality of nodes and the community structure, to achieve further performance improvements in forwarding. We show that this avoid the occurrence of the dead-ends encountered with pure global ranking schemes. We call the protocols here BUBBLE, to capture our intuition about the social structure. Messages bubble up and down the social hierarchy, based on the observed community structure and node centrality, together with explicit label data. Bubbles represent a hybrid of social and physically observable heterogeneity of mobility over time and over community, and contrast with the notion of a pocket, which is a DTN area of current wireless reachability.

10.1 Two-community Case

In order to make the study more systematic, we start with the two-community case. We use the *Cambridge* dataset for this study. By experimental design, and confirmed using our community detection algorithm, we can clearly divide the *Cambridge* data into two communities:

the undergraduate year1 and year2 group. In order to make the experiment more fair, we limit ourselves to just the two 10-clique groups found with a number-of-contact threshold 29; that is where each node at least meet another 9 nodes frequently. Some students may skip lectures and cause variations in the results, so this limitation makes our analysis yet more plausible.

First we look at the simplest case, for the centrality of nodes within each group. In this case, the traffic is created only for members within the same community and only members in the same community are chosen as relays for messages. We can see clearly from Figure 22(a) and

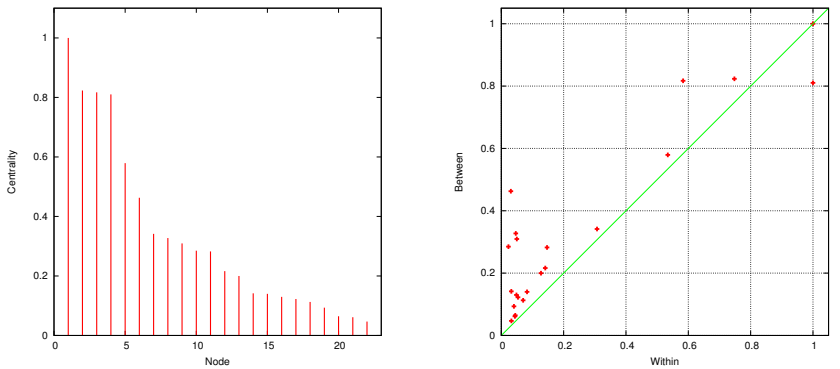


(a) Group A (b) Group B

Figure 22: Node centrality in 2 groups in Cambridge data

22(b) that inside a community, the centrality of each node is different. In Group B, there are two nodes which are very popular, and relayed most of the traffic. All the other nodes have very low centrality value. Forwarding messages to the popular nodes would make delivery more cost effective for messages within the same community.

Then we consider traffic which is created within each group and only destined for members in another group. To eliminate other outside factors, we use only members from these two groups as relays. Figure 23(a) shows the individual node centrality when traffic is created from



(a) Centrality (b) Correlation

Figure 23: Inter-group centrality and correlation between intra- and inter-group centrality (Cambridge)

one group to another. Figure 23(b) shows the correlation of node centrality within an individual

group and inter-group centrality. We can see that points lie more or less around the diagonal line. This means that the inter- and intra- group centralities are quite well correlated. Active nodes in a group are also active nodes for inter-group communication. There are some points on the left hand side of the graph which have very low intra-group centrality but moderate inter-group centrality. These are nodes which move across groups. They are not important for intra-group communication but can perform certainly well when we need to move traffic from one group to another.

We can show now why homogeneous global ranking in section 8 does not work perfectly. Figure 24 shows the correlation of the local centrality of Group A and the global centrality of the

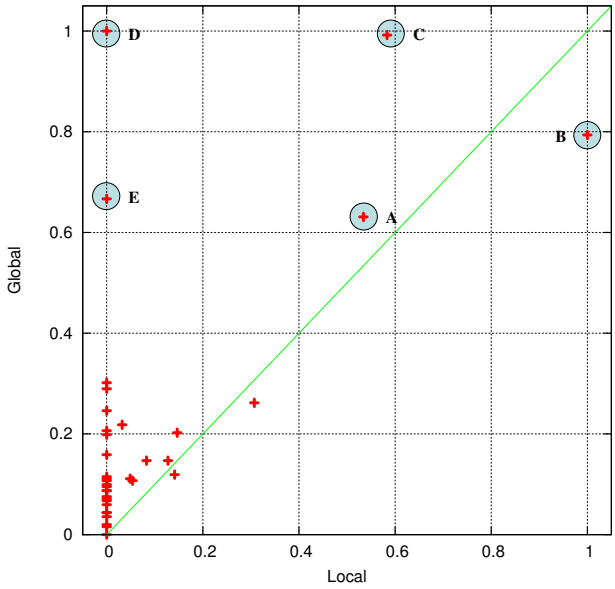


Figure 24: Correlation of local centrality of group A and the global centrality (Cambridge).

whole population. We can see that quite a number of nodes from Group A lie along the diagonal line. In this case the global ranking can help to push the traffic toward Group A. However the problem is that some nodes which have very high global rankings are actually not members of Group A, for example node D. Just as in real society, a politician could be very popular in the city of Cambridge, but not a member of the Computer Laboratory, so would not be a very good relay to deliver message to the member in the Computer Laboratory. Now we assume there is a message at node A to deliver to another member of Group A. According to global ranking, we would tend to push the traffic toward B, C, D, and E in the graph. If we pushed the traffic to node C, it would be fine, to node B would be perfect. But if it push the traffic to node D and E, the traffic could get stuck there and not route back to Group A. If it reaches node B, that is the best relay for traffic within the group, but node D has a higher global ranking than B, and would tend to forward the traffic to node D, where it would probably get stuck again.

Hence we now propose the following forwarding algorithm 1 to avoid these dead-ends:

If a node has a message destined for another node, this node would first bubble this message up the hierarchical ranking tree using the global ranking until it reaches a node which has the same label(community) as the destination of this message. Then the local ranking system will be used instead of the global ranking and continue to bubble up the message through the local ranking tree until the destination is reached or the message expired. This method does not

Algorithm 1: BUBBLE RAP

```
begin
    var use GlobalRanking ← true
    if (Label(currentNode) == Label(destination)) then
        useGlobalRanking ← false
    foreach EncounterNode i do
        if Rank(node_i) > Rank(currentNode) or Label(node_i) == Label(destination) then
            Buffer(node_i) ← Buffer(node_i) ⋃ {message}
end
```

require every node to know the ranking of all other nodes in the system, but just to be able to compare ranking with the node encountered, and to push the message using a greedy approach. We call this algorithm Bubble-A, since each world/community is like a bubble. Figure 25 illustrates the algorithm. A global bubble is always relative to local bubble. This global bubble maybe a sub-bubble of another larger bubble.

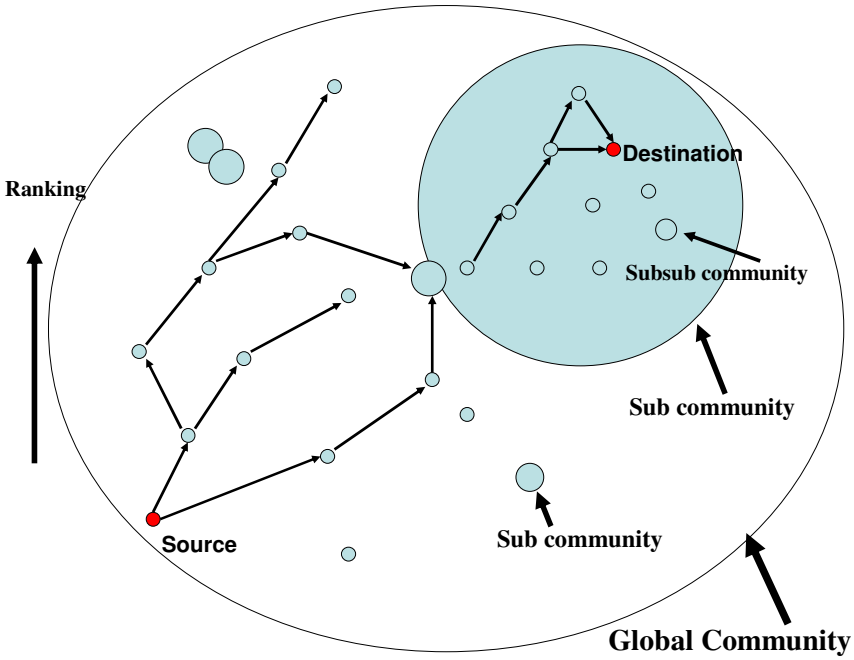


Figure 25: Illustration of the bubble forwarding algorithm.

This fits our intuition in terms of real life. First you try to forward the data via people more popular than you around you, and then bubble it up to well-known popular people in the society, such as a postman. When the postman meets a member of the destination community, the message will be passed to that community. This community member will try to identify the more popular members within the community and bubble the message up again within the local hierarchy until the message reach a very popular member, or the destination itself, or the message expires.

A modified version of this strategy is that whenever a message is delivered to the community, the original carrier can delete this message from its buffer to prevent it from further dissemination. This assumes that the community member would be able to deliver this message. We call this protocol with deletion, strategy Bubble-B.

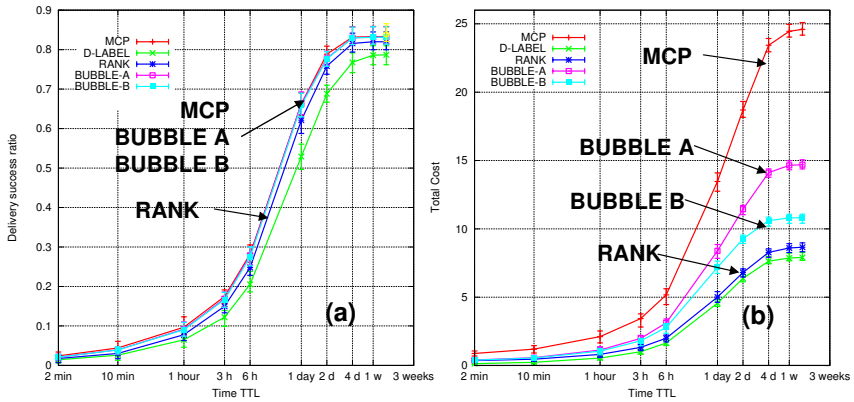


Figure 26: Comparisons of several algorithms on Cambridge dataset, delivery(left) and cost(right).

We can see from Figure 26(a) that both Bubble-A and Bubble-B achieve almost the same delivery success rate as the 4-copy-4-hop MCP. Although Bubble-B has the messages deletion mechanism, it achieves exactly the same delivery as Bubble-A. From Figure 26(b), we can see that Bubble-A only has 60% the cost of MCP and Bubble-B is even better, with only 45% the cost of MCP. Both have almost the same delivery success as MCP.

10.2 Multiple-community Cases

To study the multiple-community cases, we use the Reality Mining dataset as in section 9.2.

To evaluate the forwarding algorithm, we extract a 3 week session during term time from the whole 9 month data set. Emulations are run over this dataset with uniformly generated traffic.

There is a total 8 groups within the whole dataset. Figure 27 shows the node centrality in 4 groups, from very small size to medium size and large size group. We can see that within each group, almost every node has different centrality.

In order to make our study easier, we first isolate just one group, the largest one in Figure 27, consisting of 16 nodes. In this case, all the nodes in the system create traffic for members of this group. We can see from Figure 28(a) that Bubble-A and Bubble-B perform very similarly to MCP most of the time in the single group case, and even outperform MCP when the time TTL is set to be larger than 1 week. From Figure 28(b), we can see that Bubble-A only has 70% and Bubble-B only 55% of the cost of MCP. We can say that the Bubble algorithms are much more cost effective than MCP, with high delivery ratio and low delivery cost. After the single group case, we start looking at the case of every group creating traffic for other groups, but not for its own members. We want to find the upper cost bound for the Bubble algorithm, so we do not consider local ranking; messages can now be sent to all members in the group. This is exactly a combination of direct LABEL and greedy RANK, using greedy RANK to move the messages away from the source group. We do not implement the mechanism to remove the original message after it has been delivered to the group member, so the cost here will represent an upper bound for Bubble type algorithms.

From Figure 28(c) and Figure 28(d), we can see that of course flooding achieves the best performance for delivery ratio, but the cost is 2.5 times that of MCP, and 5 times that Bubble. Bubble is very close in performance to MCP in multiple groups case as well, and even outper-

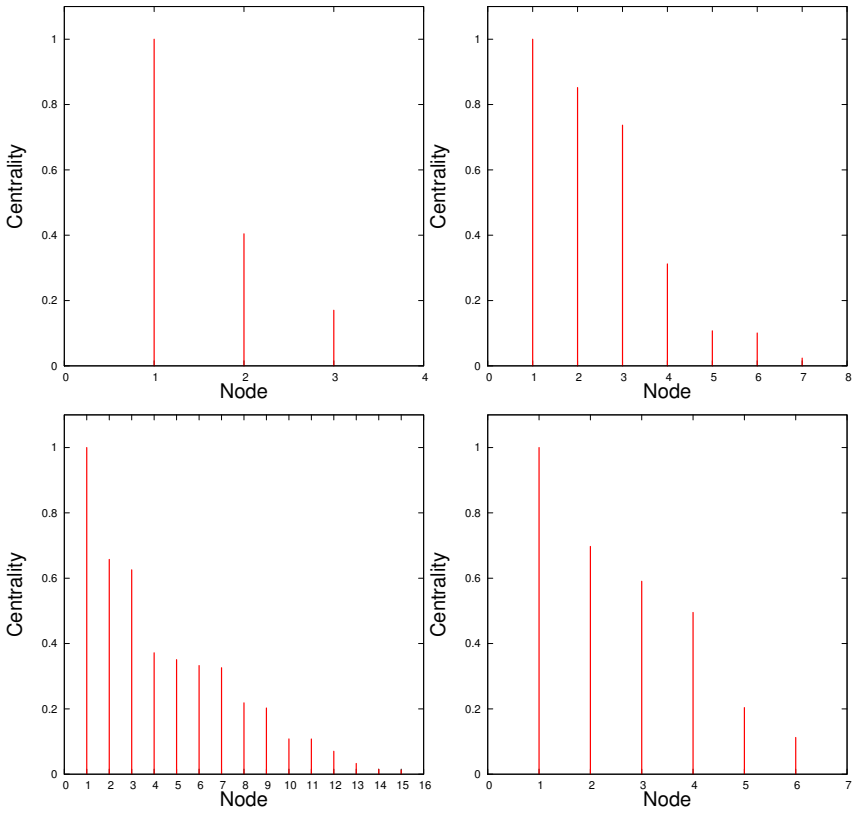


Figure 27: Node centrality in several individual groups in Reality Mining.

forms it when the time TTL of the messages is allowed to be larger than 2 weeks. However, the cost is only 50% that of MCP.

11 Making centrality practical

Although the greedy RANK algorithm fail sometimes in very heterogenous system to deliver messages to a member in a small group, it reduce a lot of the cost at the same time. And we would think it to be a good bootstrap step for other forwarding algorithms to push traffics away from the source node. If we want to deliver a message to somebody, first try to give it to someone who you know to be popular. So we would not doubt that centrality is an important metrics for a PSN. Then we would ask these questions: How can each node know its own centrality in a decentralised way? How well does past centrality predict the future.

The final contribution of this paper is to provide early answers to these two questions.

11.1 Approximating centrality

We found that the total degree (unique nodes) seen by a node throughout the experiment period is not a good approximation for the node centrality. Instead the degrees per unit time (for example the number of unique nodes seen per 6 hours) and the node centrality has a high correlation value. We can see from Figure 29 that some nodes with a very high total degree are still not good carriers. It also shows that the per 6 hour degree is quite well correlated to

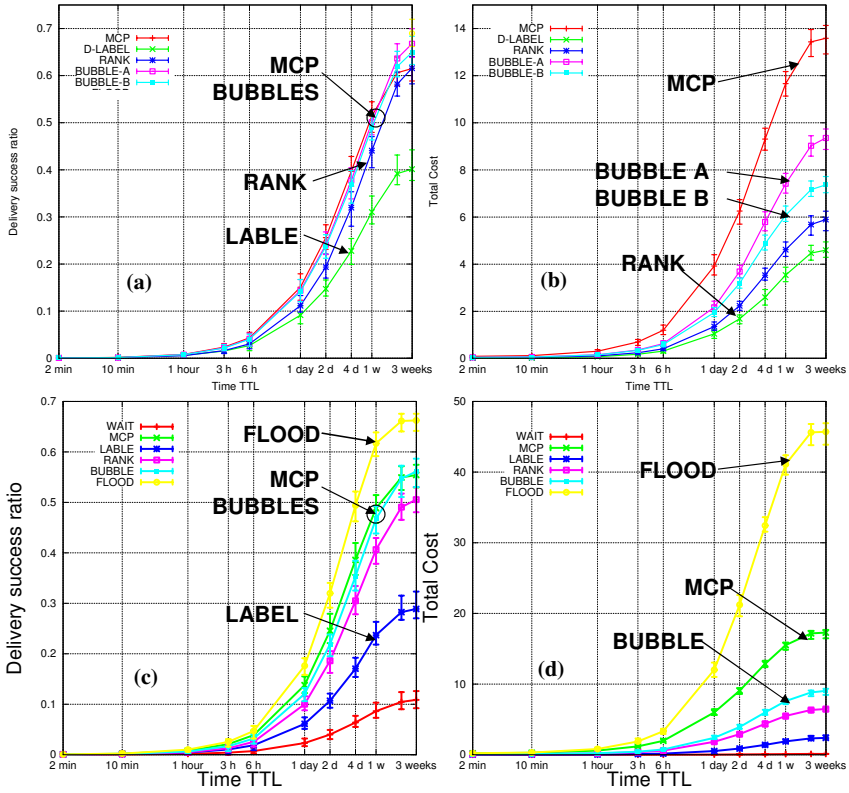


Figure 28: Comparisons of several algorithms on Reality Mining dataset, single group and all groups.

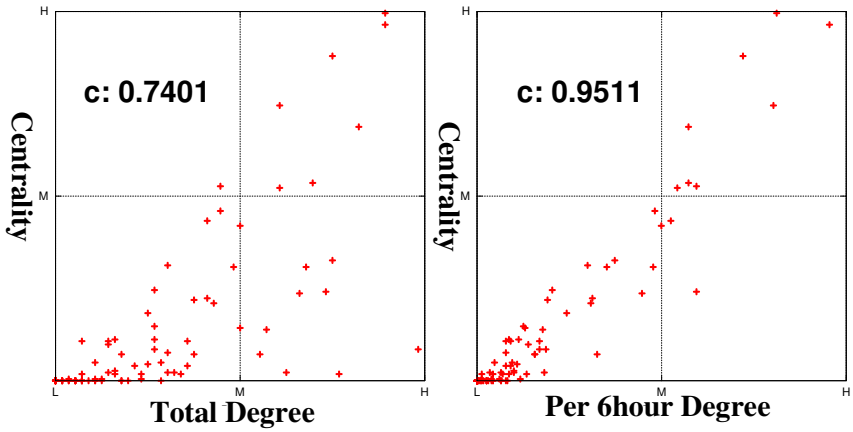


Figure 29: Correlation of rank with total degree and rank with unit time degree (Reality).

the centrality value, with correlation coefficient as high as 0.9511. The means that how many people you know doesn't matter too much, but how frequently you interact with these people matters.

In order to verify that the average unit-time degree is as good as or close to RANK, we run another sets of emulations using greedy average unit-time degree(or we simply call it DEGREE) instead of the pre-calculated centrality. Figure 30(a) and Figure 30(b) compare the delivery

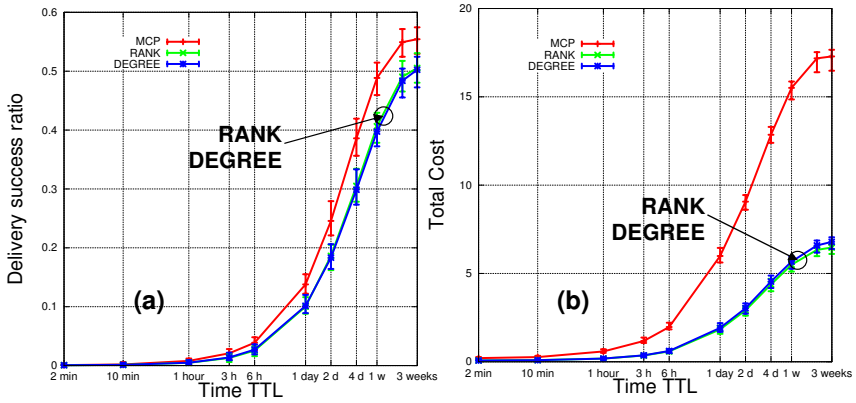


Figure 30: Comparisons of delivery(left) and cost(right) of RANK and DEGREE on Reality Mining dataset, all groups.

ratio and delivery cost of using greedy RANK and greedy DEGREE. We can see that RANK and DEGREE perform almost the same with the delivery and cost lines overlapping each other. They not only have similar delivery but also similar cost.

However, the average unit-time degree calculated throughout the whole experimental period is still difficult for each node to calculate individually. We then consider the degree for previous unit-time slot(we call this the slot window) such that when two nodes meet each other, they compare how many unique nodes they have met in the previous unit-time slot (e.g. 6 hours). We call this approach the single window (S-Window). Another approach is to calculate the average value on all previous windows, such as from yesterday to now, then calculate the average degree for every 6 hours. We call this approach the accumulative window (A-Window). This technique is similar to a statistics technique called exponential smoothing [40] and we would like to do further theoretical investigation.

The S-Window approach reflects more recent context and achieves maximum of 4% improvement in delivery ratio than DEGREE, but at double the cost. The A-Window approach measures more of the accumulative effect, and gives more stable statistics about the average activeness of a node. However, its accumulative measurement is not as good an estimate as DEGREE, which averages throughout the whole experimental period. It does not achieve as good delivery as DEGREE (not more than 10% less in term of delivery), but it also has lower cost.

All these approaches, (DEGREE, S-Window and A-Window) can provide us with a decentralised way to approximate the centrality of nodes in the system, and hence help us to design appropriate forwarding algorithms.

11.2 Human predictability

The second question above can be generalized to: how much can human interaction be predicted from the past contact history? In this section, we use vertex similarity, which has been well

studied in citation networks, to study the predictability of human interaction from the contact graph. Additionally, we run some emulations on traces to see how much the past centrality can predict the future centrality.

11.2.1 Vertex similarity

There are several ways to compare structural vertex similarity in the previous works. Two vertices are considered *structural equivalence* if they share many of the same network neighbors,

$$\sigma_{\text{Jaccard}} = \frac{|\Gamma_i \bigcap \Gamma_j|}{|\Gamma_i \bigcup \Gamma_j|} \tag{1}$$

$$\sigma_{\text{cosine}} = \frac{|\Gamma_i \bigcap \Gamma_j|}{\sqrt{|\Gamma_i||\Gamma_j|}} \tag{2}$$

$$\sigma_{\text{min}} = \frac{|\Gamma_i \bigcap \Gamma_j|}{\min\left(|\Gamma_i||\Gamma_j|\right)} \tag{3}$$

where Γ_i is the neighborhood of vertex *i* in a network, which is the set of vertices connected to vertex *i* via an edge. $|\Gamma_i|$ is the cardinality of the set Γ_i, that is equal to the degree of the vertex *i*. The Jaccard index [30] above was proposed by Jaccard over hundred years ago, and the cosine similarity has a long history of study on citation networks [33]. Here we use the vertex similarity to measure the predictability of human interaction: we can compare the vertex similarity of the contact graphs over two days and tell how similar human interaction is on these two days. Averaging over all the vertexes, we get an estimation for the whole population. We call this simply *graph similarity*. We have studied all the three metrics, but the trends are similar, and so we just present the results of the classic Jaccard measurement here.

We look at the dataset of the Reality Mining data from 1st February to 30th April 2005. The reason for choosing this period is that it is far from the new academic year so the human relationship are already relatively stable and also it is term time so the participants will be more active in the campus. We study the vertex similarity and the simple graph similarity for every two consecutive days and also for every pair of days against the date of the 1st of February for these three months. We consider it as a binary graph; we do not consider the weight for the edges, but just consider the existence of an edge. The three metrics proposed above do not apply to a weighted graph.

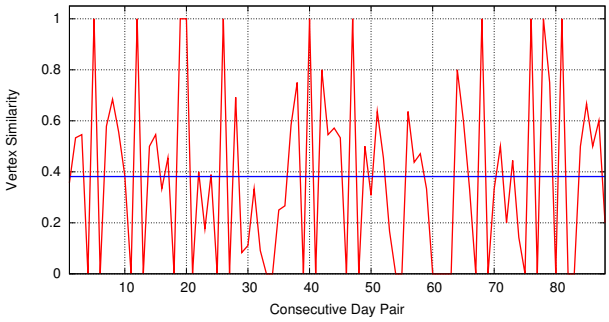


Figure 31: Vertex similarity of every consecutive day pairs of a single node

Figure 31 shows the Jaccard vertex similarity of an active node, i.e. a node with high centrality value, for the 88 consecutive day pairs. The horizontal line at the middle shows the

average value. In our calculation, when two comparing vertexes have both cardinalities equal to 0, we count their similarity to be 1, the maximum Jaccard similarity. We can see that the trough(minimum) points are corresponding to a change from weekday to weekend and also weekend to weekday; and the peak(maximum) points are corresponding to a transition from Saturday to Sunday, so there is always a peak surrounded by two troughs. We see that the nodes met by this node during the week days are very different the those nodes met during the weekend. For the weekend, the nodes meet have a very high probability to meet them during the second weekend day. But even during week day, there is around 50% of the nodes meet one day will meet again the second day. This is the case for the active nodes, but for the less active node, i.e. the nodes with low centrality value, we find out that they have the highest vertex similarity value 1 almost everyday. These nodes usually see exactly the same nodes everyday, this also explain why they have low centrality values.

Figure 32 shows the simple graph similarity for the contact graphs of every consecutive day. We can see that the average value is as high as 0.7, that is for the whole population studied the human interaction pattern of whom with whom is quite predictable for every two consecutive days. The peaks here are also corresponding to the transition from a Saturday to a Sunday.

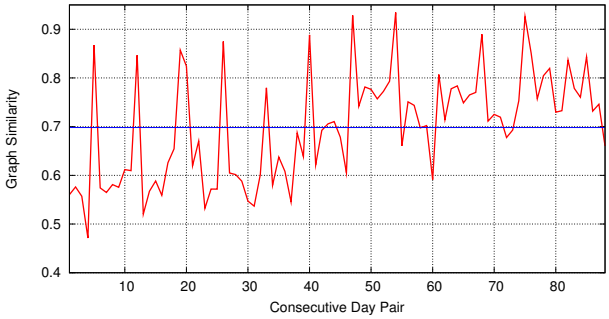


Figure 32: Simple graph similarity of every consecutive day pairs

In order to see more the phase transition from weekday to weekend more clearly, and also to look at whether there is any long term attenuation for the human interaction in this system, we compare every day with the first day of the period we studied, which is 1st February and is a weekday. Figure 33 shows the vertex similarity of every day pair corresponding to the first day of the study period. We can see that the vertex similarity drops to zero from a weekday to a weekend transition and stay zero for the whole weekend. And we didn't see the long term attenuation effect from the graphs we produced. Similar trend of changes are also observed in the graph similarity graph.

But if we want to further look at whether the same node pair stay similar time together for a day pair and also whether they meet similar number of times everyday, we need to consider a weighted version of measurement for this kind of similarity. Since we cannot find useful metrics from the literature, we need to devise our own:

$$\sigma_{\text{weight}} = \frac{\sum_0^n \min(w_{it})}{\sum_0^n \max(w_{it})} \tag{4}$$

where $n = |\Gamma_i \bigcup \Gamma_j|$, $\min(w_{it})$ is the minimum of the weight for an edge connecting node *i* and one of its neighbor in the two graphs, and $\max(w_{it})$ is the maximum of the weight for an edge connecting node *i* and one of its neighbor in the two graphs. If there is no edge in the graph,

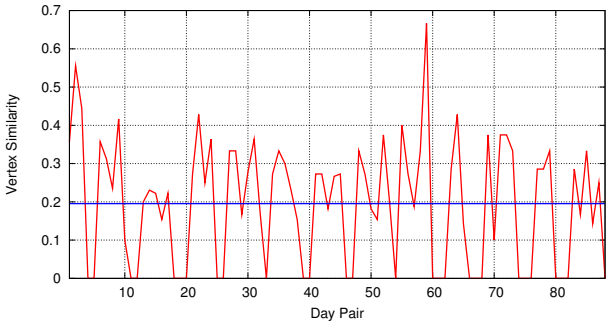


Figure 33: Vertex similarity of every day pairs with a randomly chosen weekday of a single node

we count its weight to be 0. Here we count number of contacts as the weight and calculate the vertex similarity for all nodes and also the graph similarity. Figure 34 shows the weighted vertex similarity for every consecutive day pair for the same node as show in the previous. We still observe the transition from weekday to weekend and vice versa. The horizontal lines in the middle show the average. It is around 0.3, that is not very high but the reason is because of the transition from weekday to weekend and weekend to weekday would produce two 0 values. But if we look at the whole population in Figure 35, we can see that even the contact frequency of two consecutive days are quite predictable, with an average of close to 0.7.

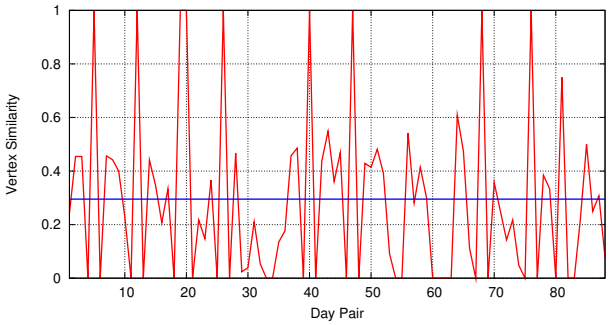


Figure 34: Weighted vertex similarity for every consecutive day pair of a single node

We will look at the similarity of different time durations, the impact of different period of the day, i.e. the nodes see during the day time should be different than the nodes during night time, and different data analysis technics such as correlation and matrix analysis will be used. This result may only limited to a academic campus but we will also look at more complex environments in the future. An early conclusion we can make here is that daily human interaction is quite predictable in the unit of per day, nodes meet on one day have quite high probability to meet again in the next day. This provide an indirect answer to the predictability of centrality as well.

11.2.2 Predictability of centrality

In order to further verify whether the centrality measured in the past is useful as a predictor for the future. We extracted three temporally consecutive 3-week sessions from the Reality

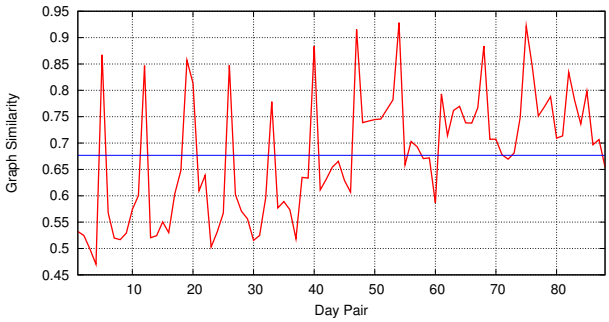


Figure 35: Vertex similarity of every day pairs with a randomly chosen weekday of a single node

dataset and then run a set of greedy RANK emulations on the last two data sessions, but using the centrality values from first session.

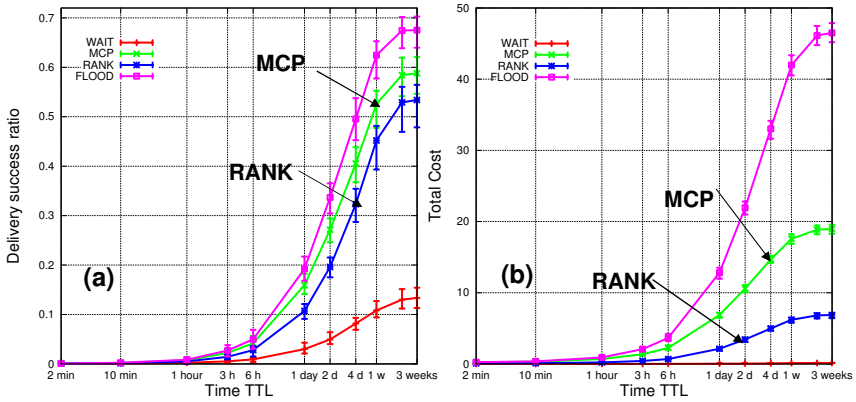


Figure 36: Delivery ratio(left) and cost(right) of RANK algorithm on 2nd data session, all groups (Reality)

Figure 36(a),(b) show the delivery ratio and cost of RANK on the 2nd data session using the centrality values from the 1st data session. It seems that the performance of RANK is not far from MCP but with much lower cost. The performance is as good as in the original dataset. Similar performance is also observed in the 3rd data session. These results imply some level of human mobility predictability, and show empirically that past contact information can be used in the future.

12 Related work

Community structures in complex networks have attracted a lot of attention in recent years. There is still no universally accepted definition of community, but in most versions, community is a subgraph of a network whose nodes are more tightly connected with each other than with nodes outside the subgraph. Detecting community is equivalent to investigating statistical properties of a graph, disregarding the roles played by specific subgraphs, and hence identify substructures/subgraph which could correspond to important functions. In the case of the World Wide Web, examples of communities are sets of Web pages dealing with the same topic [9].

In biological networks, it is widely believed that the modular structure results from evolutionary constraints and plays a crucial role in biological functions [12] [31]. In social networks, community structures correspond to human social communities [24] [22]. Finally on the Internet, community structures correspond to the autonomous systems [22], which are a connected segment of a network consisting of a collection of a collection of subnetworks interconnected by a set of routers. In the PSN we studied, community structure would correspond to human communities or some structures which are beneficial for forwarding efficiency.

Newman et al. used betweenness [26] and modularity [25] to detect community structure in complex networks. The betweenness of an edge is defined as the number of shortest paths between vertex pairs that run along it, summed over all vertex pairs. They calculate the betweenness of all edges in the network, remove the one with highest betweenness, and repeat the process until no edge remain. They also introduce a measure called *modularity* to evaluate how good a particular division is. For a division with g groups, they define a gxg matrix **e** whose component e_{ij} is the fraction of edges in the original network that connect vertices in group i to those in group j. Modularity is defined as:

$$Q = \sum_i e_{ii} - \sum_{ijk} e_{ij}e_{ki} = Tr\mathbf{e} - \|e^2\| \tag{5}$$

where $\|e^2\|$ indicates the sum of all element of e^2.

This measures the fraction of edges that are within the same community, less the expected value of the same quantity in a network with the same community division but random connection between the vertices. The difference between this algorithm and k-clique is that the k-clique approach allows overlapping community to exist, but the Newman method divides nodes into completely disjoint communities. This is the reason that we choose k-clique in our work. We have also implemented the Newman algorithm for a weighted network but for space reasons this is left to be reported in other work. For other detection methods, the recent reviews [24] and [6] may serve as introductory reading, which also include methodological overviews and comparative studies of the performance of different algorithms.

For distributed search for nodes and content in power law networks, Sarshar et al. [34] proposed using a probabilistic broadcast approach: sending out a query message to an edge with probability just above the bond percolation threshold of the network. They show that if each node caches its directory via a short random walk, then the total number of accessible contents exhibits a first-order phase transition, ensuring very high hit rates just above the percolation threshold.

For routing and forwarding in DTN and mobile ad hoc networks, there is much existing literature. Vahdat *et al* proposed the epidemic routing [38] which is similar to the "oblivious" flooding scheme we evaluated in this paper. Spray and Wait [35] is another "oblivious" flooding scheme but with a self-limited number of copies. Grossglauser *et al* proposed the two-hop relay schemes [11] to improve the capacity of dense ad hoc networks. Lindgren *et al* proposed PROPHET [21], which is a probability routing scheme based on the very early belief that community will help with routing decisions. There are also many other varied schemes such as the adaptive routing [23] by Musolesi *et al*, the practical routing scheme by Jones *et al* and Mobyspace by Leguay *et al*, these are all examples of how to use system and mobility information to improve the efficiency of routing and forwarding from "oblivious" flooding. So far, there are few empirical evaluations of the impact of community information on forwarding efficiency except a very early study by Hui *et al* [14] based on a *priori* affiliation information.

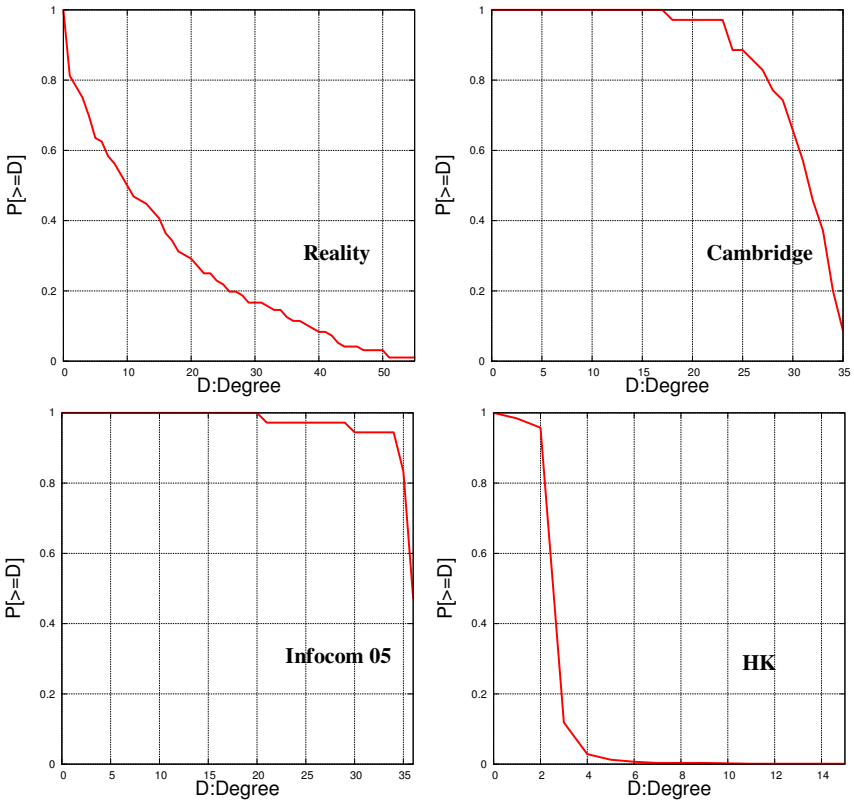


Figure 37: The degree distribution for four experiments.

13 Conclusion and future work

Networks exhibit power law node degree distributions, and scale-free networks appear to be an important model for graphs which evolve through preferential attachment and re-wiring. In this paper, we extend this modelling to mobile ad hoc and delay tolerant networks through experimental study of PSNs. We use this work to confirm the original conjecture that the use of social preferential attachment is a good heuristic for forwarding algorithms for temporal graphs in a number of ways, whether by *a priori* labels or through use of social structures inferred through observation.

A k-clique community can be built up by distributed gossipping [16]. For a complete analysis of gossipping in PSN, we model a PSN as a temporal graph with edges between two nodes that come and go following a power law distribution with certain coefficient. The power law model for edges is based on prior measurement work reported in [3] and [13]. We would like to consider several network topologies for degree attachment including simple plane lattice [16], Erdős-Rényi random graph, scale-free network and also the mobility traces we have.

Other forwarding algorithms [21] [20] have and will be devised for DTNs, and should be evaluated in the context of the mobility and social models we have described here. Use of additional resources such as geographic location data, and of infrastructural nodes to assist in forwarding must be invstigated.

In section 11.1 we chose 6 hours from the intuition that daily life is divided into 4 main periods, morning, afternoon, evening and night, each almost 6 hours. This appears to work, however, future work will look at how sensitive the system is to the choice of this period.

Current k-clique algorithm only support binary graphs, a weighted version should be targeted to eliminate the manual involvement of choosing the weight thresholds.

Further experimental work involving larger scale experiments is required to confirm our results with more confidence in a wider variety of settings. Furthermore, we believe that it should be possible to abstract mathematical models of mobility that match our empirical results that can be used to generate further data sets with which to evaluate our and other forwarding systems.

We believe that this paper represents a first step in combining rich multi-level information about social structures and interactions to drive novel and effective means for disseminating data in DTNs. A great deal of future research can follow.

14 Acknowledgement

This work was supported in part by the Haggle Project under the EU grant IST-4-027918. We would like to acknowledge comments from Steven Hand, Sid Chau, Andrea Passarella, Fernando Ramos, Eiko Yoneki, Andrew Warfield, Meng-How Lim, James Scott, and Pietro Lio. We would also like to acknowledge CRAWDAD project [18] for their hosting and sharing of the mobility data.

References

[1] L. A. Adamic, B. A. Huberman, R. M. Lukose, and A. R. Puniyani. Search in power law networks. *Physical Review E*, 64:46135–46143, October 2001.

[2] B. Bollobás. Random graphs. 2001.

[3] 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 *Proc. INFOCOM*, April 2006.

[4] H. Chang et al. An empirical approach to modeling inter-as traffic matrices. In *Proceedings of ACM Internet Measurement Conference*. ACM, 2005.

[5] H. Chang et al. To peer or not to peer: Modeling the evolution of the internet's as-level topology. In *Proceedings of IEEE Infocom*. IEEE, 2006.

[6] L. Danon, J. Duch, A. Diaz-Guilera, and A. Arenas. Comparing community structure identification, 2005.

[7] N. Eagle and A. Pentland. Reality mining: sensing complex social systems. *Personal and Ubiquitous Computing*, V10(4):255–268, May 2006.

[8] K. Fall. A delay-tolerant network architecture for challenged internets. In *Proc. SIGCOMM*, 2003.

[9] G. W. Flake, S. Lawrence, C. L. Giles, and F. Coetzee. Self-organization of the web and identification of communities. *IEEE Computer*, 35(3):66–71, 2002.

[10] L. C. Freeman. A set of measuring centrality based on betweenness. *Sociometry*, 40:35–41, 1977.

[11] M. Grossglauser and D. Tse. Mobility increases the capacity of ad-hoc wireless networks. *Transactions on Networking*, 10(4):477–486, August 2002.

[12] L. H. Hartwell, J. J. Hopfield, S. Leibler, and A. W. Murray. From molecular to modular cell biology. *Nature*, 402(6761 Suppl), December 1999.

[13] P. Hui, A. Chaintreau, J. Scott, R. Gass, J. Crowcroft, and C. Diot. Pocket switched networks and human mobility in conference environments. In *Proc. WDTN*, 2005.

[14] P. Hui and J. Crowcroft. How small lables create big improvements. In *Proc. IEEE ICMAN*, March 2007.

[15] V. Jacobson. A new way to look at networking, 2006.

[16] D. Kempe et al. Spatial gossip and resource location protocols. In *ACM Symposium on Theory of Computing*, pages 163–172, 2001.

[17] D. Kempe, J. Kleinberg, and A. Kumar. Connectivity and inference problems for temporal networks. *J. Comput. Syst. Sci.*, 64(4):820–842, 2002.

[18] D. Kotz, T. Henderson, and I. Abyzov. CRAWDAD data set dartmouth/campus (v. 2004-12-18). Downloaded from http://crawdad.cs.dartmouth.edu/dartmouth/campus, Dec. 2004.

[19] J. Leguay et al. Opportunistic content distribution in an urban setting. In *ACM CHANTS*, pages 205–212, 2006.

[20] J. Leguay, T. Friedman, and V. Conan. Evaluating mobility pattern space routing for DTNs. In *Proc. INFOCOM*, 2006.

[21] A. Lindgren, A. Doria, and O. Schelen. Probabilistic routing in intermittently connected networks. In *Proc. SAPIR*, 2004.

[22] D. Lusseau and M. E. J. Newman. Identifying the role that individual animals play in their social network. *PROC.R.SOC.LONDON B*, 271:S477, 2004.

[23] M. Musolesi, S. Hailes, and C. Mascolo. Adaptive routing for intermittently connected mobile ad hoc networks. In *Proc. WOWMOM*, 2005.

[24] M. Newman. Detecting community structure in networks. *Eur. Phys. J. B*, 38:321–330, 2004.

[25] M. E. J. Newman. Modularity and community structure in networks. *PROC.NATL.ACAD.SCI.USA*, 103:8577, 2006.

[26] M. E. J. Newman and M. Girvan. Finding and evaluating community structure in networks. *Physical Review E*, 69, February 2004.

[27] S. Okasha. Altruism, group selection and correlated interaction. *British Journal for the Philosophy of Science*, 56(4):703–725, December 2005.

[28] G. Palla, I. Dernyi, I. Farkas, and T. Vicsek. Uncovering the overlapping community structure of complex networks in nature and society. *Nature*, 435(7043):814–818, 2005.

[29] E. Paulos and E. Goodman. The familiar stranger: Anxiety, comfort, and play in public places. In *Proc. ACM SIGCHI*, 2004.

[30] P.Jaccard. *Bulletin de la Societe Vaudoise des Sciences Naturelles*, 37:547, 1901.

[31] E. Ravasz, A. L. Somera, D. A. Mongru, Z. N. Oltvai, and A. L. Barabasi. Hierarchical organization of modularity in metabolic networks. *Science*, 297(5586):1551–1555, August 2002.

[32] M. Rosvall and C. T. Bergstrom. An information-theoretic framework for resolving community structure in complex networks, Dec 2006.

[33] G. Salton. *Automatic text processing: the transformation, analysis, and retrieval of information by computer*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1989.

[34] N. Sarshar et al. Scalable percolation search in power law networks, June 2004.

[35] T. Spyropoulos, K. Psounis, and C. Raghavendra. Spray and wait: An efficient routing scheme for intermittently connected mobile n etworks. In *Proc. WDTN*, 2005.

[36] J. Su, J. Scott, P. Hui, E. Upton, M. H. Lim, C. Diot, J. Crowcroft, A. Goel, and E. de Lara. Haggle: Clean-slate networking for mobile devices. Technical Report UCAM-CL-TR-680, University of Cambridge, Computer Laboratory, Jan. 2007.

[37] A. Trusina et al. Hierarchy measures in complex networks. *Physical Review Letters*, 92:178702, 2004.

[38] A. Vahdat and D. Becker. Epidemic routing for partially connected ad hoc networks. Technical Report CS-200006, Duke University, April 2000.

[39] D. J. Watts. *Small Worlds – The Dynamics of Networks between Order and Randomneess*. Princeton University Press, Princeton, New Jersey, 1999.

[40] P. Winters. Forecasting sales by exponentially weighted moving averages. *Management Science*, 6:324–342, 1960.