

Chapter 1

Path Formation in Human Contact Networks

Nishanth Sastry and Pan Hui

Abstract The Pocket Switched Network (PSN) is a radical proposal to take advantage of short-range connectivity afforded by human face-to-face contacts, and create longer paths by having intermediate nodes ferry data on behalf of the sender. The Pocket Switched Network creates paths over time using transient social contacts. This chapter explores the achievable connectivity properties of this dynamically changing milieu, and gives a community-based heuristic to find efficient routes.

We first employ empirical traces to examine the effect of the human contact process on data delivery. Contacts between a few node pairs are found to occur too frequently, leading to inadequate mixing of data, while the majority of contacts occur rarely, but are essential for global connectivity. We then examine all successful paths found by flooding and show that though delivery times vary widely, randomly sampling a small number of paths between each source and destination is sufficient to yield a delivery time distribution close to that of flooding over all paths. Thus, despite the apparent fragility implied by the reliance on rare edges, the rate at which the network can deliver data is remarkably robust to path failures.

We then give a natural heuristic that finds routes by exploiting the latent social structure. Previous methods relied on building and updating routing tables to cope with dynamic network conditions. This has been shown to be cost ineffective due to the partial capture of transient network behavior. A more promising approach would be to capture the intrinsic characteristics of such networks and utilize them for routing decisions. We design and evaluate BUBBLE, a novel social-based forwarding algorithm, that utilizes the *centrality* and *community* metrics to enhance delivery performance. We empirically show that BUBBLE can efficiently identify good paths using several real mobility datasets.

Nishanth Sastry

Computer Laboratory, University of Cambridge, 15 JJ Thomson Avenue, Cambridge, CB3 0FD, United Kingdom. e-mail: Firstname.Lastname@cl.cam.ac.uk

Pan Hui

Deutsche Telekom Laboratories, Ernst-Reuter-Platz 7, 10587 Berlin, Germany. e-mail: pan.hui@telekom.de

1.1 Introduction

Consider a scenario in which Alice wants to convey some information to Carol. If Bob happens to meet Alice first and then Carol, he could potentially serve as a messenger for Alice’s message. Essentially, this method of communication exploits human contacts to create a path *over time* between a source and destination. Of necessity, the data paths are constructed in a store-carry-forward fashion: Various intermediate nodes *store* the data on behalf of the sender and *carry* it to another contact opportunity where they *forward* the data to the destination or another node that can take the data closer to the destination.

As normally practised, transferring information over social contacts requires manual intervention (For example, Alice requesting Bob, “When you see Carol, please tell her that. . .”) as well as a knowledge of future contacts (The human actors need to know that Bob will be meeting Carol in the future). Manual intervention can easily be avoided by automatically exchanging data between mobile devices carried by the human actors. Widely supported short-range data-transfer protocols such as bluetooth or Wi-Fi can be used for this purpose. If we do not have knowledge of future contacts, data can still be forwarded *opportunistically* from node to node, but without a guarantee that it will reach the intended destination.

This idea, of leveraging human social contacts, and using ubiquitous mobile devices in people’s pockets to opportunistically connect a source and destination over time, has been termed as a Pocket Switched Network (PSN) [16]. As a store-carry-forward network, the PSN can incur long and highly variable delays. On the other hand, it has the advantage of not requiring infrastructure setup or maintenance. It is therefore useful when infrastructure is damaged (e.g. after disasters), or does not exist (e.g. in remote areas). Also, mobility increases network capacity at the expense of delays, providing multi-user diversity gains [15]. Thus, this method can be effective as a multi-hop “sneakernet” for high-bandwidth applications that can tolerate delays.

The question remains as to how “well” the transient, local contacts can support a wider connectivity across the network, if we do not have knowledge of future contacts. The first part of the chapter explores this issue by studying two traces that recorded human contacts over extended time periods. We measure the achievable performance of the contact network over a given time window in terms of the fraction of data delivered (delivery ratio), as well as the time to delivery. The delivery ratio at the end of a time window is indicative of the fraction of node pairs connected during the window and is therefore a measure of the connectivity achieved by the network. The empirically observed cumulative distribution of delivery times can also be interpreted as the evolution in time of delivery ratio, normalised by the ratio eventually achieved at the end of the time window¹, and thus represents the rate at which connectivity is achieved.

¹ If the empirical probability that the delivery time is less than t is r , then a fraction r of the data that eventually get delivered have been delivered by time t .

We find that rare contacts are crucial for connectivity: when contacts between node-pairs which meet infrequently are removed, the network breaks apart into smaller, unconnected components. Note that in order for any path to succeed, all the contacts in the path need to occur successfully, and in the right temporal order. Thus, if one of the rare contacts involved in a path does not happen or succeed in transferring data, the path will fail. Thus, this result seems to imply a fragility in the network of social contacts.

Note that failures of individual paths does not automatically imply failure to achieve connectivity. There can be several paths between a source and destination, and if at least one path succeeds, connectivity is achieved. However, it might take longer to reach the destination if some of the quicker paths fail. We study the degradation of delivery times by examining the impact of random path failures among paths found by flooding data. Specifically, we look at two modes of path failures. The first, *proportional flooding*, assumes that a fixed fraction $\mu < 1$ of the paths between every sender-destination pair succeeds. The second, *k-copy forwarding*, allows at most k paths to succeed between each sender and destination. In both cases, we find that the time to achieve connectivity is remarkably resilient in that the distribution of delivery times with a large number of path failures remains close to the delivery times achieved when there are no path failures.

In the second part of the chapter, we address the question of how to route data from a given sender to a destination and give a social-based heuristic for finding “good” routes. Many MANET and some DTN routing algorithms [23] [1] provide forwarding by building and updating routing tables whenever mobility occurs. This approach is not appropriate for a PSN, since mobility is often unpredictable, and topology structure is highly dynamic. We need an algorithm which can cope with dynamic, repeated disconnection and re-wiring. Rather than exchange much control traffic to create unreliable routing structures, which may only capture the “noise” of the network, we prefer to search for some characteristics of the network which are less volatile than mobility. A PSN is formed by people. Hence, social metrics are intrinsic properties to guide data forwarding in such kinds of human networks.

In this context, we introduce an social-based forwarding algorithm, BUBBLE, which focuses on two key social metrics: community and centrality. Co-operation binds, but also divides human society into communities. For an ecological community, the idea of correlated interaction means that an organism of a given type is more likely to interact with another organism of the same type than with a randomly chosen member of the population [34]. This correlated interaction concept also applies to humans, so we can exploit this kind of community information to select forwarding paths. Within a community, some people are more popular, and interact with more people than others (i.e., have high *centrality*); we call them hubs. In this chapter, we will show how community and centrality help us to efficiently identify the “good” forwarding paths.

The rest of this chapter will be presented in two parts: In Part 1.2, we will study the feasibility of path formation in human contact networks. In Part 1.3, we will discuss a community-based routing strategy called BUBBLE, and evaluate its performance. Sec. 1.4 discusses related work and Sec. 1.5 concludes.

1.2 Feasibility of path formation

Our first goal is to study the extent to which the temporally changing network of human face-to-face contacts can support $N \times N$ connectivity. Since our goal is to explore feasibility in a future PSN, we employ traces drawn from “naturalistic” settings, lasting at least a month long. The contacts in these traces are highly heterogeneous, with some nodes meeting each other hundreds of times and others meeting fewer than ten times.

Our main results are as follows: Nodes which meet each other frequently turn out to be inefficient for data transfer. Intuitively, nodes which meet each other too often do not have new data to exchange during their second and subsequent contacts. In contrast, the rarely occurring contacts are more effective at “mixing” data and are crucial for reachability. Since the rare contacts do not, by definition, recur often, paths which rely on them are sensitive to chance events which either prevent the contact from occurring or prevents the data from being transferred during the contact. However, we show that connectivity of the network as a whole is not greatly affected by individual path failures. Specifically, we study two different modes of path failures and show that the distribution of delivery times remains close to optimal despite a large number of paths failing.

1.2.1 Setup and Methodology

This section motivates the choice of the traces used in the first part of the chapter, the simulation setup and the performance measures used to study feasibility of path formation.

1.2.1.1 Traces

We imagine the participants of a PSN would be a finite group of people who are at least loosely bound together by some context—for instance, first responders at a disaster situation, who need to send data to each other. Multiple PSNs could co-exist for different contexts, and a single individual could conceivably participate in several different PSNs².

Our model that PSN participants form a cohesive group places the requirement that an ideal PSN should be able to create paths between arbitrary source-destination pairs. This is reflected in our simulation setup, where the destinations for each source node are chosen randomly. Also, our traces are picked to be close to the

² Note that this is in contrast to a single unboundedly large network of socially unrelated individuals as in the famous “small-world” experiment [41] that examined a network essentially comprising all Americans and discovered an average 5.2 (≈ 6) degrees of separation.

limits of Dunbar’s number ($=147.8$, 95% confidence limits: 100.2–231.1), the average size for cohesive groups of humans [7].

The first trace comes from a four week subset of the UCSD Wireless Topology Discovery [42] project which recorded Wi-Fi Access Points seen by subjects’ PDAs. We treat PDAs simultaneously in range of the same Wi-Fi access point as a contact opportunity. This dataset traces contacts between $N = 202$ subjects. The second trace consists of bluetooth contacts recorded from 1 Nov. 2004 to 1 Jan. 2005 between participants of the MIT Reality Mining project [9]. We conservatively set five minutes as the minimum allowed data transfer opportunity and discarded contacts of durations smaller than this cutoff. This trace has contacts between $N = 91$ subjects.

The subjects in the MIT trace consist of a mixture of students and faculty at the MIT Media Lab, and incoming freshmen at the MIT Sloan Business School. The UCSD trace is comprised of a select group of freshmen, all from UCSD’s Sixth College. As such, we can expect subjects in both traces to have reasons for some amount of interaction, leading to a loosely cohesive group structure. Prior work on community mining using the same traces supports this expectation [44].

It is important to emphasize that our focus is solely on the capability and efficiency of the human contact process in forming end-to-end paths. The precise choice of the minimum data transfer opportunity is less important—it is entirely possible that a new technology would allow for faster node-node transfers. Indeed, our results are qualitatively similar for other cutoff values tested. Similarly, a different technology for local node-node transfers could have different “reach,” allowing more nodes to be in contact with each other simultaneously. Nevertheless, the substantial similarities (see rest of this section) between results based on two different technologies and traces—the Wi-Fi based UCSD trace and the bluetooth based MIT trace—gives us some confidence that the results below may be applicable beyond the traces and technologies we have considered.

Both our traces were chosen to be at least one month long, in order to obtain multiple disjoint time windows over which to test the relevance of our results.

1.2.1.2 Simulation setup and measurement

Setup: At the beginning of simulation, data is created, marked for a randomly chosen destination, and associated with the source node. An oracle with complete knowledge of the future can choose to transfer data at appropriate contact opportunities and thereby form the quickest path to the destination. To simulate this, we enumerate all possible paths found by flooding data at each contact opportunity, and choose the quickest.

Performance measure: Consider the time-ordered sequence (with ties broken arbitrarily) of contacts that occur globally in the network. Since there are $N(N - 1)$ quickest paths between different sender-destination pairs, a maximum³ of $N(N - 1)$

³ The actual number could be lesser because a contact with a rarely active node could complete multiple paths that end in that node.

contacts in the the global sequence of contacts act as path completion points. Of these, Nd become “interesting” when there are d destinations per sender. Since the destinations are chosen randomly, we might expect that on average, if k path completion points have occurred, the *fraction* of these that are interesting is independent of d : When d is greater, more data gets delivered after k path completion points, but there is also more data to deliver.

The above discussion motivates our method of measuring the efficiency of the PSN: At any point in the simulation, the *delivery ratio*, measured as the fraction of data that has been delivered, or equivalently, the number of “interesting” path completion points we have seen, is taken as a figure of merit. The more efficient the PSN is, the faster the delivery ratio evolves to 1, as the number of contacts and time increase.

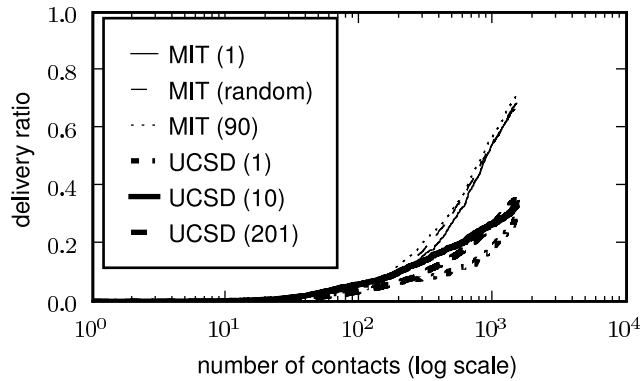


Fig. 1.1 Fraction of data delivered as a function of the number of contacts, for the MIT and UCSD traces (number of destinations per sender shown in brackets). The curves for each network are clustered together, showing that the delivery ratio evolves independently of the load.

Unless otherwise specified, our experiments examine delivery ratio evolution statistically averaged over 10 independent runs, with each run starting at a random point in the trace, and lasting for 6000 contacts. We confirm our intuition in Fig. 1.1, which shows that the delivery ratio evolves similarly, whether d is 1 or a maximum of $N - 1$ destinations per sender. We note that the graph also represents the fastest possible evolution of the delivery ratio under the given set of contacts, due to the use of flooding.

1.2.2 Order and distribution of contacts

A PSN contact trace is determined by the distribution of contact occurrences and the time order in which these contacts occur. In this section, we examine how these properties affect delivery ratio evolution.

Given two traces, the more efficient one will manage to achieve a given delivery ratio with fewer number of contacts. Our approach is to create a synthetic trace from the original trace by disrupting the property we wish to study. Comparing delivery ratio evolution in the original and synthetic traces informs us about the effects of the property.

Our main findings are that in both the traces we examine, time correlations between contacts that occur too frequently leads to non-effective contacts in which no new data can be exchanged, and that the progress of the delivery ratio as well as the connectivity of the PSN itself are precariously dependent on rare contacts.

1.2.2.1 Frequent contacts are often non-effective

To investigate the effect of the time order in which contacts occur, we replay the trace, randomly shuffling the time order in which links occur. Observe in Fig. 1.2 that the curve marked “shuffled” evolves faster than “trace” implying that the delivery ratio increases faster after random shuffling. The random shuffle has the effect of removing any time correlations of contacts in the original trace. Thus the improved delivery ratio evolution implies that time correlations of the contacts in the original data slowed down the exchange of data among the nodes, causing them to be delivered later.

Manual examination reveals several time correlated contacts where two nodes see each other multiple times without seeing other nodes. At their first contact, one or both nodes could have data that the other does not, which is then shared by flooding. After this initial flooding, both nodes contain the same data—subsequent contacts are “non-effective”, and only increase the number of contacts happening in the network without increasing the delivery ratio.

To quantify the impact, in the curve marked “effective” on Fig. 1.2, we plot delivery ratio evolution in the original trace, counting only the contacts in which data could be exchanged. This coincides well with the time-shuffled trace, showing that non-effective contacts are largely responsible for the slower delivery ratio evolution in the original trace.

Next, we construct a synthetic trace that has the same number of nodes as the original trace, as well as the same contact occurrence distribution. By this, we mean that the probability of contact between any pair of nodes is the same as in the original trace. The delivery ratio evolution of this trace, depicted as “link distr” in Fig. 1.2, is seen to evolve in a similar fashion as the time-shuffled trace. This indicates that once time correlations are removed, the delivery properties are determined mainly by the contact occurrence distribution.

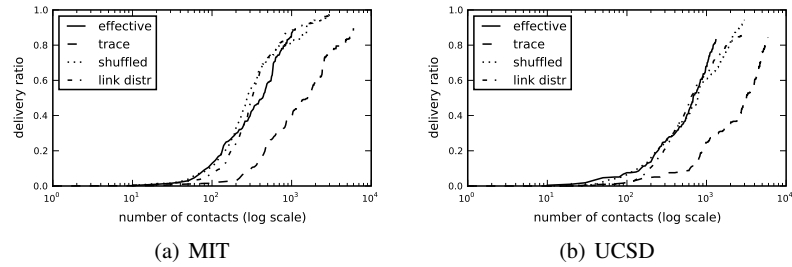


Fig. 1.2 Delivery ratio evolution for synthetically derived variants of MIT and UCSD traces. ‘Trace’ is the original. ‘Shuffled’, the same trace with time order of contacts randomly shuffled. ‘Effective’ replays ‘trace’, counting only contacts where data was exchanged. ‘Link distr’ is an artificial trace with the same size and contact occurrence distribution as the original.

1.2.2.2 Connectivity depends on rare contacts

The fact that three different traces (shuffled, effective, and link distr), which are based on the same contact occurrence distribution, essentially evolve in the same manner leads us to examine this distribution further.

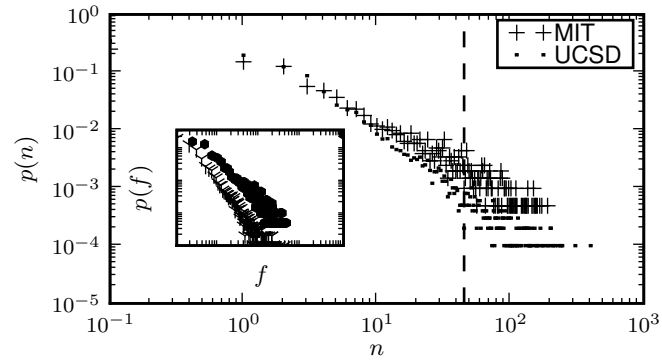


Fig. 1.3 Contact occurrence distributions (log-log): A random edge appears n times with probability $p(n)$. To the left of the dashed line at $n = 45$, the distributions for both traces coincidentally happen to be similar. The inset shows the difference when normalised by the number of contacts in the trace. In the inset, a random edge constitutes a fraction f of the trace with probability $p(f)$.

Fig. 1.3 shows that the contact occurrence distribution has both highly rare contacts (involving node pairs that meet fewer than ten times in the trace) as well as frequent contacts (nodes which meet hundreds of times). A randomly chosen contact from the trace is much more likely to be a rare contact than a frequent one.

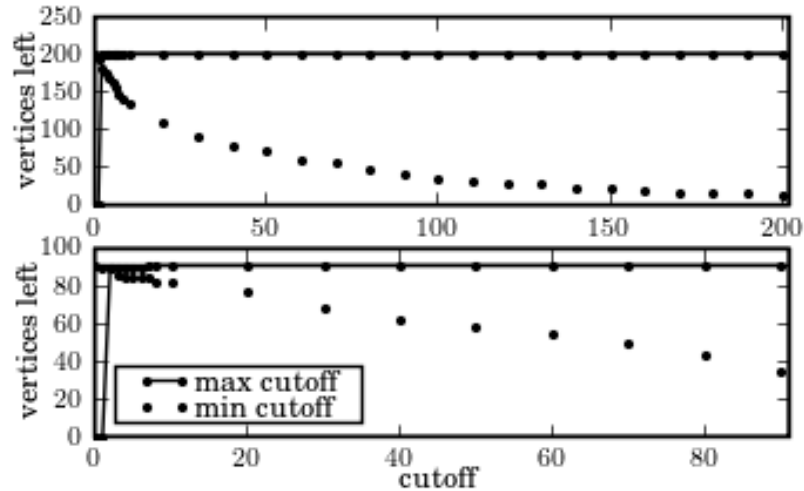


Fig. 1.4 Robustness to cutoff: MIT (below), UCSD (above). Max cutoff specifies a maximum cutoff for the frequency of contacts, thus removing the most frequently occurring ones. Min cutoff specifies a minimum frequency of contacts—removing the rarest contacts causes the number of nodes that are connected to drop precipitously.

Fig. 1.4 shows that the rare contacts are extremely important for the nodes to stay connected. When contacts that occur fewer than a minimum cutoff number of times are removed, the number of nodes remaining in the trace falls sharply. This implies that there are a number of nodes which are connected to the rest of the nodes by only a few rare contacts.

On the other hand, removing the frequent contacts (by removing contacts occurring more than a maximum cutoff number of times) does not affect connectivity greatly. For instance, the MIT trace remains connected even when the maximum cutoff is as low as 10 (i.e., contacts occurring more than ten times are removed). This suggests that nodes which contact each other very frequently are also connected by other paths, comprising only rare edges.

1.2.3 Resilience to Path Failures

In order for a path to succeed in a temporally changing network like the PSN, all edges have to occur in the right order. Therefore, the reliance on rare edges, as shown in the previous section, could lead to a large number of path failures. Note that this does not automatically imply bad connectivity. Since only one path between every source-destination pair needs to succeed for data delivery, individual path failures do not greatly impact the delivery ratio achieved at the end of a time window (unless all paths between a source-destination pair fail, disconnecting the network).

However, path failures can affect the rate at which the delivery ratio evolves: Suppose the quickest path between a pair of nodes would have arrived at t_1 , but cannot be used because of a failure. If the first usable path connects the nodes at time $t_2 > t_1$, then between t_2 and t_1 the fraction of data delivered is decreased on account of the path failure. In other words, there is a delay in data delivery, which temporarily shifts the cumulative distribution of delivery times to the right.

This section looks at the effects of path failures by studying the effects of failures on paths found by flooding. Given a sequence of contacts, flooding achieves the best possible delivery times by exploring *every* contact opportunity and thereby finding the path with the *minimum* path delay.

We look at instances of the PSN over fixed time-windows and wish to study the degradation in the delivery time distribution when not all of the paths found by flooding can be explored. Specifically, we study two failure modes: The first, proportional flooding, explores a fixed fraction μ of the paths found by flooding between each source and destination. We show that a constant increase in the fraction of paths explored brings the delivery time distribution of proportional flooding exponentially closer to that of flooding over all paths. The second failure mode, k -copy flooding, explores no more than a fixed number $k > 1$ of the paths found by flooding between each source and destination. Again, a constant increase in k brings the delivery time distribution exponentially close to the optimal delivery time distribution of flooding all paths. Empirically, even small values of k (e.g., $k = 2$ or $k = 5$) closely approximate delivery times found by flooding.

The results of this section imply that the human contact network is remarkably resilient to path failures and the delivery ratio evolves at a close-to-optimal rate even when the majority of paths fail and only a small fraction or a small, bounded number of paths can transport data to the destination. Note that we only admit paths from the original flood-tree, and do not include new paths that repair failures by joining the affected nodes to the flood tree at later contacts. Thus our results in fact underestimate the resilience of the network.

The success of k -copy flooding can provide a loose motivation for routing algorithms that use multiple paths between each sender and destination pair since this could obtain a close-to-optimal delivery time distribution. However, heuristics-based routing algorithms may not find the same paths as found by flooding. Thus, the correspondence is not exact.

1.2.3.1 Path delay distribution D

[39, 14, 45] model the performance of epidemic routing and its variants and derive a closed form for delivery time distribution, showing it to be accurate for certain common mobility models. However, several simplifying assumptions are made, including an exponential inter-contact time between node pairs. Unfortunately, human contact networks are known to have power law inter-contact times with exponential tails [4, 24]. Furthermore, [39, 14, 45] use a constant (averaged) contact rate, whereas the contact rates in our empirical traces are highly heterogeneous (see

Sec. 1.2.2). Plugging in the mean contact rate from our empirical traces into their expressions yields bad fits.

H	Hop delay, or time to next hop. Time until a path expands by one more node.
N	Number of edges per path. $N \sim \text{Poisson}(\text{mean} = \lambda)$
L	Number of paths between a random src-dest pair.
D	Path delay for a random path
D^*	Delivery time (minimum path delay across all paths between a random source-destination pair)
$G_X(s)$	Probability-generating function of X
$M_X(s)$	Moment-generating function of X . $M_X(s) = G_X(e^s)$

Table 1.1 Summary of notation used to characterise components of path delay and delivery times

Thus, in order to obtain a handle on delays incurred on paths, we take a very coarse grained and simplified approach. In particular, we only assume that path delays (D) can be treated as being independent of each other, that the distribution of time to next hop (H) can be described by a moment-generating function $M_H(s)$, and that the number of hops (N) on the paths found by flooding follows a Poisson distribution with mean λ . See Table 1.1 for a full summary of our notation.

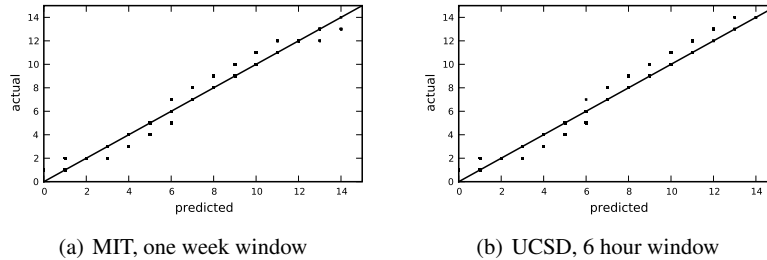


Fig. 1.5 Number of hops follows the Poisson Distribution. Each Q-Q plot shows fit through correspondence between sample deviates generated according to the theoretical distribution (predicted) and empirical (actual) values. Closeness to predicted=actual diagonal indicates better fit. Different combinations of trace and time window sizes are used to show generality of fit.

Our most specialized assumption is that the number of hops on a path formed by flooding during a fixed time-window follows the Poisson distribution. This is justified by a surprisingly good fit in our empirical traces (Fig. 1.5). We conjecture that this is a result of several factors which work together to limit the number of hops in a successful path. First, we only consider paths that form during a fixed time window. Second, the small-world nature of the human contact graph makes for short paths to a destination; and paths are frozen at the destination because the destination does not forward data further. Third, each node can join the flood-tree at most once. As the tree grows, the number of nodes available to grow the tree and extend a path decreases. Thus extremely long paths are rare.

Using the above assumptions, the path delay D can be written as:

$$M_D(s) = G_N(M_H(s)) \quad (1.1)$$

We can apply a Chernoff-type bound and write

$$P[D \geq t] \leq \min_{s>0} e^{-st} M_D(s) = \min_{s>0} e^{\lambda(M_H(s)) - st} = \exp(F_H(t)) \quad (1.2)$$

where $F_H(t) = \lambda M_H(s_{\min}(t)) - s_{\min}(t)t - \lambda$ and $s_{\min}(t)$ minimises s in the Chernoff bound.

1.2.3.2 Proportional Flooding

Consider an arbitrary source-destination pair. As described previously, we will model the path delays between them as being chosen independently and identically from the distribution in (1.1). Suppose copies of the data are sent along l randomly chosen paths between them. The obtained delivery time D_l^* is the *minimum* of the path delays across all l paths. Using (1.2) we can write

$$P[D_l^* \leq t] = 1 - \prod_{i=1}^l P[D \geq t] \geq 1 - e^{-lF_H(t)} \quad (1.3)$$

Note that the above assumes that the l path delays are independent. In reality, paths found by flooding all fan out from a single source node, and the first few hops, close to the source, are typically shared with other paths, violating the independence assumption. Therefore, the model in this section is to be considered only as a simple formulation designed to gain insight into proportional flooding. It is worth mentioning however that in the empirical data sets, we frequently find that the major component of path delay is contributed by the part of the paths closest to the destination, which are not shared with other paths. Also, in the case when only a few paths on the flood-tree are being randomly sampled, the number of hops shared is limited.

Consider source-destination pairs with $L = m$ paths connecting them. Full flooding finds the quickest of all m paths and obtains a delivery time distribution $P[D_L^* \leq t | L = m]$. Proportional flooding chooses a fraction μ of them. From (1.3), the difference $\Delta(t; \mu)$ in the delivery time distributions between full and proportional flooding, is upper bounded by

$$\Delta(t; \mu) \leq P[D_L^* \leq t | L = m] - 1 + e^{-\mu m F_H(t)} \quad (1.4)$$

Remark 1. A constant increase in μ has an exponential effect on Δ : For any t , if μ is increased by some constant, the fraction of data delivered by proportional flooding during $[0, t]$ becomes exponentially closer to that delivered by full flooding. Thus, proportional flooding quickly becomes very effective as μ is increased.

The exponential decrease in Δ with a constant increase in μ is obtained as long as $F_H(t) < 0$. In other words, our results hold when there are a Poisson number of hops in paths formed over fixed time windows, for any hop delay distribution H that has a moment generating function and satisfies $F_H(t) < 0$.

Also, since

$$\frac{\partial \Delta}{\partial \mu} = m F_H(t) e^{\mu m F_H(t)} < 0,$$

Δ decreases when μ is increased. Furthermore, the rate of decrease is higher for smaller μ – increasing μ from $\mu = 0.1$ to $\mu = 0.2$ results in a greater decrease than an increase from $\mu = 0.6$ to $\mu = 0.7$.

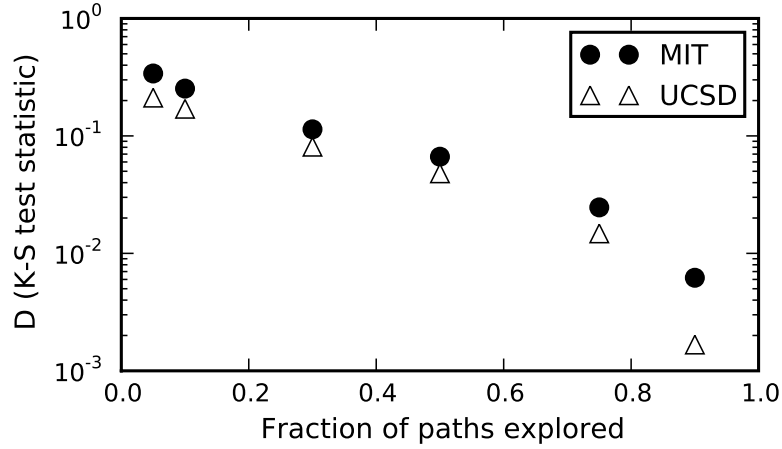


Fig. 1.6 K-S statistic (D) measuring the difference between the delivery time distributions of full flooding and proportional flooding for different μ . X-axis is linear, Y-axis is log-scale.

Fig. 1.6 empirically shows the difference between $D^*(t)$, the delivery time distribution obtained by flooding over all paths, and $D_\mu^*(t)$, the delivery time distribution for proportional flooding using a randomly selected fraction μ of paths between every source and destination. The difference is measured using the Kolmogorov-Smirnov statistic given by $D = \max_t (D^*(t) - D_\mu^*(t))$. Note that the Y-axis is log scale; a constant increase in μ shows an exponential decrease in D .

1.2.3.3 From proportional to bounded number of paths

Proportional flooding offers a mechanism to gracefully degrade from full flooding by exploring a fraction of the paths. However, in the worst case, there can be up to $N - 1$ paths to a destination in a N node PSN, and proportional flooding can be costly. This leads us to define a bounded cost strategy that explores at most a small,

fixed number, k of the paths to a destination, and still achieves delivery times similar to that of proportional flooding. Unlike proportional flooding, k -copy flooding explicitly limits the number of paths explored, and therefore can tolerate a larger number of path failures in the worst case, when there are a large number of paths between a node-pair.

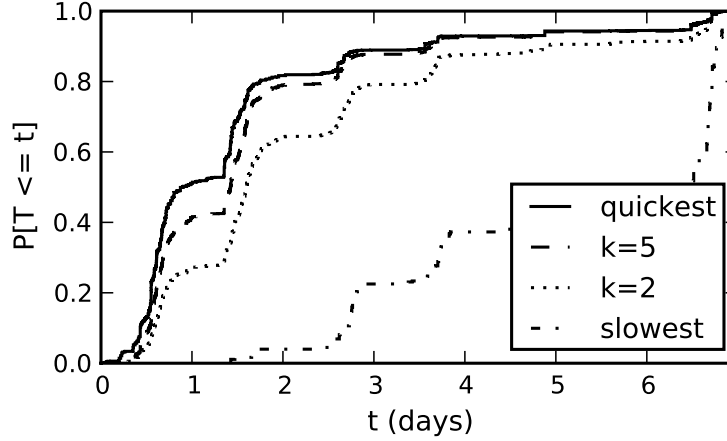


Fig. 1.7 k -copy flooding: Nodes are connected by multiple paths with different delays (CDFs of the *quickest* and *slowest* are shown). Yet, randomly choosing at most k of the paths to each destination closely approximates the quickest, even for small k . (MIT trace, one week window)

Fig. 1.7 shows empirically that in our data sets, even for small k ($= 2, 5$), the delivery time distribution of k -copy flooding starts to closely approximate full flooding. To see why, consider the *equivalent fraction* μ_k of paths in proportional flooding that gives the same expected number of paths as k -copy forwarding:

$$\sum_{l=0}^k lP[L=l] + kP[L>k] = \mu_k \mathbb{E}[L] \quad (1.5)$$

Suppose k is increased by a constant h , resulting in a new equivalent fraction μ_{k+h} . (1.5) becomes

$$\sum_{l=0}^k lP[L=l] + \sum_{j=1}^h (k+j)P[L=k+j] + (k+h)P[L>k+h] = \mu_{k+h} \mathbb{E}[L]$$

Regrouping, we get

$$\begin{aligned} & \sum_{l=0}^k lP[L=l] + k \left(\sum_{j=1}^h P[L=k+j] + P[L > k+h] \right) \\ & + \sum_{j=1}^h jP[L=k+j] + hP[L > k+h] = \mu_{k+h} \mathbb{E}[L] \end{aligned}$$

Comparing with (1.5), we can write

$$\mu_k \mathbb{E}[L] + \sum_{j=1}^h jP[L=k+j] + hP[L > k+h] = \mu_{k+h} \mathbb{E}[L]$$

Thus the *increase* in the equivalent fraction of paths is

$$\begin{aligned} \mu_{k+h} - \mu_k & \geq \frac{h}{\mathbb{E}[L]} \left(\sum_{j=1}^h P[L=k+j] + P[L > k+h] \right) \\ & = h(P[L > k] / \mathbb{E}[L]) \end{aligned} \quad (1.6)$$

Remark 2. A constant increase in k is equivalent to at least a (scaled) constant increase in the fraction of paths explored by proportional flooding. Thus, as a simple consequence of Remark 1, a constant increase in the number of paths explored in k -copy forwarding moves its delivery time distribution exponentially closer to that of full flooding.

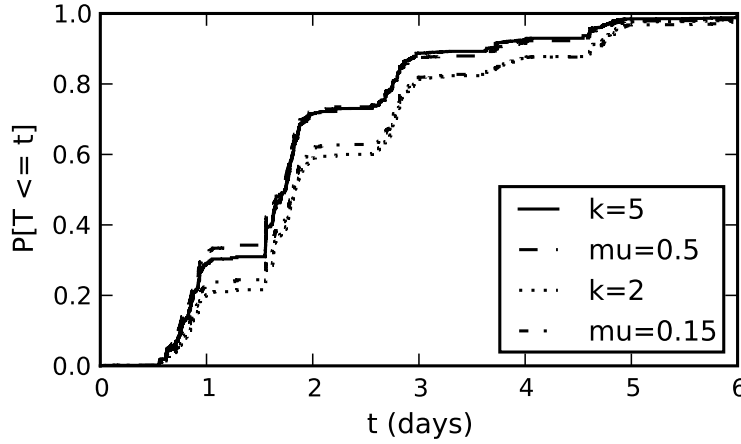


Fig. 1.8 Proportional flooding with $\mu_2 = 0.15$ of paths has similar delivery times as $k = 2$ -copy routing. Similarly $k = 5$ corresponds to $\mu_5 = 0.5$. (MIT, one week window)

This explains why exploring at most a small number k of paths has a delivery time distribution approaching that of flooding over all paths. Fig. 1.8 empirically shows the equivalent fractions μ_k for the $k = 2$ and $k = 5$ cases discussed previously.

1.3 BUBBLE : a community based routing algorithm

After establishing the feasibility of $N \times N$ connectivity, in this second part of the chapter, we present a concrete mechanism to route data over human contacts. The key insight is to exploit the structure inherent in human social contacts. Bubble leverages the heterogeneity in popularity—certain individuals, such as a postman, are likely to have contacts with many different persons and are therefore useful in bridging disjoint nodes. In BUBBLE, messages “bubble” up and down the social hierarchy in order to reach the destination. Messages traverse the hierarchy, using the highly central nodes to bridge data between disjoint communities where necessary, until they reach the destination.

1.3.1 Traces

For evaluating BUBBLE, we use four experimental datasets gathered by the Huggle Project⁴ over two years, referred to as *Infocom05*, *HongKong*, *Cambridge*, *Infocom06* and one dataset from the MIT Reality Mining Project [8], referred to as *Reality*. Previously, the characteristics of these datasets such as inter-contact and contact distribution have been explored in several studies [3] [25] [28], to which we refer the reader for further background information.

- 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.).
- 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. This dataset covers 11 days.
- 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.

⁴ <http://www.huggleproject.org>

Experimental data set	Infocom05	Hong-Kong	Cambridge	Infocom06	Reality
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 1.2 Characteristics of the five experimental data sets

- 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.

The five experiments are summarised in Table 1.2. A remark about the datasets is that the experiments do not have the same granularity and the finest granularity is limited to 120 seconds. This is because of the trade-off between the duration of the experiments and the accuracy of the samplings.

The four Huggle datasets were chosen to allow us greater insight into the actual (ground truth) community structure, whereas the *Reality* dataset is used to demonstrate that BUBBLE is robust to inferred community structure as well.

1.3.2 Inferring Human Communities

In a PSN, the social network could map to the computer network since people carry the computing devices. In this section, we introduce and evaluate two centralised community detection algorithms: *K-CLIQUE* by Palla *et al.* [35] and weighted network analysis (WNA) by Newman [31]. We use these two centralised algorithms to uncover the community structures in the mobile traces. We believe our evaluation of these algorithms can be useful for future traces gathered by the research community.

Many centralised community detection methods have been proposed and examined in the literature (see the review papers by Newman [32] and Danon *et al.* [6]). The criteria we use to select a centralised detection method are the ability to uncover overlapping communities, and a high degree of automation (low manual involvement). In real human societies, one person may belong to multiple communities and hence it is important to be able to detect this feature. The *K-CLIQUE* method satisfies this requirement, but was designed for binary graphs, thus we must threshold the edges of the contact graphs in our mobility traces to use this method, and it is difficult to choose an optimum threshold manually [35]. On the other hand, (WNA) can work on weighted graphs directly, and does not need thresholding, but it can-

not detect overlapping communities [31]. Thus we chose to use both *K-CLIQUE* and *WNA*; they each have useful features that complement one another.

1.3.2.1 Contact Graphs

In order to help us to present the mobility traces and make it easier for further processing, we introduce the notion of a *contact graph*. 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, the edges are the contacts, and the weights of the edges are the values based on the metrics specified such as the number of contacts during the experiment. We can measure the relationship between two people by how many times they meet each other and how long they stay with each other. We naturally think that if two people spend more time together or see each other more often, they are in a closer relationship.

First, we find the distribution of contact durations and number of contacts for the two conference scenarios are quite similar. To prevent redundancy, in the later sections we only selectively show one example, in most cases *Infocom06*, since it contains more participants.

Figure 1.9 and Figure 1.10 show the contact duration and number of contacts distribution for each pair in four experiments. For the *HongKong* experiment we include the external device because of the network sparseness, but for the other three experiments we use only the internal devices. These contact graphs created are used for the community detection in the following subsections.

1.3.2.2 K-CLIQUE Community Detection

Palla *et al.* [35] 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. 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. We have applied this on all the datasets above. Figure 1.11 shows the 3-clique communities in the *Infocom06* dataset. More detailed descriptions about the *k*-clique communities on these datasets can be found in our previous work [18] [19].

1.3.2.3 Weighted Network Analysis

In this section, we implement and apply Newman's weighted network analysis (*WNA*) for our data analysis [31]. This is an extension of the unweighted *modularity* method proposed in [33] to a weighted version. We use this as a measurement of the fitness of the communities it detects.

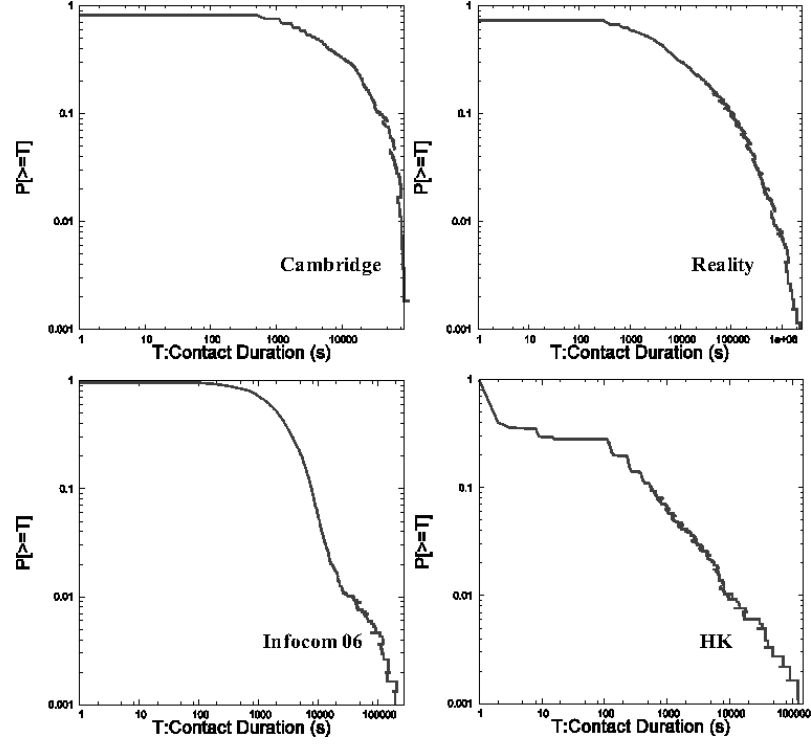


Fig. 1.9 The distribution of pair-wise contact durations

For each community partitioning of a network, one can compute the corresponding modularity value using the following definition of *modularity* (Q):

$$Q = \sum_{vw} \left[\frac{A_{vw}}{2m} - \frac{k_v k_w}{(2m)^2} \right] \delta(c_v, c_w) \quad (1.7)$$

where A_{vw} is the value of the weight of the edge between vertices v and w , if such an edge exists, and 0 otherwise; the δ -function $\delta(i, j)$ is 1 if $i = j$ and 0 otherwise; $m = \frac{1}{2} \sum_{vw} A_{vw}$; k_v is the degree of vertex v defined as $\sum_w A_{vw}$; and c_i denotes the community vertex i belongs to. *Modularity* is defined as the difference between this fraction and, the fraction of the edges that would be expected to fall within the communities if the edges were assigned randomly, but we keep the degrees of the vertices unchanged. The algorithm is essentially a genetic algorithm, using the modularity as the measurement of fitness. Rather than selecting and mutating current best solutions, we enumerate all possible merges of any two communities in the current solution, and evaluate the relative fitness of the resulting merges, and choose the best solution as the seed for the next iteration.

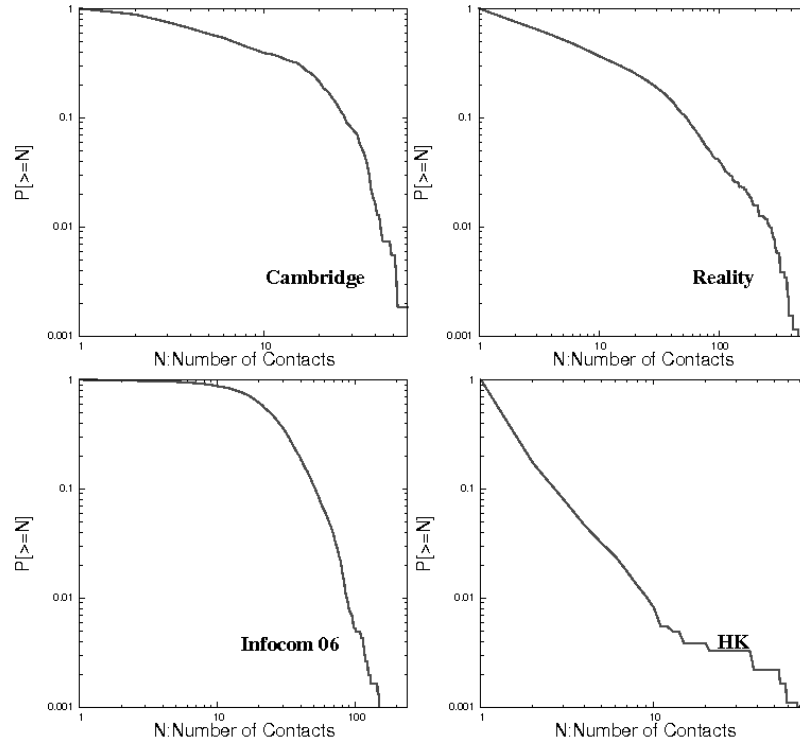


Fig. 1.10 The distribution of pair-wise number of contacts

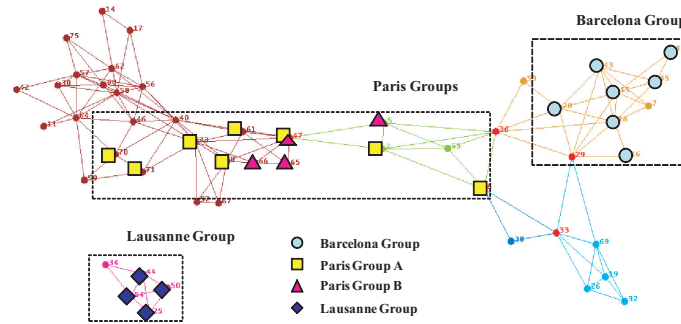


Fig. 1.11 3-clique communities based on contact durations with weight threshold that equals 20 000 s (Infocom06; circles, Barcelona group; squares, Paris group A; triangles, Paris group B; diamonds, Lausanne group)

Table 1.3 summarises the communities detected by applying WNA on the four datasets. According to Newman [31], non-zero Q values indicate deviations from randomness; values around 0.3 or more usually indicate good divisions. For the *Infocom06* case, the Q_{max} value is low; this indicates that the community partition is not very good in this case. This also agrees with the fact that in a conference the community boundary becomes blurred. For the *Reality* case, the Q value is high; this reflects the more diverse campus environment. For the *Cambridge* data, the two groups spawned by WNA is exactly matched the two groups (1st year and 2nd year) of students selected for the experiment.

Dataset	Info06	Camb	Reality	HK
Q_{max}	0.2280	0.4227	0.5682	0.6439
Max. Community Size	13	18	23	139
No. Communities	4	2	8	19
Avg. Community Size	8.000	16.500	9.875	45.684
No. Community Nodes	32	33	73	868
Total No. of Nodes	78	36	97	868

Table 1.3 Communities detected from the four datasets

These centralised community detection algorithms give us rich information about the human social clustering and are useful for offline data analysis on mobility traces collected. We can use them to explore structures in the data and hence design useful forwarding strategies, security measures, and killer applications.

1.3.3 Heterogeneity in Centrality

In human society, 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 [18]. Here, we want to employ heterogeneity in popularity to help design more efficient forwarding strategies: we prefer to choose popular hubs as relays rather than unpopular ones.

A temporal network or time-evolving network is a kind of weighted network. The centrality measure in traditional weighted networks may not work here since the edges are not necessarily concurrent (i.e., the network is dynamic and edges are time-dependent). Hence we need a different way to calculate the centrality of each node in the system. Our approach is as follows:

1. Carry out a large number of emulations of unlimited flooding with different uniformly distributed traffic patterns created.
2. 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 the same message is delivered to the destination through different paths, where we only count the delivery with the shortest delay.

We call the number calculated above the *betweenness centrality* of this node in this temporal graph. Of course, it can be normalised to the highest value found. Here we use unlimited flooding since it can explore the largest range of delivery alternatives with the shortest delay. This definition captures the spirit of Freeman centrality [13].

For the emulation, we developed an emulator called *HaggleSim* [17], which can replay the mobility traces collected and emulate different forwarding strategies on every contact event. This emulator is driven by contact events. The original trace files are divided into discrete sequential contact events, and fed into the emulator as inputs. In all the simulations for the BUBBLE algorithm (including the evaluations in Section 1.3.4), we divided the traces into discrete contact events with granularity of 100 seconds. Our emulator reads the file line by line, treating each line as a discrete encounter event, and makes a forwarding decision on this encounter based on the forwarding algorithm under study.

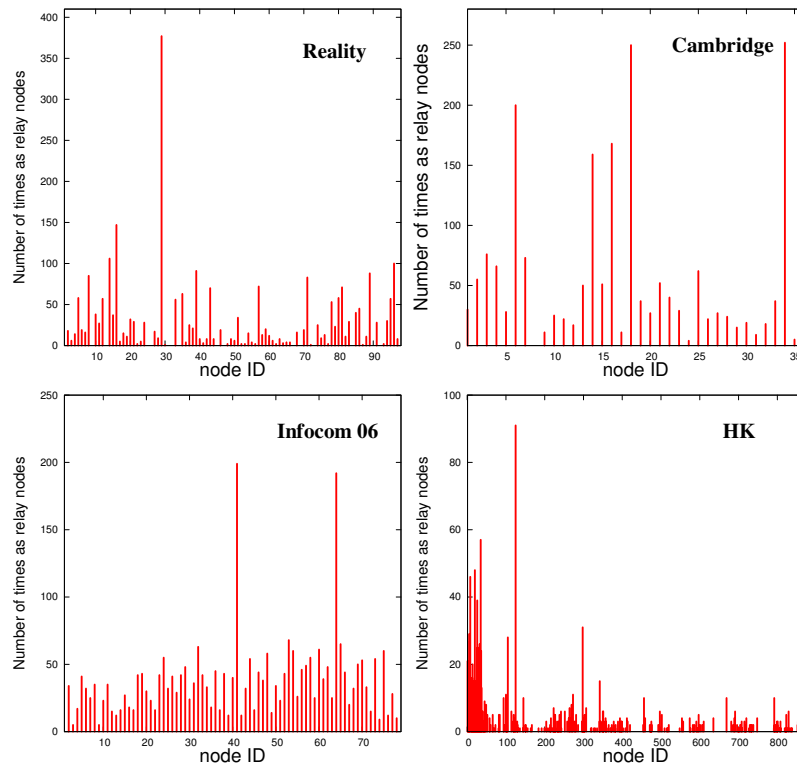


Fig. 1.12 Number of times a node as relays for others on four datasets

Figure 1.12 shows the number of times a node falls on the shortest paths between all other node pairs. We can treat this simply as the centrality of a node in the

system. We observe 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 that have moderate 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 popular hub for the whole city, i.e., a postman or a newspaper man in a popular underground station, who relayed a certain amount of cross city traffic. The 30th, 70th percentiles and the means of normalised individual node centrality are shown in Table 1.4. These numbers summarise the statistical property of the centrality values for each system shown in Figure 1.12.

Experimental dataset	30th percentile	Mean	70th percentile
Cambridge	0.052	0.220	0.194
Reality	0.005	0.070	0.050
Infocom06	0.121	0.188	0.221
Hong Kong	0.000	0.017	0.000

Table 1.4 Statistics about normalised node centrality in 4 experiments

1.3.4 Social-based Routing

The contribution of this section is to combine the knowledge of both centralities of nodes and community structure, to achieve further performance improvements in forwarding. We show that this avoids 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.

1.3.4.1 Overview of Forwarding Algorithms

In order to compare and evaluate the efficiency of the forwarding algorithms in finding the good paths for the destination. Forwarding algorithms can be divided into two broad categories: those that are aware of the social structure, and those oblivious to social structure. BUBBLE exploits the latent social structure. We evaluate its performance in relation to the following naïve strategies which attempt to forward data without using any social structure.

- WAIT: Hold onto a message until the sender encounters the recipient directly, which represents the lower bound for delivery cost. WAIT is the only single-copy algorithm in this chapter.
- FLOOD: Messages are flooded throughout the entire system, which represents the upper bound for delivery and cost.
- Multiple-Copy-multiple-hop (MCP): Multiple copies are sent subject to a time-to-live hop count limit on the propagation of messages. By exhaustive emulations, the 4-copy-4-hop MCP scheme was found to be the most cost-effective scheme in terms of delivery ratio and cost for all naive schemes among most of the datasets.

In contrast to the above, we explore four different algorithms which leverage various different aspects of social structure:

- LABEL: Explicit labels are used to identify forwarding nodes that belong to the same organisation. Optimisations 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: The forwarding metric used in this algorithm is the node centrality. A message is forwarded to nodes with higher centrality values than the current node. It is based on observations in the network plane, although it also reflects the hub popularity in the human dimension.
- DEGREE: The forwarding metric used in this algorithm is the node degree, more specifically the observed average of the degree of a node over a certain time interval. Either the last interval window (S-Window), or a long-term cumulative estimate, (C-Window) is used to provide a fully decentralised 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 and observed community structure with explicit labels, to decide on the best forwarding nodes. BUBBLE is an example algorithm which uses information from both human aspects and also the physically observable aspects of mobility.

BUBBLE is a combination of LABEL and RANK. It uses RANK to spread out the messages and uses LABEL to identify the destination community. For this algorithm, we make two assumptions:

- Each node belongs to at least one community. Here we allow single node communities to exist.
- Each node has a global ranking (i.e., global centrality) in the whole system and also a local ranking within its community. It may belong to multiple communities and hence may have multiple local rankings.

Figure 1.13 shows the design space for the social-based forwarding algorithms. The vertical axis represents the explicit social structure. This is the social or human dimension. The two horizontal axes represent the network structural plane, which

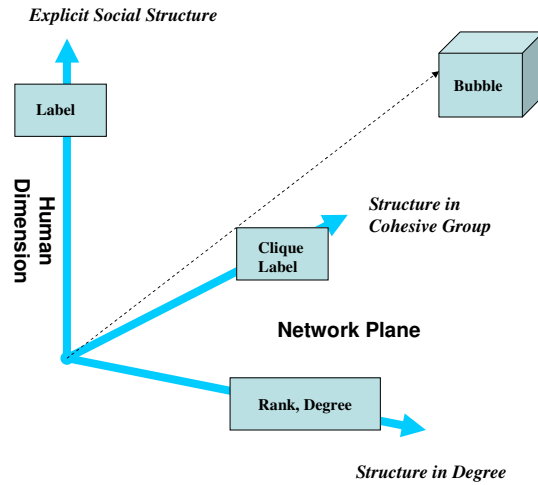


Fig. 1.13 Design space for forwarding algorithms

can be inferred purely from observed contact patterns. The Structure-in-Cohesive Group axis indicates the use of localised cohesive structure, and the Structure-in-Degree axis indicates the use of node ranking and degree. These are observable physical characteristics. In our design framework, it is not necessary that physical dimensions are orthogonal to the social dimension, but since they represent two different design parameters, we would like to separate them. The design philosophy here is to consider both the social and physical aspects of mobility.

1.3.4.2 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 as confirmed using our community detection algorithm, we can clearly divide the *Cambridge* data into two communities: the undergraduate year-one and year-two 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 of 9; 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 clearly see from Figures 14(a) and 14(b) that inside a community, the centrality of each node is different. In Group B, there are two nodes which are very popular, and have relayed most of the traffic. All the other nodes have low centrality value.

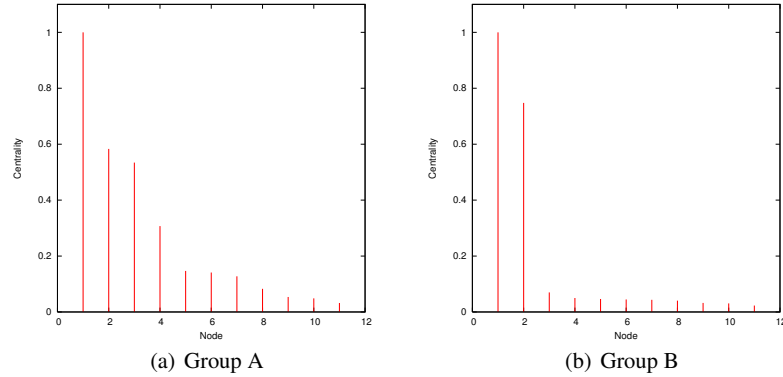


Fig. 1.14 Node centrality in 2 groups in *Cambridge* data, see Section 1.3.3 for the method of calculating the centrality values.

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 1.15 shows the individual node centrality when traffic is created from one group to another and the correlation of node centrality within an individual group and inter-group (for data deliveries only to other groups but not to its only 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 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 will be useful when we need to move traffic from one group to another.

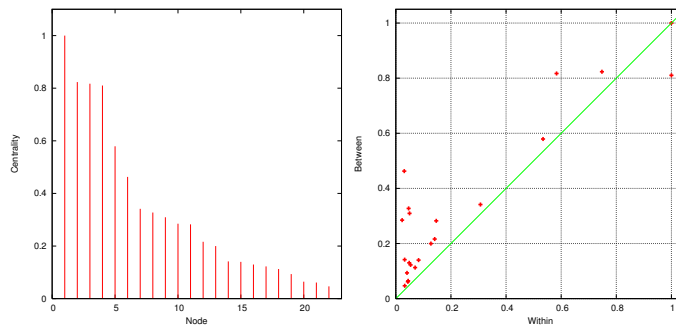


Fig. 1.15 Inter-group centrality (left) and correlation between intra- and inter-group centrality (right), *Cambridge*

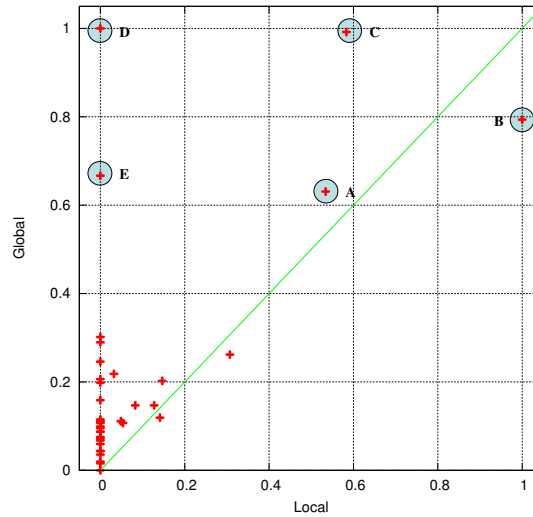


Fig. 1.16 Correlation of local centrality of group A and the global centrality (*Cambridge*)

Figure 1.16 shows the correlation of the local centrality of Group A and the global centrality of the 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 may 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, and to node B it would be perfect. But if it push the traffic to node D and E, the traffic could get stuck there and not be routed 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. Here we propose the BUBBLE algorithm to avoid these dead-ends.

Forwarding is carried out as follows. 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 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, since each

world/community is like a bubble. Figure 1.17 illustrates the BUBBLE algorithm and the pseudo code can be found in our previous work [19].

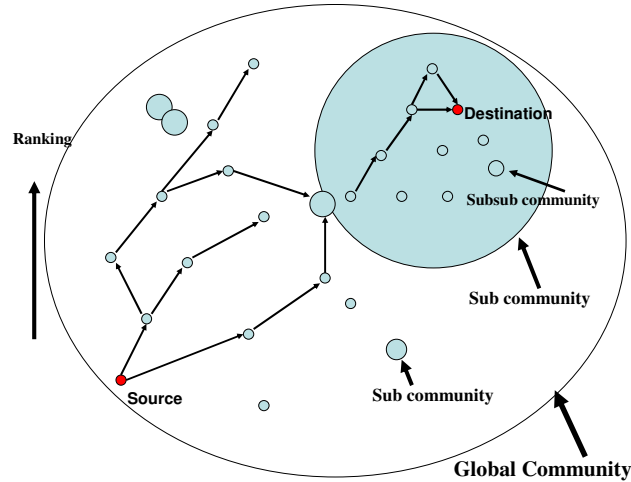


Fig. 1.17 Illustration of the BUBBLE forwarding algorithm.

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 reaching 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, and the original algorithm introduced above BUBBLE-A.

We can see from Figure 1.18 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 message deletion mechanism, it achieves exactly the same delivery as BUBBLE-A. 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.

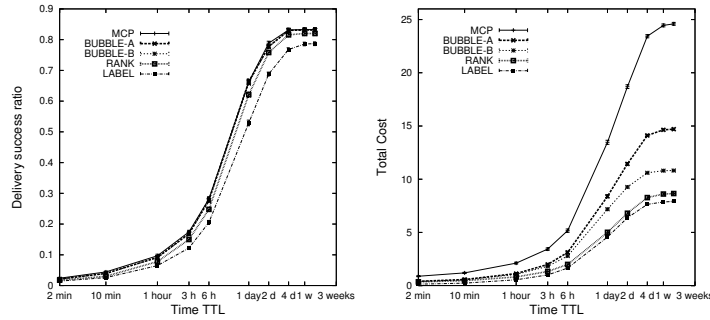


Fig. 1.18 Comparisons of several algorithms on *Cambridge* dataset

1.3.4.3 Multiple-community Cases

To study the multiple-community cases, we use the *Reality* dataset. To evaluate the forwarding algorithm, we extract a 3-week session during term time from the whole 9-month dataset. Emulations are run over this dataset with uniformly generated traffic.

There is a total of 8 groups within the whole dataset. Figure 1.19 shows the node centrality in 4 groups, from small-size to medium-size and large-size group. We can see that within each group, almost every node has different centrality.

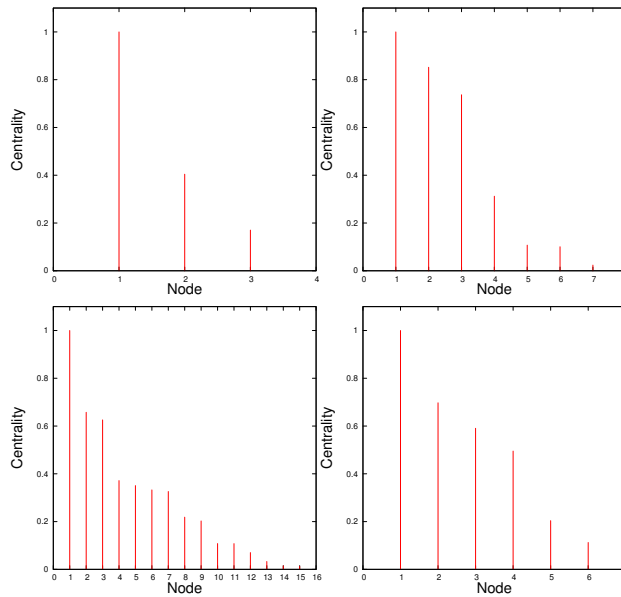


Fig. 1.19 Node centrality in several individual groups (*Reality*)

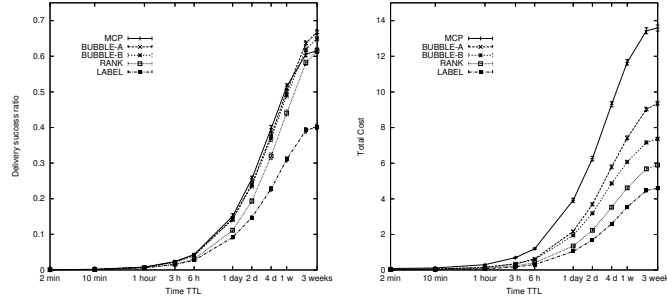


Fig. 1.20 Comparisons of several algorithms on *Reality* dataset, single group

In order to make our study easier, we first isolate the largest group in Figure 1.19, 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 1.20 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. 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 (i.e., only global 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 the BUBBLE algorithms.

From Figure 1.21, 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 of BUBBLE. BUBBLE is very close in performance to MCP in multiple groups case as well, and even outperforms it when the time TTL of the messages is allowed to be larger than 2 weeks.⁵ However, the cost is only 50% that of MCP. Figure 1.22 shows the same performance evaluations with the *Infocom06* dataset. In this case, the delivery ratio of RANK is approaching that of MCP but with less than half of the cost. The performance of BUBBLE over RANK is not as significant as in the *Reality* case because in a conference scenario the people are very mixing and hence the community factors are less dominating. We can also see that even in this case, the delivery cost for BUBBLE only increases slightly, which indicates that even in a

⁵ Two weeks seems to be very long, but as we have mentioned before, the *Reality* network is very sparse. We choose it mainly because it has long experimental period and hence more reliable community structures can be inferred. The evaluations here can serve as a proof of concept of the BUBBLE algorithm, although the delays are large in this dataset.

mixing environment, BUBBLE is still very robust towards the possible misleading of the community factors.

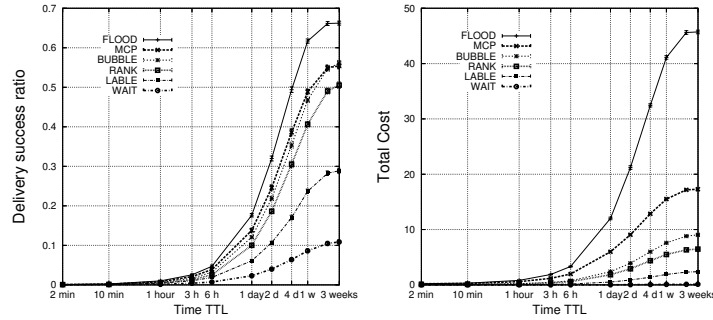


Fig. 1.21 Comparisons of several algorithms on *Reality* dataset, all groups

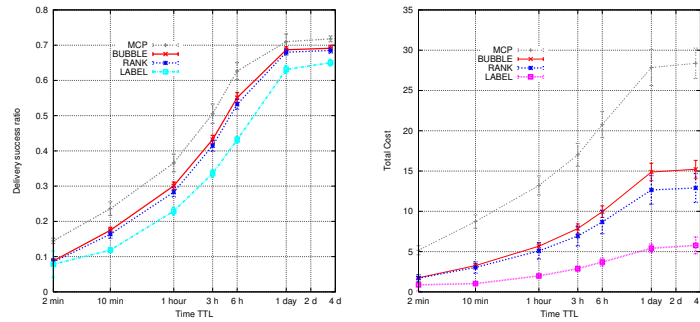


Fig. 1.22 Comparisons of several algorithms on *Infocom06* dataset, all groups

In BUBBLE and RANK algorithm, nodes with high centrality are more likely to act as relay nodes than the others. Excessive traffic through a node might cause the node to run out of battery or possibly lead to package losses. An easy fix is to impose admission control at each node. Each node maintains a limited buffer for storing data for other nodes and if the buffer has reached its limit, it will not admit incoming data. This may lower the delivery efficiency but can get rid of the excessive traffic problem. We will further study the trade-off and optimal buffer size in future work.

1.4 Related Work

Conceptually, PSNs are Delay-Tolerant Networks [12], and generic results from that framework apply. For instance, a forwarding algorithm that has more knowledge about contacts is likely to be more successful [22], and the best performance is achieved by an oracle with knowledge of future contacts.

Nevertheless, the fact that our underlying network is made up of human contacts and is less predictable has a large impact: For instance, reasonably predictable traffic patterns of buses allow a distributed computation of route metrics for packets in vehicular DTNs [22, 2]. Similarly, fixed bus routes allow the use of throwboxes [47] to reliably transfer data between nodes that visit the same location, but at different times.

The variability of PSNs has naturally led to a statistical approach: The inter-contact time distribution of human social contacts has been used to model transmission delay between a randomly chosen source-destination pair [4, 24]. In this work, we take a more macroscopic view and look at the ability of the PSN to simultaneously deliver data between multiple source-destination pairs. This leads us to look at the distribution of the *number* of contacts between randomly chosen source-destination pairs, and find that this distribution is not only crucial for global data delivery performance, but also for the connectivity of the PSN itself.

[39, 14, 45] model the performance of epidemic routing and its variants. In particular, they derive a closed form for delivery time distribution, and show it to be accurate for certain common mobility models. However, several simplifying assumptions are made, including an exponential inter-contact time between node pairs. Unfortunately, human contact networks are known to have power law inter-contact times with exponential tails [4, 24]. Furthermore, [39, 14, 45] use a constant contact rate, whereas our studies show that human contacts are highly heterogeneous. [29] considers heterogeneous contact rates between mobile devices but only in the context of establishing an epidemic threshold for virus spread.

The number of paths found by flooding is crucial to the success of proportional and k -copy flooding. Counting differently, [10] reports a phenomenon of “path explosion” wherein thousands of paths reach a destination shortly after the first, many of which are duplicates, shifted in time. In contrast, duplicate paths are prevented in our method of counting, by having nodes remember if they have already received some data, resulting in a maximum of $N - 2$ paths between a source and destination.

The power of using multiple paths has been recognised. Binary Spraying, which forms the basis for two schemes (spray and wait, spray and focus) has been shown to be optimal in the simple case when node movement is independent and identically distributed [40]. [11] noted that among routing schemes evaluated, those using more than one copy performed better. Furthermore, all algorithms employing multiple paths showed similar average delivery times. The success of k -copy flooding suggests a possible explanation for this result. Similarly, [21] finds that the delivery ratio achieved by a given time is largely independent of the propensity of nodes to carry other people’s data. They suggest the existence of multiple paths as an explanation. At an abstract level, the refusal of a node to carry another node’s data can

be treated as a path failure. Thus Sec. 1.2.3 corroborates [21] and provides a direct explanation.

For distributed search for nodes and content in power-law networks, Sarshar *et al.* [37] proposed using a probabilistic broadcast approach: sending out a query message to an edge with probability just above the bond⁶ 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 DTNs and mobile ad hoc networks, there is much existing literature. Vahdat *et al.* proposed epidemic routing, which is similar to the “oblivious” flooding scheme we evaluated in this chapter [43]. Spray and Wait is another “oblivious” flooding scheme but with a self-limited number of copies [40]. Grossglauser *et al.* proposed the two-hop relay schemes to improve the capacity of dense ad hoc networks [15]. Many approaches calculate the probability of delivery to the destination node, where the metrics are derived from the history of node contacts, spatial information and so forth. The pattern-based Mobyspace Routing by Leguay *et al.* [27], location-based routing by Lebrun *et al.* [26], context-based forwarding by Musolesi *et al.* [30] and PROPHET Routing [1] fall into this category. PROPHET uses past encounters to predict the probability of future encounters. The transitive nature of encounters is exploited, where indirectly encountering the destination node is evaluated. Message Ferry by Zhao *et al.* [46] takes a different approach by controlling the movement of each node.

Recent attempts to uncover a hidden stable network structure in DTNs such as social networks have been emerged. For example, SimBet Routing [5] uses ego-centric centrality and its social similarity. Messages are forwarded towards the node with higher centrality to increase the possibility of finding the potential carrier to the final destination. LABEL forwarding [17] uses affiliation information to help forwarding in PSNs based on the simple intuition that people belonging to the same community are likely to meet frequently, and thus act as suitable forwarders for messages destined for members of the same community. We have compared BUBBLE with LABEL and demonstrate that by the exploitation of both community and centrality information, BUBBLE provide further improvement in forwarding efficiency. The mobility-assisted Island Hopping forwarding [36] uses network partitions that arise due to the distribution of nodes in space. Their clustering approach is based on the significant locations for the nodes and not for clustering nodes themselves. Clustering nodes is a complex task to understand the network structure for aid of forwarding.

Interested readers can obtain further details about the research presented in this chapter from [20] and [38].

⁶ A percolation which considers the lattice edges as the relevant entities.

1.5 Conclusion

This chapter discussed the idea of Pocket Switched Networks, which proposes to use human contacts to opportunistically transfer data over time from sender to destination. We first examined the feasibility of using local contacts for achieving global $N \times N$ connectivity in the temporal network formed by human contacts and showed that although the frequently occurring edges are not very effective for data transfer, the network exhibits a remarkable resilience in the face of path failures. We also showed that it is possible to uncover important characteristic properties of social network from a diverse set of real world human contact traces and demonstrated that community and centrality social metrics can be effectively used in forwarding decisions. Our BUBBLE algorithm is designed for a delay tolerant network environment, built out of human-carried devices, and we have shown that it has similar delivery ratio to, but much lower resource utilisation than flooding and control flooding. We believe that this approach represents an early step in combining rich multi-level information of social structures and interactions to drive novel and effective means for disseminating data. A great deal of future research can follow.

References

1. A.Lindgren, A.Doria, O.Schelen: Probabilistic routing in intermittently connected networks. In: Proc. SAPIR (2004)
2. Balasubramanian, A., Levine, B.N., Venkataramani, A.: DTN Routing as a Resource Allocation Problem. In: SIGCOMM (2007)
3. Chaintreau, A., Hui, P., Crowcroft, J., Diot, C., Gass, R., Scott, J.: Impact of human mobility on the design of opportunistic forwarding algorithms. In: Proc. INFOCOM (2006)
4. Chaintreau, A., et al.: Impact of human mobility on opportunistic forwarding algorithms. IEEE Transactions on Mobile Computing **6**(6), 606–620 (2007)
5. Daly, E., Haahr, M.: Social network analysis for routing in disconnected delay-tolerant manets. In: Proceedings of ACM MobiHoc (2007)
6. Danon, L., Duch, J., Diaz-Guilera, A., Arenas, A.: Comparing community structure identification. J. Stat. Mech. p. P09008 (2005)
7. Dunbar, R.I.M.: Co-evolution of neocortex size, group size and language in humans. Behavioral and Brain Sciences **16** (1993)
8. Eagle, N., Pentland, A.: Reality mining: sensing complex social systems. Personal and Ubiquitous Computing **V10**(4), 255–268 (2006)
9. Eagle, N., Pentland, A.S.: CRAWDAD data set mit/reality (v. 2005-07-01). <http://crawdad.cs.dartmouth.edu/mit/reality>
10. Erramilli, V., Chaintreau, A., Crovella, M., Diot, C.: Diversity of forwarding paths in pocket switched networks. In: Proceedings of ACM Internet Measurement Conference (2007)
11. Erramilli, V., Crovella, M., Chaintreau, A., Diot, C.: Delegation forwarding. In: MobiHoc (2008)
12. Fall, K.: A delay-tolerant network architecture for challenged internets. In: "SIGCOMM" (2003)
13. Freeman, L.C.: A set of measuring centrality based on betweenness. Sociometry **40**, 35–41 (1977)
14. Groenevelt, R., Nain, P., Koole, G.: The message delay in mobile ad hoc networks. Perform. Eval. **62**(1-4), 210–228 (2005)

15. Grossglauser, M., Tse, D.N.C.: Mobility increases the capacity of ad hoc wireless networks. *IEEE/ACM Trans. Netw.* **10**(4), 477–486 (2002)
16. Hui, P., Chaintreau, A., Scott, J., Gass, R., Crowcroft, J., Diot, C.: Pocket switched networks and the consequences of human mobility in conference environments. In: *Proceedings of ACM SIGCOMM first workshop on delay tolerant networking and related topics* (2005)
17. Hui, P., Crowcroft, J.: How small labels create big improvements. In: *Proc. IEEE ICMAN* (2007)
18. Hui, P., Crowcroft, J.: Human mobility models and opportunistic communications system design. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences* **366**(1872), 2005–2016 (2008)
19. Hui, P., Crowcroft, J., Yoneki, E.: Bubble rap: Social-based forwarding in delay tolerant networks. In: *MobiHoc '08: Proceedings of the 9th ACM international symposium on Mobile ad hoc networking & computing* (2008)
20. Hui, P., Crowcroft, J., Yoneki, E.: Bubble rap: Social-based forwarding in delay tolerant networks. *IEEE Transactions on Mobile Computing* **99**(PrePrints) (2010). DOI <http://doi.ieeecomputersociety.org/10.1109/TMC.2010.246>
21. Hui, P., Xu, K., Li, V., Crowcroft, J., Latora, V., Lio, P.: Selfishness, altruism and message spreading in mobile social networks. In: *Proc. of First IEEE International Workshop on Network Science For Communication Networks (NetSciCom09)* (2009)
22. Jain, S., Fall, K., Patra, R.: Routing in a delay tolerant network. In: *SIGCOMM* (2004)
23. Jones, E.P.C., Li, L., Ward, P.A.S.: Practical routing in delay-tolerant networks. In: *Proc. WDTN* (2005)
24. Karagiannis, T., Le Boudec, J.Y., Vojnovic, M.: Power law and exponential decay of inter contact times between mobile devices. In: *MOBICOM* (2007)
25. Karagiannis, T., Le Boudec, J.Y., Vojnović, M.: Power law and exponential decay of inter contact times between mobile devices. In: *ACM MobiCom '07* (2007)
26. Lebrun, J., Chuah, C.N., Ghosal, D., Zhang, M.: Knowledge-based opportunistic forwarding in vehicular wireless ad hoc networks. *IEEE VTC* **4**, 2289–2293 (2005)
27. Leguay, J., Friedman, T., Conan, V.: Evaluating mobility pattern space routing for DTNs. In: *Proc. INFOCOM* (2006)
28. Leguay, J., Lindgren, A., Scott, J., Friedman, T., Crowcroft, J.: Opportunistic content distribution in an urban setting. In: *ACM CHANTS*, pp. 205–212 (2006)
29. Mickens, J.W., Noble, B.D.: Modeling epidemic spreading in mobile environments. In: *WiSe '05: Proceedings of the 4th ACM workshop on Wireless security* (2005)
30. Musolesi, M., Hailes, S., et al.: Adaptive routing for intermittently connected mobile ad hoc networks. In: *Proc. WOWMOM* (2005)
31. Newman, M.E.J.: Analysis of weighted networks. *Physical Review E* **70**, 056,131 (2004)
32. Newman, M.E.J.: Detecting community structure in networks. *Eur. Phys. J. B* **38**, 321–330 (2004)
33. Newman, M.E.J., Girvan, M.: Finding and evaluating community structure in networks. *Physical Review E* **69** (2004). DOI 10.1103/PhysRevE.69.026113. URL <http://arxiv.org/abs/cond-mat/0308217>
34. Okasha, S.: Altruism, group selection and correlated interaction. *British Journal for the Philosophy of Science* **56**(4), 703–725 (2005)
35. Palla, G., Derényi, I., Farkas, I., Vicsek, T.: Uncovering the overlapping community structure of complex networks in nature and society. *Nature* **435**(7043), 814–818 (2005). DOI 10.1038/nature03607. URL <http://dx.doi.org/10.1038/nature03607>
36. Sarafijanovic-Djukic, N., Piorkowski, M., Grossglauser, M.: Island hopping: Efficient mobility-assisted forwarding in partitioned networks. In: *IEEE SECON* (2006)
37. Sarshar, N., Boykin, P.O., Roychowdhury, V.P.: Scalable percolation search in power law networks (2004). URL <http://arxiv.org/abs/cond-mat/0406152>
38. Sastry, N., Manjunath, D., Sollins, K., Crowcroft, J.: Data delivery properties of human contact networks. *IEEE Transactions on Mobile Computing* **99**(PrePrints) (2010). DOI <http://doi.ieeecomputersociety.org/10.1109/TMC.2010.225>

39. Small, T., Haas, Z.J.: The shared wireless infostation model: a new ad hoc networking paradigm (or where there is a whale, there is a way). In: *MobiHoc* (2003)
40. Spyropoulos, T., Psounis, K., Raghavendra, C.S.: Efficient routing in intermittently connected mobile networks: the multiple-copy case. *IEEE/ACM Trans. Netw.* **16**(1) (2008)
41. Travers, J., Milgram, S.: An experimental study of the small world problem. *Sociometry* **32**(4), 425–443 (1969)
42. "UCSD": Wireless topology discovery project. <http://sysnet.ucsd.edu/wtd/wtd.html> (2004)
43. Vahdat, A., Becker, D.: Epidemic routing for partially connected ad hoc networks. Tech. Rep. CS-200006, Duke University (2000)
44. Yoneki, E., Hui, P., Crowcroft, J.: Visualizing community detection in opportunistic networks. In: *CHANTS'07: Proc. of the second ACM workshop on Challenged Networks*, pp. 93–96 (2007). DOI <http://doi.acm.org/10.1145/1287791.1287810>
45. Zhang, X., Neglia, G., Kurose, J., Towsley, D.: Performance modeling of epidemic routing. *Comput. Netw.* **51**(10), 2867–2891 (2007)
46. Zhao, W., Ammar, M., Zegura, E.: A message ferrying approach for data delivery in sparse mobile ad hoc networks. In: *Proceedings of the MobiCom 2004* (2004)
47. Zhao, W., et al.: Capacity Enhancement using Throwboxes in DTNs. In: *Proc. IEEE Intl Conf on Mobile Ad hoc and Sensor Systems (MASS)* (2006)