Identifying Social Communities in Complex Communications for Network Efficiency

Pan Hui[1,*], Eiko Yoneki[2], Jon Crowcroft[2], and Shu-Yan Chan[2]

[1] Deutsche Telekom Laboratories / TU Berlin,
Ernst-Reuter-Platz 7, 10587 Berlin, Germany
`Pan.Hui@telekom.de`
[2] Computer Laboratory, University of Cambridge,
15 JJ Thomson Avenue, CB3 0FD Cambridge, UK
`lastname.firstname@cl.cam.ac.uk`

Abstract. Complex communication networks, more particular Mobile Ad Hoc Networks (MANET) and Pocket Switched Networks (PSN), rely on short range radio and device mobility to transfer data across the network. These kind of mobile networks contain duality in nature: they are radio networks at the same time also human networks, and hence knowledge from social networks can be also applicable here. In this paper, we demonstrate how identifying social communities can significantly improve the forwarding efficiencies in term of delivery ratio and delivery cost. We verify our hypothesis using data from five human mobility experiments and test on two application scenarios, asynchronous messaging and publish/subscribe service.

Keywords: Pocket Switched Networks, Human Mobility, Community, Social Network, Asynchronous Messaging, Publis/Subscribe.

1 Introduction

We envision a future in which a multitude of devices carried by people are dynamically networked, forming Pocket Switched Networks (PSN) [1]: a type of Delay Tolerant Network (DTN) [2] for such environments. A PSN uses contact opportunities to allow humans to communicate without network infrastructure.

An efficient data forwarding mechanism over the temporal graph of the PSN [3] is required that copes with dynamic network topology by human mobility, and repeated disconnection and re-wiring. We believe the traditional approach of building and updating routing tables is not cost effective for a PSN, since mobility patterns are often unpredictable and topology changes can be rapid. Rather than exchanging much control traffic to create unreliable routing structures, we search for characteristics of the network that are less volatile than mobility. A PSN is formed by people, and the social relationships among people may prove to be a more stable network structure. Unicast and multicast

---

"routes" in this system are emergent properties of the community structure and community interests, respectively. Thus, such a social backbone can be used for better forwarding decisions.

Community is an important attribute of PSNs. Cooperation binds, but also divides human society into communities. Human society is structured. 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 [4]. This correlated interaction concept also applies to human, so we can exploit this kind of community information to select forwarding paths. We believe identifying social communities can help to choose next relays for particular destinations, and hence reduce the number of unwanted traffic generated (delivery cost).

In this paper, we use five experimental datasets, which cover a rich diversity of environments from busy metropolitan city to quite university town, with an experimental period from several days to almost one year, to verify our hypothesis that identifying social communities can help to improve forwarding efficiency. We evaluate our results on both single-point communication and multi-point communication to make a more general conclusion. For these two kinds of communication, we use more particularly the *asynchronous messaging* and *publish/subscribe* applications.

2   Experimental Datasets

In this paper, we use four experimental datasets gathered by the Haggle Project [1] over two years, referred to as *Infocom05*, *HongKong*, *Cambridge*, and *Infocom06*; one dataset from the MIT Reality Mining Project [5], referred to as *Reality*. Previously, the characteristics of these datasets such as inter-contact and contact distribution have been explored in several studies [6] [1] [7], to which we refer the reader for further background information. We believe these five datasets cover a rich diversity of environments from busy metropolitan city (*HongKong*) to quite university town (*Cambridge*), with an experimental period from several days (*Infocom06*) to almost one year (*Reality*). Datasets from cellular operators, for example the one used by Gonzalez *et al.* [8], can be much larger in scale but lack of peer-to-peer proximity logging of neighbor devices, hence can not be used for evaluation of PSN applications.

 – 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 *Hong-Kong*, the people carrying the wireless devices were chosen independently in a Hong-Kong bar, to avoid any particular social relationship

Table 1. Characteristics of the five experimental data sets

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

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.
- 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.

3   Communities in the Mobility Traces

A social network consists of a set of people forming socially meaningful relationships, where prominent patterns or information flow are observed. In PSN, social networks could map to computer networks since people carry the computer devices. In this section, we implement and apply Newman's weighted network analysis (WNA) for our data analysis [9].

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

$$Q = \sum_{vw} \left[\frac{A_{vw}}{2m} - \frac{k_v k_w}{(2m)^2} \right] \delta(c_v, c_w) \tag{1}$$

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 of which vertex i belongs to. Therefore the term in the formula $\frac{\sum_{vw} A_{vw}}{2m} \delta(c_v, c_w)$ is equal to $\frac{\sum_{vw} A_{vw} \delta(c_v, c_w)}{\sum_{vw} A_{vw}}$, which is the fraction of the edges that fall within communities. *Modularity* is defined as the difference between this fraction and, the fraction of the edges that would be expected to

Table 2. Communities detected from the four datasets

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

fall within the communities if the edges were assigned randomly but keeping the degrees of the vertices unchanged. The algorithm is essentially a genetic algorithm, using the modularity as the measurement of fitness. Instead of testing on some mutations of the current best solutions, it enumerates all possible merges of any two communities in the current solution, evaluates the relative fitness of the resulting merges, and chooses the best solution as the seed for the next iteration.

Table 2 summarises the communities detected by applying WNA on the four datasets. According to Newman [9], nonzero 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 spound by WNA is exactly matched the two groups (1st year and 2nd year) of students selected for the experiment.

4   Single-Point Communication

Here we propose the BUBBLE algorithm, with the intention of bringing in a concise concept of community into PSN forwarding to achieve significant improvement of forwarding efficiency. BUBBLE combines the knowledge of community structure with the knowledge of node centrality to make forwarding decisions. There are two intuitions behind this algorithm. Firstly, people have varying roles and popularities in society, and these should be true also in the network – the first part of the forwarding strategy is to forward messages to nodes which are more popular than the current node. Secondly, people form communities in their social lives, and this should also be observed in the network layer – hence the second part of the forwarding strategy is to identify the members of destination communities, and to use them as relays. Together, we call this BUBBLE forwarding. 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 [10]) across the whole system, and also a local ranking within its local community. It may also belong to multiple communities and hence may have multiple local rankings.
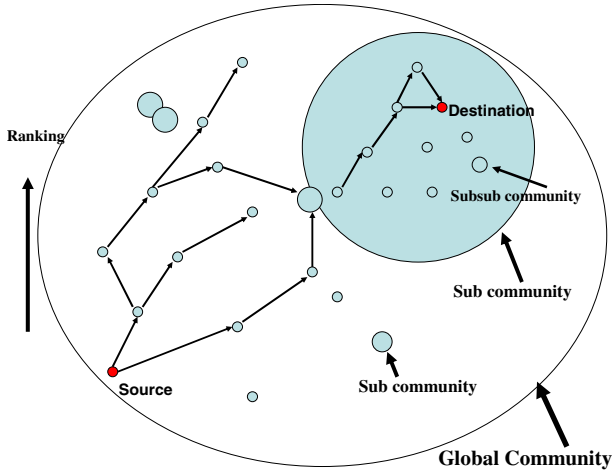
Fig. 1. Illustration of the BUBBLE algorithm

Forwarding is carried out as follows. If a node has a message destined for another node, this node first *bubbles* the message up the hierarchical ranking tree using the global ranking, until it reaches a node which is in the same community as the destination node. Then the local ranking system is used instead of the global ranking, and the message continues to bubble up through the local ranking tree until the destination is reached or the message expires. 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. In order to reduce cost, we also require that whenever a message is delivered to the community, the original carrier can delete this message from its buffer to prevent further dissemination. This assumes that the community member can deliver this message. We call this algorithm BUBBLE, using the metaphor of *bubble* for a community.

The forwarding process fits our intuition and is taken from real life experiences. First you try to forward the message via surrounding people more popular than you, and then you bubble it up to well-known popular people in the wider-community, such as a postman. When the postman meets a member of the destination community, the message will be passed to that community. The first community member who receives the message will try to identify more popular members within the community, and bubble the message up again within the local hierarchy, until the message reaches a very popular member, or the destination itself, or the message expires. Figure 1 illustrates the BUBBLE algorithm.

5   Multi-point Communication

Creating an overlay for message dissemination has been a popular technique for multi-point communication. Below, we describe a brief discussion of existing

approaches in MANETs along gossip based approaches. This discussion leads to our proposal: *Socio-Aware Overlay.*

State maintenance requires control traffic, which could be expensive to operate, while a stateless approach could also be expensive if using event flooding. Stateful approaches suffer under frequent topology changes, and stateless approaches are more suitable for topology change and the partitioning and isolation of nodes. Thus, dealing with mobility and partitioning of networks shows that the basis of event dissemination mechanisms should be epidemic. The basic gossip dissemination sends each message to a randomly chosen group of nodes. This approach operates in a decentralised fashion and is robust against node and network link failures. Cluster-based protocols partition a wireless network into several disjoint and equally sized regions, and select a cluster head in each region to operate message exchange. Protocols with clustering techniques include *GeoGRID* [11] and *Obstacle-Free Single-Destination Geocasting Protocol* (OFSGP). *Socio-Aware Overlay* takes a clustering-based approach and a membership of the group is dynamically detected through community detection process rather than implicitly defined as the set of nodes within a certain area in geographical or physical casting.

Structured overlays assign identifiers to nodes and control the identifiers of neighbours in overlay networks and the keys of the objects they store. This is effective since lookups can be done with cost $O(logN)$; this is better than a flooding approach. However, the characteristics of MANETs require a significant amount of traffic to maintain the overlay links. Thus, strict layering may not work.

In [12], a structured P2P overlay network is used for a publish-subscribe system. Subscriptions are mapped to keys and sent to a rendezvous node. There is some optimisation such as bundled notification dissemination. The performance of this approach depends on the real mapping between the overlay network and the underlying network topology.

We propose multi-point event dissemination using an overlay constructed by closeness centrality nodes in communities. Detected communities are well connected implying that socially they share the same interests with high chances. Thus, similar subscriptions may coexist within the same community. The fundamental idea of this approach is instead of artificially constructing an overlay based on various contexts (e.g. location, group mobility), the existing structure is detected and mapped to the function. Thus, this approach strongly depends on the dynamic community detection mechanisms. A crucial factor is how good the community detection mechanism is. The current simple detection algorithm detects approximately 60% of communities compared to the centralised approach. We are adding messaging passing at a certain temporal point to improve the community detection.

We currently choose a closeness centrality node for the broker node as the closeness centrality imply the best visibility in the community. Thus, once this node gets the message, delivery to any member of the community has high reliability. Because of the characteristics of human networks (i.e. scale-free networks), many nodes within a community are tightly connected and multiple closeness
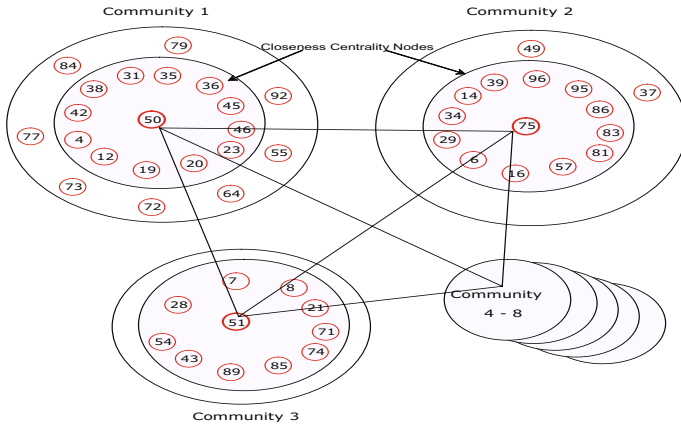
Fig. 2. Community Structure

centrality nodes can coexist. This is an advantage to balance the workload of brokers and will be the subject of future work to add a load balancing mechanism. We are also investigating other criteria to select the broker node, which is work in progress. Subscription propagation occurs as part of gossiping, thus, it does not cause any extra cost. The proposed multi-point communication takes advantage of PSNs, where various communication methods can be used to control delay in DTNs.

Communication between brokers can have two modes: *Unicast* and *Direct*. *Unicast* is based on the underlying unicast algorithms. Thus, it could end up epidemic routing. *Direct* provides more direct communication mechanism such as WiFi access points or GPRS *Direct* approach gives accelerated message delivery with some cost. When *Unicast* is used for the communication between broker nodes, the average hop count follows the distance of the pair nodes (i.e. 1.6 hops for MIT Reality mining trace). Using the betweenness centrality, where a node has dual visibility from and to communities will improve the hop counts, and we are investigating this extension.

6   Results and Evaluations

In this section, we show the results for both single-point and multi-point communication, using asynchronous messaging and publish/subscribe service as the specific applications.

6.1   Single-Point Communication

In order to evaluate different forwarding algorithms, we use a discrete event simulator called *HaggleSim*. The original trace files are divided into discrete sequential contact events, and they are fed into the emulator as inputs. For every discrete encounter event, the emulator makes a forwarding decision based on the forwarding algorithm under study.

For each emulation in this paper, 1000 messages are created, uniformly sourced between all node pairs. Each emulation is repeated 20 times with different random seeds for statistical confidence. For all the emulations we have conducted for this work, we have measured the following metrics and for all the metrics, we compute the 95th percentile using t-distribution.

Delivery ratio: The proportion of messages that have been delivered out of the total unique messages created.
Delivery cost: The total number of messages (include duplicates) transmitted across the air. To normalize this, we divide it by the total number of unique messages created.

We compare our algorithms against the following five benchmark algorithms.

WAIT: Hold on to a message until the sender encounters the recipient directly, which represents the lower bound for delivery and cost.
FLOOD: Messages are flooded throughout the entire system, which represents the upper bound for delivery and cost.
MCP: Multiple-Copy-Multiple-Hop. Multiple Copies are sent subject to a time-to-live hop count limit on the propagation of messages. This is a controlled flooding strategy.
LABEL: A social based forwarding algorithm introduced by Hui *et al.* [13]. Messages are only forwarded to the nodes in the same community (i.e.with the same label) as the destination.
PROPHET: A standard non-oblivious benchmark that has been evaluated against several previous works [14]. It calculates the delivery predictability at each node for each destination by using history of encounters and transitivity. A message is forwarded to a node if it has higher delivery predictability than the current node for that particular destination.

In this paper, we only show the *Reality* dataset as an example due to the limit of space. To evaluate the forwarding algorithm, we extract a 3 week session during term time from the whole 9 month dataset. Emulations were run over this dataset with uniformly generated traffic. There is a total 8 groups within the whole dataset. We observed that within each individual group, the node centralities demonstrate diversity similar to the *Cambridge* case.

From Figure 3(a) and Figure 3(b), we can see that of course flooding achieves the best 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 the multiple-group case as well, and even outperforms 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.

Regarding LABEL forwarding, we can observe from Figure 3 that LABEL 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 [13]. A person in one group may not meet members in another group
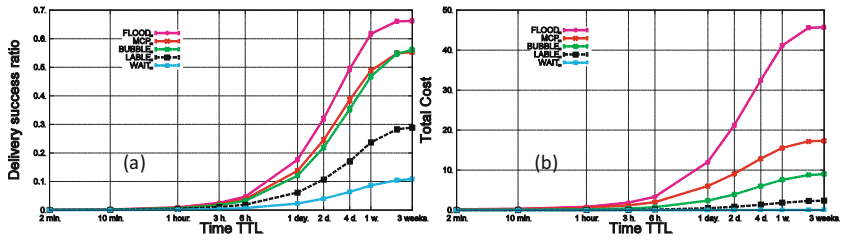
Fig. 3. Comparisons of several algorithms on *Reality* dataset, all groups

so often, waiting to meet a member of the destination group before transmitting is not effective.

In order to further justify the significance of social based forwarding, we also compare BUBBLE with a benchmark 'non-oblivious' forwarding algorithm, PROPHET[14]. PROPHET uses the history of encounters and transitivity to calculate the probability that a node can deliver a message to a particular destination. Since it has been evaluated against other algorithms before and has the same contact-based nature as BUBBLE (i.e. do not need location information), it is a good target to compare with BUBBLE.

PROPHET has four parameters. We use the default PROPHET parameters as recommended in [14]. However, one parameter that should be noted is the time elapsed unit used to age the contact probabilities. The appropriate time unit used differs depending on the application and the expected delays in the network. Here, we age the contact probabilities at every new contact. In a real application, this would be a more practical approach since we do not want to continuously run a thread to monitor each node entry in the table and age them separately at different time.

Figure 4 (a) and (b) shows the comparison of the delivery ratio and delivery cost of BUBBLE and PROPHET. Here, for the delivery cost, we only count the
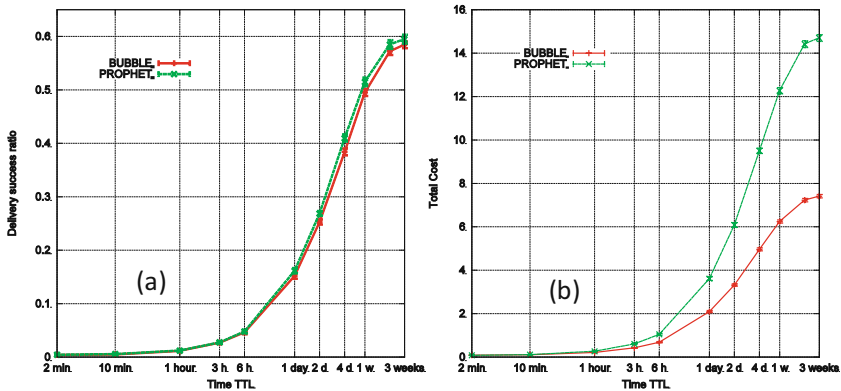

Fig. 4. Comparisons of BUBBLE and PROPHET on *Reality* dataset

number of copies created in the system for each message as we have done before for the comparison with the 'oblivious' algorithms. We did not count the control traffic created by PROPHET for exchanging routing table during each encounter, which can be huge if the system is large (PROPHET uses flat addressing for each node and its routing table contains entry for each known node). We can see that most of the time, BUBBLE achieves a similar delivery ratio to PROPHET, but with only half of the cost.

Considering that BUBBLE does not need to keep and update an routing table for each node pairs, the improvement is significant. Similar significant improvements by using BUBBLE are also observed in other datasets, these demonstrate the generality of the BUBBLE algorithm, but because of page limit, we can not include the results here.

6.2   Multi-point Communication

For validation and evaluation of the proposed approach, we use a discrete event emulator to replay the connectivity traces. The original trace files are divided into discrete sequential contact events and fed into the emulator as inputs. Although the current subscription model is simply topic-based, content-based filtering can be operated in the broker nodes. In the experiments, ten topics are predefined. Randomly selected nodes create 20 to 100 unique subscriptions, and 200 to 1000 publications unless stated otherwise. The message creation times are uniformly distributed throughout the experimental duration. The experiment is performed with MIT (100 devices) traces. Table 3 summarises the results of the *Socio-Aware Overlay* approach. The second column (*Average hops*) is hop counts per publication. The experiment with the MIT trace shows around 1.3 hops regardless of the scale of publication/subscription. The average pair distance of the network is 1.6 hops, which indicates that the *Socio-Aware Overlay* approach performs better than flooding to every subscriber by epidemic approach. The total hop count in the entire operation is shown in the final column (*Total Hops*). A pure epidemic approach results in larger hop counts. In the experiments, communication between brokers is assumed to use direct methods such as access-point WiFi or GPRS. This approach does not need to wait for the next contact with devices to communicate. Thus, if communication between brokers uses unicast or an epidemic approach, *Average hops* will increase. In the experiments, a group of brokers are used instead of a single broker in the community. This requires

Table 3. Event Dissemination with Socio-Aware Overlay

# Pub/Sub	Average Hops	Contact to Sub	Pub to Sub	Latency	Undelivered	Total Hops
1000/100	1.28	5.6 units	631.6 units	5.26 mins	261(26%)	6431
500/50	1.34	4.6 units	828.5 units	6.90 mins	242(48%)	1373
200/20	1.32	4.3 units	831.4 units	6.93 mins	115(58%)	204
1000/100C	1.35	2.7 units	449.4 units	3.75 mins	33(3%)	-

further work for balancing network work load of brokers and increasing reliability by replication of brokers.

Each publication has three stages during the simulation: (i) a publication is created at time unit (A), (ii) a publisher contacts the other devices to inject its publication to the network at time unit (B), and (iii) the publication is delivered to the subscriber at time unit (C). The timeline of a publication's life is depicted below:

A: Publication Created
B: Publisher → First Node Contact
C: Subscriber Received Publication

The third column (*Contact to Sub*) indicates $C - B$ in the number of time units. The fourth column (*Pub to Sub*) shows that total duration of publishing ($C - A$). A single time unit has a duration of 0.5 seconds, and *Latency* indicates the approximate latency in minutes. Thus, $C - A$ and $C - B$ are indicators of the latency of publications. $C - B$ is much smaller than $C - A$ and $C - A \approx B - A$. On average, it takes over 3 days to get a first contact from when a publication is ready. However, the majority of nodes gets much shorter waiting time until getting a first contact (see Fig. 5). Once the publication is passed to the contacted device, in a few minutes subscribers will receive a publication.

Fig. 5 depicts the distribution of values $C - A$ in three different settings of publication and subscription. The result shows a power law distribution indicating that most event dissemination has short durations. Fig. 6 depicts the distribution of values $C - B$ from publisher's and subscriber's aspects from an experiment with 1000 publications and 100 subscriptions. Certain subscribers (e.g. 70-80) have higher durations, which has various reasons such as that these nodes are away from the centrality nodes in the community (i.e. more than single hop distance), or these nodes may not be part of the community despite them being detected. This will require further investigation.
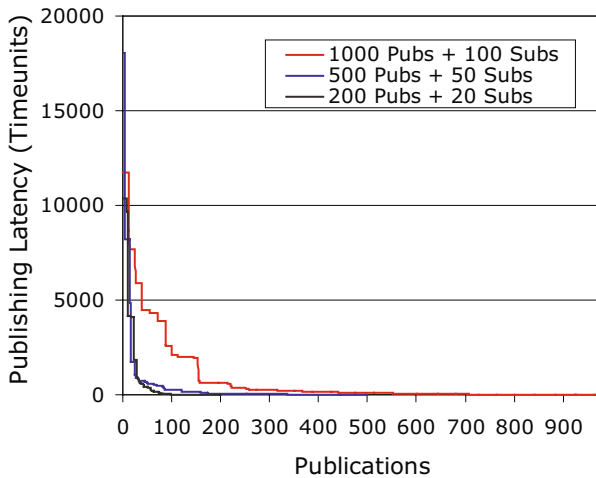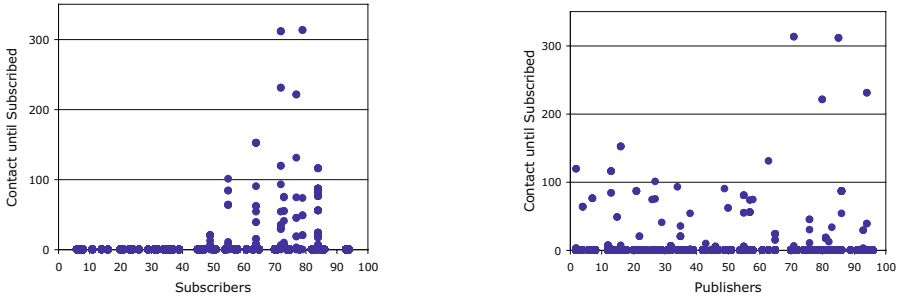

Fig. 5. Latency of Publications I
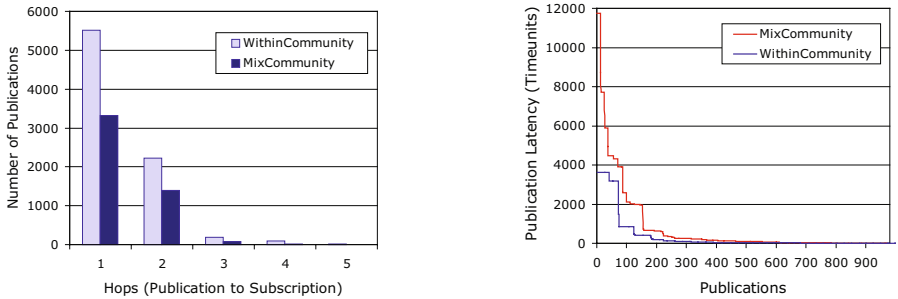
Fig. 6. Latency of Publications II

Fig. 7. Latency: Within and Mix Community

The value of *Undelivered* indicates the reliability of delivery. The ratio varies from 26% to 58% in 3 settings. The result shown in the last row of Table 3 has the same setting as the first row except publishers and subscribers are in the same communities. When both publishers and subscribers are in the same communities, the *Undelivered* ratio decreases significantly to 3%. In the real world, this may happen frequently as shared interest often creates communities.

Fig. 7 depicts a comparison of two different settings of publishers and subscribers. *MixCommunity* indicates publishers and subscribers are spread across different communities and *WithinCommunity* indicates 90% of both subscribers and publishers of the same topics reside within the same community. Fig. 7a depicts hop counts from publishers to subscribers and shows that topic sharing within communities gives higher reliability with delivery of events in fewer hops. Fig. 7b depicts the distribution of the latency of publications $(C - A)$. *MixCommunity* shows high value of latency of the few nodes. Fig. 7b fundamentally presents a power law distribution indicating that the majority of nodes have low latency.

7   Conclusion

We have shown empirically that identifying social communities enhances communication efficiency for complex mobile networks for both single-point com-

munication and multi-point communication. Social information can be further explored to provide better applications, for example a city-wide Pocket Switched Network, community-based media sharing application, and social-tagging metadata system for information searching. We are devoting two projects [2] to further characterise and understand social networks, and utilize this knowledge to develop more novel applications.

References

1. Hui, P., Chaintreau, A., et al.: Pocket switched networks and human mobility in conference environments. In: Proc. WDTN (2005)
2. Fall, K.: A delay-tolerant network architecture for challenged internets. In: Proc. SIGCOMM (2003)
3. Kempe, D., et al.: Connectivity and inference problems for temporal networks. J. Comput. Syst. Sci. 64(4), 820–842 (2002)
4. Okasha, S.: Altruism, group selection and correlated interaction. British Journal for the Philosophy of Science 56(4), 703–725 (2005)
5. Eagle, N., Pentland, A.: Reality mining: sensing complex social systems. Personal and Ubiquitous Computing V10(4), 255–268 (2006)
6. Chaintreau, A., Hui, P., et al.: Impact of human mobility on the design of opportunistic forwarding algorithms. In: Proc. INFOCOM (2006)
7. Leguay, J., Lindgren, A., et al.: Opportunistic content distribution in an urban setting. In: ACM CHANTS, pp. 205–212 (2006)
8. Gonzalez, M.C., Hidalgo, C.A., Barabasi, A.L.: Understanding individual human mobility patterns. Nature 453(7196), 779–782 (2008)
9. Newman, M.E.J.: Analysis of weighted networks. Physical Review E 70, 056131 (2004)
10. Freeman, L.C.: A set of measuring centrality based on betweenness. Sociometry 40, 35–41 (1977)
11. Liao, W.H., Tseng, Y.C., Lo, K.L., Sheu, J.P.: GeoGRID: A geocasting protocol for mobile ad hoc networks based on grid. Journal of Internet Technology 1(2) (2000)
12. Baldoni, R., Marchetti, C., Virgillito, A., Vitenberg, R.: Content-based publish/subscribe over structured overlay networks. In: Proc. ICDCS (2005)
13. Hui, P., Crowcroft, J.: How small labels create big improvements. In: Proc. IEEE ICMAN (2007)
14. Lindgren, A., Doria, A., et al.: Probabilistic routing in intermittently connected networks. In: Dini, P., Lorenz, P., de Souza, J.N. (eds.) SAPIR 2004. LNCS, vol. 3126, pp. 239–254. Springer, Heidelberg (2004)

---

[2] http://www.social-nets.eu/, http://www.amillionpeople.net/