Impact of Social Structure on Forwarding Algorithms in Opportunistic Networks

Ning Wang #1, Eiko Yoneki #2

#*Computer Laboratory, University of Cambridge*
United Kingdom
1ning.wang@cl.cam.ac.uk
2eiko.yoneki@cl.cam.ac.uk

***Abstract*—Opportunistic networks are formed among mobile wireless devices based on spontaneous connectivity such as mobile phone networks using short range radio. Different setting of social structure in such networks gives significant impact on the feasibility and performance. In this paper we aim at understanding how social structure affects forwarding algorithm in various opportunistic network configurations. Having human mobility traces from the real world, we focus on the social structure in terms of centrality and community. We exploit different community detection and centrality calculation from the trace to present the features of such networks. We study a collection of Social-based Forwarding algorithms, such as LABEL, RANK, and BUBBLE [10]. Furthermore, we implement those forwarding algorithms over a Xen-based Haggle testbed [9]. We investigate the impact of community structure and centrality on performance and demonstrate that social structure influences the performance of the social-based Forwarding algorithms. Our result demonstrates that it is important to find appropriate centrality and communities for social networks with complex structure in the design of the social-based data dissemination algorithms.**

I. Introduction

Human society can be characterized by diverse social structures, which are likely to be identified by many factors, such as geographical separation, population density, or spacial distribution. In this paper, we concentrate on two specific aspects of social metrics: community and centrality. The fact that people inherently form groups creates the concept of community, which is defined as partitions of network nodes into densely connected subgroups [15]. Community detection in complex networks has been investigated for a long time [17]. And numerous methods to detect community structures has been proposed and evaluated in centralized [16] and distributed [11] ways. Centrality is another notable pattern in social networks. Within a community, some people might have high centrality, which indicates that they appear to be more popular, and thus influence more people than others. Centrality defines the significance of a node in a network [8], and can be assessed by various metrics which may based on degree, betweenness, and closeness, etc.

With increasing number of mobile users carrying portable devices (e.g. smart phone, tablet computer), communication in future tend to be more pervasive and autonomous in the presence of opportunistic connectivity. This provides us a chance to construct Pocket Switched Networks (PSNs)[10] known as a subclass of Delay Tolerant Networks (DTNs) [12]. PSNs use opportunistic data dissemination to facilitate human-to-human communication in a dynamical and repeatedly disconnected network environment. The traditional networks are based on infrastructure-based networking, while there is no such infrastructure in PSNs. Thus, major challenge in PSNs lies in how to design routing protocols without end-to end path. The nodes in PSNs makes use of store-and-forward mechanism to carry data to the destination. Given that social characteristics are more stable than routing structures in such kinds of human networks, it is inspired to use social metrics to design forwarding algorithms. It has been proved that knowledge on social structure can help us to design good strategies for forwarding algorithm [10]. In our previous work, we have proposed various Social-based Forwarding algorithms, such as LABEL, RANK [10], and ultimately BUBBLE forwarding design space [10].

All of them are able to exploit social characteristics to facilitate data delivery to members of a specific community.

We foresee that the communication in PSNs is community-oriented, and encompasses diverse approaches, such as many-to-many, any-to-many and one-to-any. However, the Social-based Forwarding algorithms focus primarily on the end-to-end communication. Thus, we proposed Socio-Aware Overlay mainly focusing on multi-point communication in [20]. The Socio-Aware Overlay operates by creating a overlay with broker nodes based on centrality and community. The broker nodes apply multicast over the physical topology graph within communities in response to changes in the topology, and forward data between communities which are connected by the backbone for content sharing in disconnected environments. In this way, data dissemination is performed over the Socio-Aware Overlay in a publish/subscribe fashion.

There are numerous papers which have investigated the way to design discover social structure in human networks [16]. The performance of the Social-based Forwarding algorithms relies on how to choose suitable community structure and centrality. So there is a need to uncover optimized social structure from human encounter traces and study impact of social structure on various Social-based Forwarding algorithms. The contribution of this paper is as follows. The contribution of this paper is as follows. First, we perform analysis of the comparison of effects of different community detection on the performance of the Social-based Forwarding algorithms. Second, we study the performance of the Socio-Aware Overlay under different community detections and centralities.

The remainder of the paper is organized as follows. Section 2 describes related work in the area of forwarding and community detection in opportunistic networks. Section 3 introduces real-world human traces. Section 4 presents the Social-based Forwarding algorithms. Section 5 reviews social structure in terms of centrality and community detection. Section 6 presents the Socio-Aware Overlay. Section 7 gives the overview of implementation of the Social-based Forwarding algorithm and Haggle testbed. In Section 8, we perform an evaluation of the Social-based Forwarding algorithm and the Socio-Aware Overlay under different social structure configurations, which is followed by Conclusion Section.

II. Related Work

Several community detection techniques have been proposed and examined in the literatures.The recent reviews [17] has given as introductory reading in numerous centralized methods which are beneficial for offline data analysis on mobility traces collected to explore structures. Authors in [13] have reviewed, and compared the benefits of various definitions of centralities in contemporary social networks. Graphs have been proved as a powerful tool to represent social relations in a quantified and measurable manner [16]. [4] has introduced a proposal for a standard benchmark test of community detection methods, and uncovered that the most accurate methods tend to be more computationally expensive, and that both aspects need to be considered when choosing a method for practical purposes. [2] has defined a measure of local community structures and an algorithm that infers the hierarchy of communities by enclosing a given vertex by exploring the graph at a time. Hui et al. [11] have studied the area of distributed community detection for self-organizing networks, where the mobile devices are able to sense and detect their own local communities instead of relying on a centralized server.

Forwarding algorithms for DTNs and opportunistic networks have been extensive researched in recent years.[21] has overviewed a variety of routing strategies in DTNs, and utilized some extent of calculations to cope with complexity of network semantics such as location tracking or mobility. [14] has studied propose a probabilistic routing protocol for such networks. Unfortunately none of them use the derived forwarding from in-depth understanding of social structures. Several Social-based Forwarding algorithms for PSNs have been proposed. For example, LABEL algorithm [10] has presented a simple LABEL forwarding based on communities. [10] has demonstrated the RANK and the BUBBLE algorithms, which has used community structures and centrality for further improvement of forwarding algorithms.

There are many papers attempt to uncover social structure in PSNs from real world human connectivity traces. Community detection has been studied in centralized [16] and distributed [11] manners. [19] shows visualization of detected community structures uncovered by different community detection algorithms from human encounter traces. Nevertheless, there are not yet an in-depth study of how social characteristics detected in the real world human connectivity traces impacts the performance of forwarding algorithms and overlays. This is necessary because in PSNs the performance is not only determined by networking conditions but also by human activities. We take an empirical approach and work directly with the real world connectivity traces to investigate this issue.

III. Real-world Human Traces

In this paper, we choose traces from different sources: two experimental datasets collected by the Haggle [9] Project, known to as CAM, INFC06; one dataset from the MIT Reality Mining Project [5], referred to as MIT. These traces can be obtained from CRAWDAD [3]. These datasets have been extensively analyzed. Here we just refer the reader for further background information, which are listed below.

- MIT: in the MIT Reality Mining project [5], 97 smart phones were deployed to students and staff at MIT over a period of 9 months. These phones were running software that logged contacts.
- CAM: In the Haggle project, 36 iMotes were deployed to 1st year and 2nd year undergraduate students at the University of Cambridge for 10 days [10]. iMotes are sensor boards equipped Bluetooth for detecting proximity devices.
- INFC06: 78 iMotes were deployed at the Infocom 2006 conference for 4 days [10].

In order to make it easier for processing, we convert human mobility traces into weighted contact graphs [10] is based on the number of contacts. 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 number of contacts during the experiment.

IV. Social-based Forwarding

Social-based forwarding largely benefits from the social attributes such as community and centrality. In order to use the Social-based Forwarding, we assume that each node belongs to at least one community and has a ranking known as centrality. Nodes may belong to multiple communities and hence may have multiple rankings.

In this paper, we use 3 forwarding algorithms, namely LABEL, RANK, and BUBBLE [10].

- LABEL is a straightforward Social-based Forwarding scheme. It is assumed that every node possesses a label informing others about its affiliation/organization. With this, it compares label of the potential relay nodes and the label of the destination node, and consequently forwards the data objects to nodes that belong to the same community as the destinations. The problem for LABEL is obvious, i.e. , the source is probably not accessible to a member of the community of the destination in some scenarios where multiple communities are far away. More information about LABEL can be found in [10].

- RANK use node centrality as forwarding metric to forward data objects. It is assumed that each node knows only its own ranking and the rankings of those it encounters. So RANK simply forwards the data objects to nodes which have a higher ranking than the current node, until either the destination is reached, or the messages expire.
- BUBBLE is a family of approaches that combines the observed hierarchy of centrality of nodes and observed community structure with explicit labels [10]. Explicitly in this paper we combine both LABEL and RANK, i.e., we use RANK to disseminate the data objects and uses LABEL to reach the destination community.

BUBBLE could have many variances. In this paper BUBBLE operates as follows. If a node has a data object destined for node in the same community, BUBBLE will send it using normal forwarding. If a node has a data object destined for node in another community, this node would first send this data object up the hierarchical ranking tree using the ranking until it reaches a node which has the same label (community) as the destination of this message. Then the local forwarding system will be used to forward the data object until the destination is reached or the message expired.

V. Social Structure

Social structures can be determined by many features. Two specific aspects of social metrics, i.e. centrality and community, are chosen to develop the Social-based Forwarding algorithm and the Socio-Aware Overlay.

A. Centrality

Centrality indicates the significance of a node in accordance with graph theory. In this section, we present three well-recognized centrality measures: Degree, Betweenness, and Closeness Centrality.

- Degree centrality implies the total number of connects incident upon a node. Higher degree means that the node may be more popular and thus more important than others in the design of forwarding algorighms. [1] demonstrates that the degree distribution usually follows a power-law in social interaction graphs of movie actor collaboration, science collaboration, phone calls or graphs of sexual contacts.
- Betweenness centrality is the percentage of the number of shortest-paths that a node is part of, over all graph shortest-paths [8]. Nodes with high betweenness bridge two nonadjacent nodes and may impact on data flow between two communities. Thus, betweenness centrality is a key metric to investigate the data flow over different communities.
- Closeness centrality is defined as the average of the distance between node and all other nodes. It exhibits how many hops it will take data to spread the others in the community. The node with less closeness has the shortest path to all others and thus the best visibility in the network and sub-network (i.e. community).

B. Community Detection

There are several novel community detection approaches with great potential to detect both static and temporal communities [17]. Two centralized algorithms are used to discover the community structures in the real-world traces, namely, K-Clique [18] and Fiedler [7]. See [19] for details of community detection with the contact network traces.

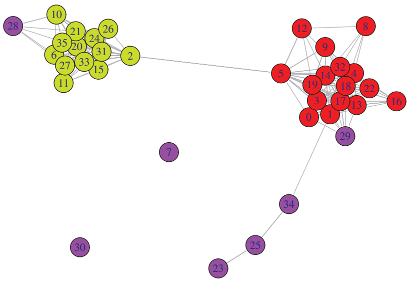

Fig. 1: K-Clique Community Detection for the CAM dataset

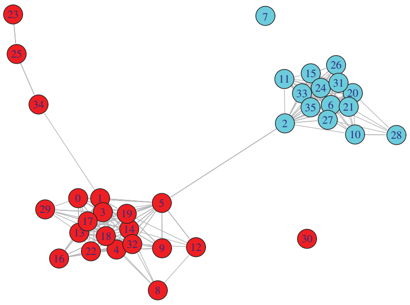

Fig. 2: Fiedler Community Detection for the CAM dataset

- K-Clique define a community as a union of all k-cliques (complete sub-graphs 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 [18]. An advantage of this approach is that it allows overlapping communities, which is useful as, in human society, one person may belong to multiple communities.
- Fiedler Clustering use Fiedler vector, which is the eigenvector for the nonzero smallest eigenvalue of a Laplacian matrix [7]. This vector can be used for decomposing graphs into structural components.

Figure 1 and 2 depicts connection maps in the CAM dataset for different algorithms of community definition. The nodes are colored with their respective communities found by K-Clique and Fiedler. Furthermore, we can observe that there are some devices that do not belong to any communities, named Loners.

VI. Socio-Aware Overlay

The Socio-Aware Overlay is a publish/subscribe system for multi-point dissemination introduced in [20]. The fundamental idea of the Socio-Aware Overlay is to take advantage of the prevailing social structures to construct an overlay based upon centrality and community detection rather than various

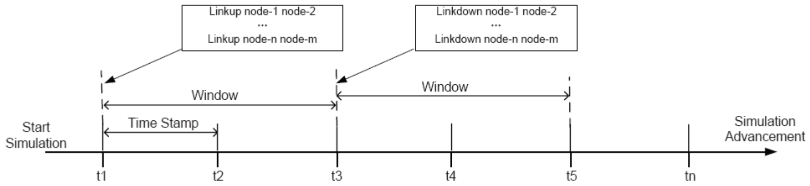

Fig. 4: A generic sequential procedure for Haggle testbed

contexts (e.g. location, group mobility). Figure 3 depicts a publish/subscribe broker overlay, which is constructed through the community detection and centrality nodes.

The Socio-Aware Overlay employs the concept of community and centrality. It operates as follows. First, community detection is used to divide whole networks into several communities, which is in turn mapped to the overlay. This is because people within the community are more likely to same data with each other, and similar subscriptions may coexist within the same community. Thus performance of such system can be improved. Second, centrality nodes, namely brokers, are identified to exchange data between communities. We currently select closeness centrality nodes for the broker node as closeness centrality implies the best visibility inside the community [20]. So each node has a local view of the community. In this way, once the broker receives the message, it will forward it to any member of the community with reliability. Thanks to the characteristics of human networks (i.e. scale-free networks), many nodes within a community are tightly connected and multiple closeness centrality nodes can coexist. Thus multiple brokers can exist in the same community, which provides a potential load balance for data forwarding.Our Socio-Aware Overlay is mapped over detected communities, which gives a certain level of stable network topology. Communication between broker nodes might have two modes: Unicast and Direct. Unicast is dependent on the underlying unicast algorithms, which might end up with epidemic routing. Direct offers a more direct communication mechanism, which gives increased performance of message delivery sacrificed with some computing costs. Our Socio-Aware Overlay operates in direct mode, i.e.,the nodes within the community sends data object to the broker node. On receipt of data object, the broker node employs LABEL forwarding algorithm by matching the LABEL of target to the LABEL of other brokers to efficient deliver data object to the member within other communities. Therefore, the performance of this approach is determined not only by the real mapping between the overlay network but also by the broker nodes.

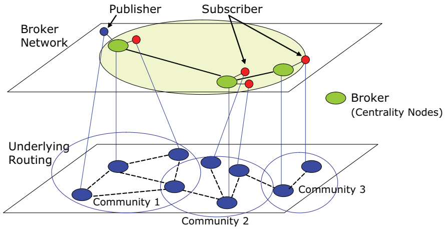

Fig. 3: **Overlay over Communities**

VII. Implementation

A. *Forwarding Algorithm on Haggle Platform*

Haggle [9] is a data-centric communication architecture for opportunistic networks. It is comprised of six independent Managers including data manager, naming manager, forwarding manager, resource manager, connectivity manager, and protocol manager.

Since forwarding manager is responsible for delivering data object to the destinations, we have implemented three new Social-based Forwarding managers, namely, ForwarderLabel, ForwarderRank, and ForwarderBubble, and Socio-Aware Overlay manager, ForwarderOverlay on Haggle platform. In this section, we mainly introduce the implementation of ForwarderBubble in Haggle framework because it combines the features of both LABEL and Rank.

The ForwarderBubble class inherited from the ForwarderAsynchronous class, is responsible for forwarding the routing information and data objects. It operates as follows. Whenever a new data object has been created by the data manager or may has been received from other nodes, the ForwarderBubble verifies if the target is the node which is currently a neighbor (i.e., reachable at least through one of the available interfaces). If this the case, the data object is forwarded to it. Otherwise, it use _generateDelegatesFor() method to locate the ideal delegates according to BUBBLE algorithm and set them as relay nodes for the data objects.

The mechanism to exchange social-based routing information has been developed in the ForwarderBubble class. Two new fields in the routing information are defined as Label and Rank to contain social-based metrics in routing information . All nodes exchange their routing information as a exclusive type of data objects immediately after they encounter each others. Every node needs to maintain a forwarding list to map encountered node_id to forwarding metrics. As soon as receiving new routing information, the nodes have to add the new node_id and its metrics to its forwarding list.

B. *Haggle testbed*

The Haggle testbed is based on Xen emulator, which is built on a PC with an 8-core CPU and 24GB RAM running Debian Squeeze. It can be used to simulate a number of virtual nodes running dependently with small amounts of RAM. The settings of the testbed is customized by predetermined scenario file, which specifies the comprehensive behaviors of simulation, i.e. number of nodes, change of topology, and applications running on each node. It generates a huge selection of results, usually demonstrating the entire distribution of possible outcomes.

The changes of topology in the Haggle testbed can be modeled as a series of discrete sequential contact events. The original trace files can be converted into configuration files, which contain the events that determine the statue of link between two nodes and the order where these events occur along with the time. The testbed is fed up with these events as a sequence of activities in preprocess procedure, and perform these activities, in between which it waits in queue for durations which are normally predefined by user before simulation.

VIII. RESULT AND EVALUATION

The evaluations in this paper mainly focus on: 1) the comparison of effects of different community detection on the performance of the Social-based Forwarding algorithms, 2) the performance of the Socio-Aware Overlay under different community detection and centrality. The evaluation is performed on the Haggle testbed. The experiment is performed with the CAM, the INFC06, and the MIT traces. We extract a 2 week session during term time from the whole dataset in order to reduce simulation time. Since social contacts in our trace are highly variable, we average the results with 5 iterations of the experiments.

A. Performance Metrics

We utilize several metrics for the above evaluation:

- Hop Counts are calculated as the number of hops the data objects take from source to destination.
- Delivery Success Ratio is calculated as the number of successfully received data objects divided by the total number of sent data objects over all nodes in the networks.

B. Social-based Forwarding

In this section, we evaluate the impact of social structure on the performance of the Social-based Forwarding algorithms. We use K-Clique and Fiedler to detect community against different connectivity traces, and examine the LABEL, the RANK and the BUBBLE forwarding algorithms.

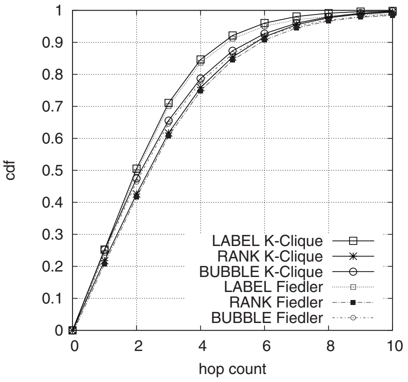

Fig. 5: CDF of hop counts for the Social-based Forwarding (CAM dataset)

Figure 5 presents the CDF of hop counts for the Social-based Forwarding in the CAM dataset. It illustrates that BUBBLE with Fielder delivers the better performance with 50% of hop counts within less than 2 hops. It also reveals that for CAM dataset K-Clique performs almost as well as Fiedler. This is due to that CAM trace has relatively simple community structure and the impact of community structure on performance of forwarding is reasonably minor.

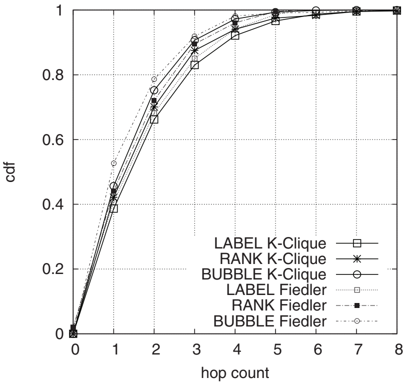

Fig. 6: CDF of hop counts for the Social-based Forwarding (INFC06 dataset)

We use the INFC06 dataset to study more complicated community structure. Figure 6 shows the CDF graph of hop counts for the INFC06 trace. K-Clique has detected 6 communities, while Fiedler has detected 8 communities. We can see from Figure 6 that Fiedler community detection performs very similarly to K-Clique, and Fielder with BUBBLE gives a slightly better performance (80% of hop counts within less than 2 hops) than the others.

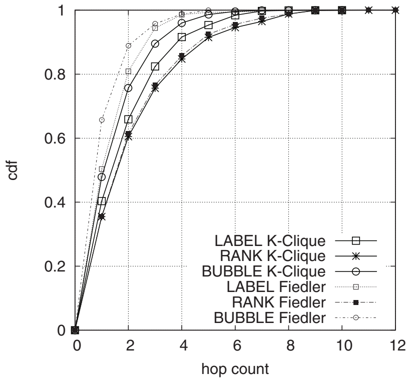

Fig. 7: CDF of hop counts for the Social-based Forwarding (MIT dataset)

Figure 7 shows the CDF of hop counts at the time data are delivered to the destinations for the MIT trace. There is a total 8 communities detected by both K-Clique and Fielder in the MIT trace. It can be seen that BUBBLE with Fiedler gives the best performance in terms of hop counts, i.e., there are 90% of data objects delivered with 2 hop counts for BUBBLE with Fiedler. In addition, LABEL with Fiedler tend to be delivered 80% within 2 hop counts. We can see that Fiedler community detection outperforms K-Clique in the MIT dataset.

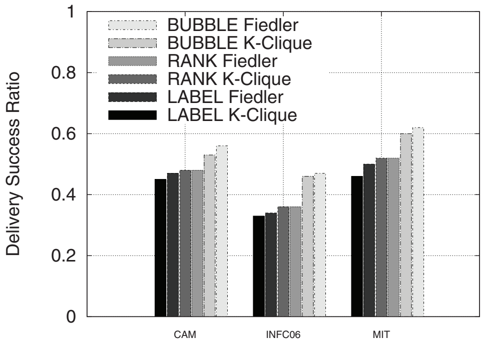

Fig. 8: Delivery Success Ratio for the Social-based Forwarding

Figure 8 presents the delivery success ratio comparing Fiedler and K-Clique community detection. We can see that the BUBBLE algorithms gives higher delivery ratio and thus are much more efficient than RANK and LABEL. It demonstrates that the most of the time Fiedler community detection can improve the performance when combine with BUBBLE. Even the difference of the performance between K-Clique based communities and Fiedler clustering based communities is less than 10% range, it indicates the difference will be amplified when the scale of experiments gets larger.

C. Socio-Aware Overlay

In this section we evaluate the performance of the Socio-Aware Overlay in comparison of different communities and centralities detected in real-world human traces. We have chosen two different methods to extract centrality nodes. The first one is based on the concept of closeness proposed in [8]. This represents the node with the shortest path to all others and the best visibility in the network (i.e. community). We have calculated the closeness $C_C(a)$ for a vertex a is inverse sum of distances to other nodes b:

$$C_C(a) = \ 1 \ / \ \sum_b d_{ab} \tag{1}$$

The second method (J-Algorithm) is based on the technique called Joint diagonalization (JD), which is a technique used to estimate an average eigenspace of a set of matrices. The average eigenspace is used to construct a graph which represents the *average spanning tree* of the network or a representation of the most common propagation paths. We then examine the distribution of deviations from the average and find that this distribution in real-world contact networks is multi-modal; thus indicating several modes in the underlying network. These modes are identified and are found to correspond to particular times. Thus JD may be used to decompose the behaviour, in time, of contact networks and produce average static graphs for each time. See [6] for further information of this technique.

We exploit the above two methods and employ K-Clique and Fiedler to detect community against different connectivity traces.

Figure 9 depicts the CDF of hop counts for the Socio-Aware in the CAM trace. It reveals that in the CAM dataset

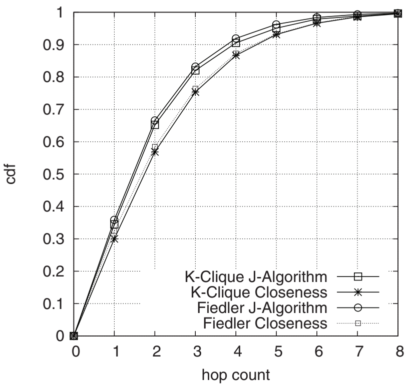

Fig. 9: CDF of hop counts for the Socio-Aware Overlay (CAM dataset)

the performance of K-Clique gives a similar performance of hop counts to Fiedler. This is due to the fact community structure in the CAM are fairly simply (only 2 communities). On the other hand, J-Algorithm provides better performance with 80% of hop counts less than 3 hops. We can discover that the impact of centrality on performance of forwarding is reasonably significant in the dataset where simple community structure can be observed.

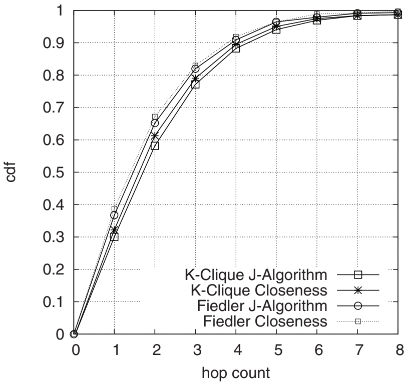

Fig. 10: CDF of hop counts for the Socio-Aware Overlay (INFC06 dataset)

Figure 10 exhibits the the CDF of hop counts for the Socio-Aware Overlay in the INFC06 dataset. One interesting observation is that Fiedler gives better performance than K-Clique, whilst J-Algorithm and closeness present almost the same performance. This implies that in the INF06 dataset community gives more significant impact on the performance in terms of hop counts, and the centrality nodes has less impact on overall performance. Because there are maximum 8 communities in the INF06 trace and the largest community only has 8 members. As a result, the community members number is relatively small so that centrality nodes is easier to achieve less hop counts by spreading data within community.

Figure 11 presents the CDF of hop counts for the Socio-Aware Overlay in the MIT dataset. We can view that the Fiedler with closeness centrality gives best performance in hop counts by delivering more than 90% within 2 hops. It is noticed that in the MIT dataset closeness and Fiedler

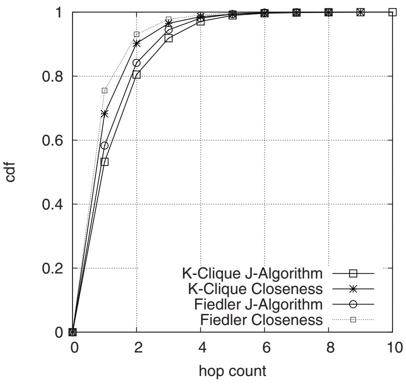

Fig. 11: CDF of hop counts for the Socio-Aware Overlay (MIT dataset)

performs better than J-Algorithm and K-Clique. The plot also clearly conveys that different community structures and centralities gives distinctive performance. This is due to that 8 communities are detected in the MIT trace, and the largest community contains 21 members. We conclude that centrality plays a vital role for maintaining the performance of the Socio-Aware Overlay in terms of hop counts for the dataset with large community members.

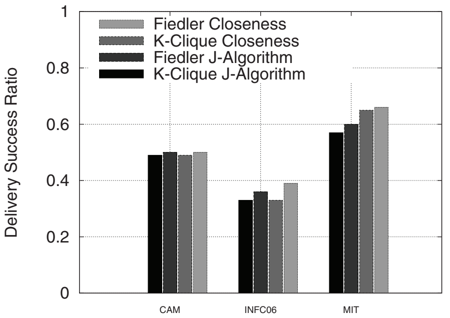

Fig. 12: Delivery Success Ratio for the Socio-Aware Overlay

We assess the delivery success ratio for Socio-Aware Overlay in Figure 12. It is observed that the MIT trace gives highest delivery success ratio with closeness centrality being higher than J-Algorithm. We can also see that for all three dataset, Fielder performs better than K-Clique.

IX. Conclusions and Future work

In this paper we have investigated the impact of social structure on forwarding algorithms in opportunistic networks by exploiting the real-network trace of human contacts. We have reviewed a collection of the Social-based Forwarding algorithms, such as LABEL, RANK and BUBBLE and also investigated the Socio-Aware Overlay. The implementation and experiments have been done over Xen-based Haggle testbed. The Social-based Forwarding algorithms have been integrated with the Haggle platform. We have revealed that social structures influence the performance of the the Social-based Forwarding. We have observed that BUBBLE combined with Fielder community detection delivers better performance in terms of hop counts and delivery ratio. The result demonstrated in this paper deals with fairly small scale networks, but it is consistent. Thus, we expect that larger scale experiments will result in significant performance improvement. The Fiedler community structure is based on hierarchical structure and the result indicates spreading information in hierarchical manner in the real society. We also express that the centrality nodes plays a vital role for maintaining the performance of the Socio-Aware Overlay for the dataset. Finally we come to the conclusion that it is crucial to choose appropriate social structures (centrality and communities), which will in turn improve the performance of the Social-based Forwarding and the Socio-Aware Overlay for social networks with complex structure. We are working on running experiments in a larger scale including synthetic dynamic network traces based on the extracted characteristics of real world traces.

Acknowledgment

This research is part-funded by the EU grants for the RECOGNITION project (FP7-ICT 257756) and the EPSRC DDEPI Project, EP/H003959.

References

[1] A. Barab aacute si. Emergence of Scaling in Random Networks. *Science*, 286(5439):509–512, oct 1999.

[2] A. Clauset. Finding local community structure in networks. *Physical Review E*, Jan 2005.

[3] CRAWDAD. http://crawdad.cs.dartmouth.edu/.

[4] L. Danon, J. Duch, A. Diaz-Guilera, and A. Arenas. Comparing community structure identification. *arXiv.org*, cond-mat.dis-nn, may 2005.

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

[6] D. Fay, J. Kunegis, and E. Yoneki. On Joint Diagonalization in Network. In *WIDS*, 2011.

[7] M. Fiedler. A property of eigenvectors of nonnegative symmetric matrices and its application to graph theory. *Czech Math J.*, 25, 1975.

[8] L. C. Freeman and L. C. Freeman. A Set of Measures of Centrality Based on Betweenness. *Sociometry*, 1977.

[9] Haggle Project (EU FP6). http://www.haggleproject.org/.

[10] P. Hui, J. Crowcroft, and E. Yoneki. BUBBLE Rap: Social-based Forwarding in Delay Tolerant Networks. In *MobiHoc*, 2008.

[11] P. Hui, E. Yoneki, S. Chan, and J. Crowcroft. Distributed community detection in delay tolerant networks. *MobiArch*, pages 1–8, 2007.

[12] IETF. Delay Tolerant Network Research Group (NTNRG), 2004. http://www.dtnrg.org/.

[13] E. Le Merrer. Centralities. *EuroSys Workshop on Social Network*, 2009.

[14] A. Lindgren, A. Doria, and O. Schelén. Probabilistic routing in intermittently connected networks. *ACM SIGMOBILE Mobile Computing and Communications Review*, 7(3):19–20, jul 2003.

[15] M. Newman. Finding and evaluating community structure in networks. *Physical Review E*, Jan 2004.

[16] M. Newman. Finding community structure in networks using the eigenvectors of matrices. *Physical Review E*, 2006.

[17] M. E. J. Newman. Detecting community structure in networks. *European Physical Journal B*, 38(2):321–330, 2004.

[18] G. Palla, I. Derenyi, I. Farkas, and T. Vicsek. Uncovering the overlapping community structure of complex networks in nature and society. *arXiv.org*, physics.soc-ph, jun 2005.

[19] E. Yoneki. Visualizing communities and centralities from encounter traces. In *ACM MobiCom Workshop on Challenged Networks (CHANTS)*, 2008.

[20] E. Yoneki, P. Hui, S. Chan, and J. Crowcroft. A socio-aware overlay for publish/subscribe communication in delay tolerant networks. *MSWiM*, pages 225–234, 2007.

[21] Z. Zhang. Routing in intermittently connected mobile ad hoc networks and delay tolerant networks: overview and challenges. *IEEE Communications Surveys Tutorials*, 8(1):24–37, 2006.