Detecting Dynamic Communities in Opportunistic Networks

Kuang Xu*, Guang-Hua Yang*, Victor O.K. Li*, Shu-Yan Chan†
*The University of Hong Kong †University of Cambridge

***Abstract*— In opportunistic networks, communities of mobile entities may be utilized to improve the efficiency of message forwarding. However, identifying communities that are dynamically changing in mobile environment is non-trivial. Based on random walk on graphs, in this paper we present a community detection algorithm that takes into account the aging and weight of contacts between mobile entities. Our idea originates from message-forwarding operations in opportunistic networks. We evaluate the algorithm on both computer-generated networks and real-world human mobility traces. The result shows that our proposed algorithm can find the communities and detect the changes in their structures over time.**

***Index Terms*— Opportunistic network, community detection, message forwarding, algorithm.**

I. Introduction

Communities are submodules or substructures that are functionally important to their corresponding networks. Categorizing objects into communities has numerous potential applications. In protein-protein networks, identifying modules that have significant functions [30] can provide valuable insights in developing new therapeutics to fight disease; in social networks, grouping individuals into communities [31] can help to highlight interaction patterns and common attributes among them; in the World Wide Web, capturing clusters of related pages [12] can aid the design of efficient search engine and content filtering; and in mobile networks, identifying communities of mobile entities can help improve the efficiency of message forwarding [9] [14] [33] so as to facilitate opportunistic communication. In the literature, a host of community detection algorithms has been proposed [8] [22] [23] [26], by agglomerative clustering [16], divisive partitioning [25], clique percolation [26], or maximization of certain measure function [24]. However, detecting communities in networks is non-trivial. While the majority of existing work addresses community detection in static network and focuses on its structural properties, the study for dynamic network that continually changes is still lacking.

Opportunistic network that utilize short-range wireless radio connection is a variant of disruption tolerant network (DTN) [17], in which portable devices exploit contact opportunities to communicate with each other without network infrastructure. Its underlying network, from a node's perspective, is characterized by contact time and inter-contact time between entities, which are, respectively, the durations that two entities stay within and beyond their connection range. To study a more macroscopic characteristic of opportunistic networks, the authors in [15] present a community detection approach for opportunistic networks by adapting several classic community finding algorithms. Through simulation on several human mobility traces, they find that the adapted algorithm could give satisfactory performance by adjusting certain parameters. However, human mobile networks are dynamic in nature, and two general problems for finding communities in opportunistic network require further considerations:

- *Continual change*. Opportunistic network is by nature non-static, in which the communities are dynamically evolving over time. These dynamics involve community growth, contraction, merging, splitting, etc. For routing in opportunstic netoworks, the recently proposed schemes [9] [14] explicitly utilize the notion of community of contacts, by aggregating all the contacts in the past (or considering there is an edge between two entities if at least one contact between them has occurred in the past). But this may not properly reflect the existing and changing community structure [29]. Since certain contacts in history would become irrelevant in the current network, taking into consideration the aging of contacts is necessary. Hence, it is important to study the evolution of communities in opportunistic networks.
- *Weighted interaction*. In opportunistic networks, contact and inter-contact duration could be transformed as the weights of interactions between entities, which play an important role in forming and dividing communities. A common approach [15] to simplify the weight is to convert a weighted graph to an unweighted one by utilizing certain threshold to discard edges with weights below it. This approach inevitably incurs inaccuracies since it does not differentiate the strengths of the remaining contacts and omits the importance of those discarded *weak* links [13]. Hence an effective approach to capture all the weighted contacts is necessary.

In this paper, we focus on the dynamics of communities, presenting a community detection algorithm for opportunistic networks. We formulate message forwarding in opportunistic networks as random walks on weighted finite graphs, based on which we utilize the *normalized commute time* as the criterion to extract communities from its underlying network. This approach takes into consideration the weighted contacts (we also refer to them as interactions in the rest of this paper) between mobile entities. To account for the temporal attribute, namely, the time-varying nature, of opportunistic networks, we propose a method that ages interactions between entities

978-1-4244-4216-4/09/$25.00 © 2009 IEEE

over time. The performance of this method can be adjusted by the introduced aging coefficient τ. We evaluate our proposed algorithm on both computer-generated mobile social networks and real-world human mobility traces gathered in two research projects (i.e. *Haggle* [1] and *Reality Mining* [2]). The result shows that our proposed algorithm can find the expected communities and detect the changes in their structures over time.

We proceed in this paper as follows. In Section II we review the work related to the dynamics and evolution of communities. Then we describe our community detection algorithm in Section III. The evaluation of our algorithm and discussion of results are presented in Section IV. Finally we conclude this paper and discuss the potential future work in Section V.

II. Related Work

Dynamics of communities have been studied in the literature. For example, the authors in [19] infer interactions within a university network from the email headers recorded over time, and find that network evolution is mainly affected by the network topology and its organizational structure. By utilizing temporal information on interactions, the authors in [4] [29] propose a framework to identify communities and analyze their evolutions in dynamic social networks. They formulate the problem of finding developing communities from the observed interactions as a combinatorial optimization problem. The above work provides an insightful investigation on the dynamic communities in social networks, but our work differs from it since we focus on the clustering behavior of objects in opportunistic networks. By utilizing weighted contacts between mobile entities, we aim at uncovering time-varying communities that would facilitate information dissemination [14] in human social life. Community dynamics have also been studied in online social networks. In [5] [7], communities are identified within some time windows and then merged to reflect their evolution, and heuristics are proposed to approximate the optimal solution. The authors in [3] resort to several large on-line datasets that embed explicit user-defined communities, finding that the underlying structure is the key that affects whether an individual would join a community. We do not focus on how the structural feature influences the development of communities, but study the continually changing communities in human mobile networks, accounting for the aging of interactions. The work in [27] accounts for time variability of the information from mobile networks, as in [2], regarding a community as a densely connected subgraph over time, and a node as its member only when it attaches to it in a series of time steps. Our study is motivated by message forwarding in opportunistic networks, developing a community detection algorithm that takes into account the aging and weight of contacts between mobile entities.

III. Community Detection Algorithm

We seek to develop an algorithm that can, based on contacts/interaction patterns between entities in opportunistic networks, detect their communities and the change in their structures over time. The framework comprises two major procedures:

1) Transform contacts/interactions between mobile entities into weights of edges in a graph, to account for their temporal connections. To achieve this, we develop a mechanism that takes into consideration the aging of contacts between entities.
2) Apply a node-grouping mechanism to the obtained weighted graph. We propose a mechanism that is motivated by random walk on finite graphs. We will show that this mechanism can be easily realized by resorting to the Laplacian eigensystem of a graph.

For a better presentation of this algorithm, in the following we first introduce how we group mobile entities into communities (Section III-A), and then present how we transform contacts between mobile entities into a weighted graph while maintaining their temporal properties (Section III-B).

A. Nodes grouping

In this section, we first formulate message forwarding in opportunistic networks as random walk on finite graphs, based on which we elaborate how certain property from random walk can be utilized to study the structure of communities of mobile entities in opportunistic networks. Then we introduce a general technique (resistance distance on electrical networks), which utilizes the Laplacian eigensystem of a graph, to realize this algorithm. To present this technique properly, we also briefly review the basics of spectral analysis of Laplacian matrix.

Consider a finite connected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ is the set of nodes (i.e. mobile entities that forward messages) and $\mathcal{E}$ is the set of edges. An edge means that message-forwarding operation is possible between the pair of nodes (i.e. by short range radio transmission). Let $n = |\mathcal{V}|$ and $m = |\mathcal{E}|$ be the size of $\mathcal{V}$ and $\mathcal{E}$, respectively. We also denote by $\mathcal{N}_u$ the set of neighbors of node u, and $k_u = |\mathcal{N}_u|$. Note that $\mathcal{N}_u \subseteq \mathcal{V}$. In random walk based forwarding, a message stays at a node for certain time before opportunities arise to travel to the next neighboring node. The message-forwarding operation at the current hop is independent of those in previous hops. For each pair of neighboring nodes $u, v \in \mathcal{V}$, we denote s_{uv} the expected sojourn time of a message forwarded from u to v. For each edge (u, v), we assign a weight w_{uv} to represent the availability of message-forwarding operations between u and v. In this paper we consider undirected graphs such that $w_{uv} = w_{vu}$. We define the probability that a message will be forwarded from node u to its neighbor $v \in \mathcal{N}_u$ as:

$$p_{uv} \triangleq \frac{w_{uv}}{\sum_{v' \in \mathcal{N}_u} w_{uv'}}$$

In opportunistic networks, the contact/interaction between mobile entities are characterized by their contact time and inter-contact time. Apparently, the longer two mobile entities stay together (i.e. within radio transmission range) or the more frequently they meet each other, the more opportunities for message-forwarding operations between them. Hence, we assume that the chance of message-forwarding between two

[1]http://www.haggleproject.org.

[2]http://reality.media.mit.edu/.

neighboring nodes is in proportion to their contact duration or contact frequency, such that:

$$\begin{cases} w_{uv} = \dfrac{1}{s_{uv}} \\ s_{uv} = \dfrac{1}{c \cdot l_{uv}} \end{cases} \tag{1}$$

where s_{uv} denotes the expected sojourn time to forward a message from u to v, and $l_{uv} > 0$ is obtained from a series of contacts, accounting for the temporal attribute of interactions between mobile entities. We will introduce l_{uv} in Section III-B. c in the above equation is a normalization factor.

A random walk on a weighted undirected graph is actually a finite Markov chain that is time-reversible [20]. The hitting time H_{uv} is the expected number of hops a message takes to traverse to v, starting from u. It can be expressed recursively by [3]:

$$H_{uv} = \begin{cases} \sum_{w \in \mathcal{N}_u} p_{uw}(s_{uw} + H_{wv}) \\ \quad = s_u + \sum_{w \in \mathcal{N}_u} p_{uw} H_{wv}, & u \neq v, \\ 0, & u = v. \end{cases} \tag{2}$$

where $s_u \triangleq \sum_{w \in \mathcal{N}_u} p_{uw} \cdot s_{uw}$ is the expected sojourn time at node u. Note that normally $H_{uv} \neq H_{vu}$. The commute time

$$C_{uv} = H_{uv} + H_{vu} \tag{3}$$

is the round-trip time between node u and node v (i.e. expected number of hops a message travels from u to v and then get back to u). Then for a network with a finite number of entities, we can obtain a commute time matrix C. In this paper, we use the *normalized* commute time matrix C' to identify communities of mobile entities, in such a way that the shorter the commute time between two entities, the more confidently we can group them into the same community. Intuitively, the shorter the round-trip time between two entities, the closer they tend to stay with each other, with more possibilities for messaging forwarding between them. We will show later that since we consider a connected graph, the commute time of a message from any given node u to any other node v is positive. However, obtaining C' from Eqns. (2) and (3) is non-trivial. In the following we present an approach to obtain C' in terms of resistance distance [32] on electrical networks, which is related to the Laplacian eigensystem of a graph. We first give a brief review of the basics of Laplacian spectral theory.

Consider the above obtained weighted graph $\mathcal{G}$. Its corresponding Laplacian matrix, denoted by L, is a square matrix of order n with entry (u, v) defined by:

$$L_{uv} = \begin{cases} d_u, & u = v, \\ -w_{uv}, & u \text{ and } v \text{ are adjacent}, \\ 0, & \text{otherwise}. \end{cases} \tag{4}$$

[3]We consider a slotted system such that one hop takes place in one slot time.

where $d_u = \sum_{v' \in \mathcal{N}_u} w_{uv'}$ is the weighted degree of node u (total weight of the edges adjacent to u). The Laplacian matrix L of the graph $\mathcal{G}$ is a positive semi-definite singular matrix that has no inverse. It has n eigenvalues λ_i and eigenvectors ϕ_i, referred to as Laplacian eigenvalues and Laplacian eigenvectors of $\mathcal{G}$. We label the Laplacian eigenvalues in decreasing order

$$\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_n,$$

then we have $\lambda_n = 0$ and the corresponding eigenvectors $\phi_n = (1, 1, \cdots, 1)/\sqrt{n}$. The second smallest Laplacian eigenvalue λ_n represents the connectivity [21] of graph $\mathcal{G}$. $\lambda_{n-1} \neq 0$ if and only if $\mathcal{G}$ is connected. In this paper we only consider connected graph (for a graph that is not connected, we focus on its largest connected component). The resistance distance r_{ij} in an electrical network, constructed so as to correspond to $\mathcal{G}$, is defined as the voltage difference between i and j when a unit current is injected at i and removed from j. Note that $i, j \in \mathcal{V}$ may not be adjacent. For a connected graph, r_{ij} can be expressed as a function of its Laplacian eigenvalues and Laplacian eigenvectors [18]:

$$r_{ij} = \sum_{k=1}^{n-1} \frac{1}{\lambda_k} (\phi_{ki} - \phi_{kj})^2. \tag{5}$$

In this paper, we use the above technique to obtain the normalized commute time matrix C'. We assign each edge $(u, v) \in \mathcal{E}$ a conductance of value w_{uv}. For the commute time C_{uv} between any two vertices u and v in the graph $\mathcal{G}$, Chandra *et al.* [6] has shown that it is precisely characterized by the resistance distance between them:

$$C_{uv} = r_{uv} \sum_{(u',v') \in \mathcal{E}} w_{u'v'}(s_{u'v'} + s_{v'u'}). \tag{6}$$

Eqns. (1), (5) and (6) provide a direct and simple way to compute the matrix C' of any connected weighted graph, which is expressed in terms of the Laplacian eigensystem of a graph:

$$\begin{aligned} C'_{uv} &= \frac{r_{uv}}{\max_{(u',v') \in \mathcal{E}} C_{u'v'}} \cdot \sum_{(u',v') \in \mathcal{E}} w_{u'v'}(s_{u'v'} + s_{v'u'}) \\ &= \frac{1}{C_{max}} \cdot \sum_{k=1}^{n-1} \frac{1}{\lambda_k} (\phi_{ki} - \phi_{kj})^2 \sum_{(u',v') \in \mathcal{E}} w_{u'v'}(s_{u'v'} + s_{v'u'}) \\ &= \frac{2m}{C_{max}} \sum_{k=1}^{n-1} \frac{1}{\lambda_k} (\phi_{ki} - \phi_{kj})^2 \end{aligned} \tag{7}$$

In other words, C' is expressed solely in terms of the matrix l that describes the interaction patterns of mobile entities in opportunistic networks. Based on Eqn. (7), we utilize *weighted pair group method with arithmetic mean (WPGMA)* [28] to create a hierarchical cluster tree, which uses the average distance between all pairs of nodes in any two clusters, according to the following distance function:

$$d(R, S) = \frac{1}{|R| \cdot |S|} \sum_{u \in R} \sum_{v \in S} C_{uv} \ ,$$

where R and S are two different clusters. Finally, we horizontally cut through the tree at certain height that leaves the well-formed communities.

B. Contacts transformation

In opportunistic networks, communities of mobile entities are evolving, and community membership will be broken or newly formed over time. To account for these dynamics of interactions between mobile entities, we introduce the interaction matrix l to capture the temporal attribute of contacts between entities, which functions as the input of the procedure in Section III-A. The matrix l is obtained from contact duration and/or contact frequency of mobile entities. Since certain contacts in history would become irrelevant in the current network, it is necessary to account for their aging. In this section, we propose a formula that transforms contacts into the interaction matrix l.

Let us still consider the graph $\mathcal{G}$ in Section III-A, abstracted so as to correspond to an opportunistic network. For each pair of nodes $u, v \in \mathcal{V}$, we denote by l_{uv}^t the interaction weight between u and v at time t. l_{uv}^t constitutes the interaction matrix l (l_{uv} could be initialized to zero). We also assign two variables to the node pair: a contact recorder σ_t that records the contacts between them at time step t, and an aging coefficient $\tau > 0$ that indicates the aging granularity of contacts. At each time step, the contacts (i.e contact duration or frequency) between u and v are recorded as σ_t, and inserted into the interaction weight function at the next time step according to the following formula:

$$l_{uv}^t = \frac{\tau \cdot l_{uv}^{t-1} + q\sigma_{t-1}}{\tau + 1}, \tag{8}$$

where q is a normalization factor. l_{uv}^t takes into account all the contact histories up to t. The algorithm needs to be able to *forget* old associations between nodes when they fade away, yet have to be resilient to the temporal fluctuation. Consider after time t_e that there is no further contact between node u and v, then the interaction weight l_{uv} ages exponentially according to:

$$\begin{aligned} l_{uv}^t &= l_{uv}^{t-1} \cdot \tau/(\tau+1) \\ &= l_{uv}^{t_e} \cdot (\tau/\tau+1)^{(t-t_e)} \\ &= l_{uv}^{t_e} \cdot e^{-\gamma(t-t_e)}, \end{aligned} \tag{9}$$

and the half-life of the decay is thus $\frac{\ln 2}{\gamma} + t_e$.

When the aging coefficient τ is less than 1, the interaction history ages rapidly since for each new stage, the influence of the entire contact history is less than that of the new contact statistic. On the other hand, if τ is set greater than 1, the resulting interaction weight will be more resilient to temporal fluctuation as more emphases are put on the history than the new contact statistic, yet the influence of past contacts will still fade away over time. The algorithm allows each pair of nodes to have their own aging coefficient τ, which can be updated in response to the change in the interaction pattern. For ease of study, in this paper we assume all pairs of nodes have the same τ, and study the effect of τ with different values.

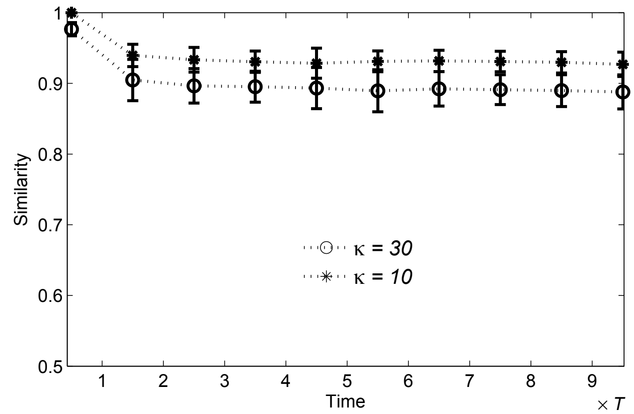

Fig. 1. Similarity between extracted communities and well-defined ones at different times for the computer-generated networks

IV. Result and Evaluation

The computational complexity of the algorithm is in general $O(n^3+n^2) \approx O(n^3)$, which scales well with the network size n. To evaluate our algorithm, we first construct a simple social mobile network with well-defined community structure that also changes over time, to see if the algorithm could extract these clusters at different times.

Consider a network with N nodes, initially divided into K communities. During each time interval t, contacts between nodes are placed independently at random with probability p_{in} for a contact to fall between nodes in the same community and p_{out} to fall between nodes in different communities. Each intra-contact (between nodes of the same community) and inter-contact (between nodes from two different communities) lasts for a period of αt and βt, respectively. Note that $0 \leq \beta < \alpha \leq 1$. The assumption is that entities from the same community are more likely to contact each other than entities from different communities, and each intra-contact will last longer than an inter-contact. To have the community structure change over time, we adjust the community composition after each time period $T = \kappa t$.

To measure the similarity of the original and detected communities of a network at different stages, we utilize *normalized mutual information (NMI)* [1] [10] measure, which is based on the confusing matrix $M(t)$. The entry $M_{ij}(t)$ denotes the number of nodes in the original community $i(t)$ appearing in the detected community $j(t)$, such that:

$$S(A,B,t) = \frac{-2\sum_{i=1}^{c_A}\sum_{j=1}^{c_B} M_{ij}\log(\frac{M_{ij}N}{M_{i.}M_{.j}})}{\sum_{i=1}^{c_A} M_{i.}\log(\frac{M_{i.}}{N}) + \sum_{j=1}^{c_B} M_{.j}\log(\frac{M_{.j}}{N})} \ ,$$

where c_A and c_B represent the number of original and detected communities in the network at time t, respectively, and

$$M_{i.}(t) = \sum_{j=1}^{c_B} M_{ij}(t) \ , \quad M_{.j}(t) = \sum_{i=1}^{c_A} M_{ij}(t) \ .$$

$S(A,B,t)$ takes its maximum value of 1 when the detected communities and the original ones are identical, and 0 if they are totally independent of each other.

Figure 1 shows the similarity between extracted communities and well-defined ones for the computer-generated networks. We sample the networks at the middle of each time period T. The results are averaged over 10 realizations of $N = 2 \times 10^2$ networks with $K = 10$ communities that are equal in size. Error bars are measured standard deviations. We set $p_{in} = 0.7$ and $p_{out} = 2 \times 10^{-4}$ for the contact probabilities in the networks. In addition, for each pair of nodes that are in contact at each time step t, α and β are chosen uniformly from the range [0.7 , 1] and [0 , 0.3], respectively. We also set the aging coefficient τ to 0.1. As we can see from the plot, the algorithm performs well on the network with communities restructured over time. The accuracies of extracted communities are above 92.6% and 88.3% of the original ones for $\kappa = 10$ and $\kappa = 30$, respectively. For either case, the similarity at the first points (i.e. $T/2$) is notably higher (up to 100% for $\kappa = 10$) since there is no community-restructuring operations before that time. The first change of community structure results in a slight dip of the accuracies but they keep stable for the subsequent time points. This can be accounted for by the small aging coefficient τ, which makes previous contacts less influential on the current community structures. We can conclude from the graph that our algorithm is more sensitive to the connectivity than the strength of each connection, and in particular, the algorithm is more sensitive to connections between communities than those within a community.

We further evaluate the algorithms on real human mobility traces. We utilize two experimental datasets gathered by the *Haggle* project at Infocom2005 conference and the *Reality Mining* project at MIT campus. In these experiments, the Bluetooth-enabled mobile devices ran software logging contacts with each other by doing Bluetooth device discovery periodically. Table I summarizes the two experiments[4]. The characteristics of these datasets, such as contact pattern and clustering behavior, have been explored in several other studies [11] [14] [27]. They cover a period from several days (*Infocom05*) to nine months (*Reality*). Since we do not have *a priori* information of communities at different times, we compare the extracted communities at certain time with those extracted at the previous step, and investigate the effect of different values of the aging coefficient τ. To study the dynamics of well-formed communities rather than trivial ones such as singletons, we only focus on certain largest communities. We utilize the classic Jaccard index as the similarity measure:

$$S_{ab} = \frac{|\mathcal{C}_a \bigcap \mathcal{C}_b|}{|\mathcal{C}_a \bigcup \mathcal{C}_b|},$$

where $\mathcal{C}_a$ represents the set of members of community a and $|\mathcal{C}_a|$ is the number of members in community a.

Figure 2 shows the result of our algorithm with different values of τ on the Infocom05 dataset. For a clearer presentation of the result in the graph, we sample the network every two hours. We can clearly observe that (for example the plot at $time = 40$ and $time = 64$) with smaller aging coefficient

[4]These datasets are available from CRAWDAD at Dartmouth http://crawdad.cs.dartmouth.edu/index.php.

Experimental dataset	Infocom05	Reality
Device	iMote	Phone
Duration (days)	3	246
Granularity (seconds)	120	300
No. of experimental devices	41	97
No. of internal contacts	22,459	54,667
Average no. of contacts/pair/day	4.6	0.024

TABLE I
CHARACTERISTICS OF THE TWO MOBILITY TRACES DATASETS

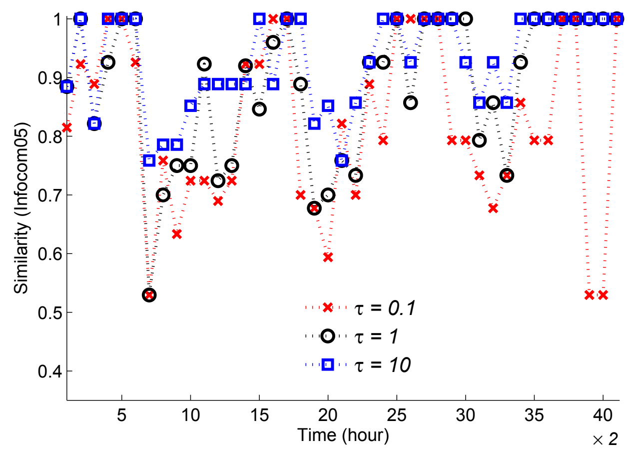

Fig. 2. Similarity between communities extracted at the current and the previous step on Infocom05 data

τ, the detected communities at current time appear less similar to those at previous time. This conveys that the algorithm with greater τ is more resilient to temporal fluctuations. The graph also shows the same periodicity of the results with different values of τ. This periodicity comes from the social patterns in the activities of participants during the conference (attending the same sessions, having lunch together, etc.) But we can see that the algorithm with smaller τ adapts faster to the structural changes in communities over time.

Figure 3 presents the result of our algorithm on the Reality dataset. We can see from Table I that this network is quite sparse (since many participants switched off their Bluetooth transceivers during the experiment), hence we plot the result with a sampling every four weeks. As in Figure 2, we can see there are more variabilities with the result of smaller aging coefficient τ (if we draw a linear regression line for each curve, apparently, the square line is at the top and the asterisk line is at the bottom). In addition, we also observe the same trend (ascending) of all the resulting curves with different values of τ.

V. Conclusion and Future Work

Motivated by message forwarding in opportunistic networks, in this paper we develop a community detection algorithm that takes into account the aging and weight of contacts between mobile entities in opportunistic networks. The evaluation result shows that our proposed algorithm can find the communities and detect the change in their structures over time. This centralized algorithm has computational complexity $O(n^3)$ which scales well with the network size n. However, in reality, a mobile device is unable to have a global view of

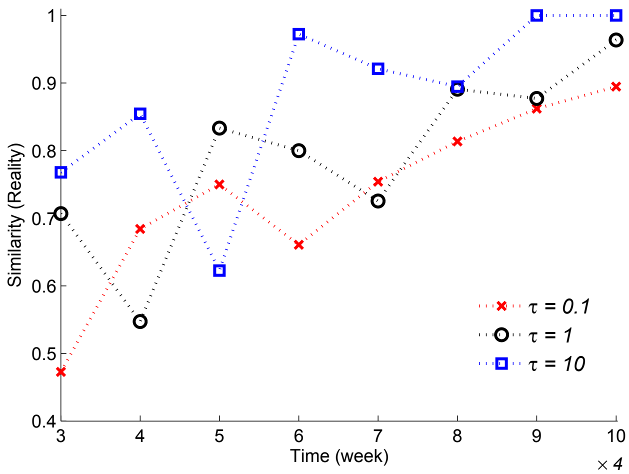

Fig. 3. Similarity between communities extracted at the current and the previous step on Reality data

the network and only has limited battery and computational resources. In the future, we would like to develop a distributed version of the algorithm and with fewer resource requirements. In addition, it would be interesting to see how the dynamic communities affect opportunistic communications over time.

ACKNOWLEDGEMENT

This research is supported in part by the University of Hong Kong Strategic Research Theme on Information Technology.

REFERENCES

[1] L. Ana and A. Jain. Robust data clustering. *Computer Vision and Pattern Recognition*, 2:II–128–133, June 2003.

[2] A. Arenas et al. Synchronization reveals topological scales in complex networks. *Physical Review Letters*, 96(11), March 2006.

[3] L. Backstrom et al. Group formation in large social networks: membership, growth, and evolution. In *Proc. of the Twelfth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 44–54. ACM, 2006.

[4] T. Y. Berger-Wolf and J. Saia. A framework for analysis of dynamic social networks. In *Proc. of the Twelfth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 523–528. ACM, 2006.

[5] D. Chakrabarti, R. Kumar, and A. Tomkins. Evolutionary clustering. In *Proc. of the Twelfth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 554–560. ACM, 2006.

[6] A. K. Chandra et al. The electrical resistance of a graph captures its commute and cover times. *Computational Complexity*, 6(4):312–340, December 1996.

[7] Y. Chi et al. Structural and temporal analysis of the blogosphere through community factorization. In *Proc. of the Thirteenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 163–172. ACM, 2007.

[8] A. Clauset. Finding local community structure in networks. *Physical Review E*, 72:026132, 2005.

[9] E. Daly and M. Haahr. Social network analysis for routing in disconnected delay-tolerant manets. In *The ACM International Symposium on Mobile Ad Hoc Networking and Computing*, Montréal, QC, Canada, 2007. ACM.

[10] L. Danon et al. Comparing community structure identification. *Journal of Statistical Mechanics: Theory and Experiment*, September 2005.

[11] N. Eagle and A. Pentland. Reality mining: sensing complex social systems. *Personal and Ubiquitous Computing*, 10(4):255–268, May 2006.

[12] G. Flake, S. Lawrence, C. Giles, and F. Coetzee. Self-organization and identification of web communities. *IEEE Computer*, 35(3):66–70, Mar 2002.

[13] M. S. Granovetter. The strength of weak ties. *The American Journal of Sociology*, 78(6):1360–1380, 1973.

[14] P. Hui, J. Crowcroft, and E. Yoneki. Bubble rap: social-based forwarding in delay tolerant networks. In *The ACM International Symposium on Mobile Ad Hoc Networking and Computing*, Hong Kong, China, 2008. ACM.

[15] P. Hui, E. Yoneki, S.-Y. Chan, and J. Crowcroft. Distributed community detection in delay tolerant networks. In *The ACM International Workshop on Mobility in the Evolving Internet Architecture*, Kyoto, Japan, 2007.

[16] A. K. Jain, M. N. Murty, and P. J. Flynn. Data clustering: a review. *ACM Computing Surveys*, 31(3):264–323, 1999.

[17] S. Jain, K. Fall, and R. Patra. Routing in a delay tolerant network. In *Proc. of the 2004 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications (SIGCOMM)*, pages 145–158. ACM, 2004.

[18] D. J. Klein. Graph geometry, graph metrics and wiener. *MATCH: Communications in Mathematical and in Computer Chemistry*, 35:7–27, 1997.

[19] G. Kossinets and D. J. Watts. Empirical analysis of an evolving social network. *Science*, 311:88–90, January 2006.

[20] L. Lovász. Random walks on graphs: a survey. *Combinatorics: Paul Erdős is Eighty*, 2:1–46, 1993.

[21] R. Merris. Laplacian matrices of graphs: a survey. *Linear Algebra and its Applications*, 197-198:143–176, 1994.

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

[23] M. E. J. Newman. Finding community structure in networks using the eigenvectors of matrices. *Physical Review E*, 74:036104, 2006.

[24] M. E. J. Newman. Modularity and community structure in networks. *Proceedings of the National Academy of Sciences (USA)*, 103:8577, 2006.

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

[26] G. Palla et al. Uncovering the overlapping community structure of complex networks in nature and society. *Nature*, 435(7043):814–818, 2005.

[27] A. Scherrer, P. Borgnat, E. Fleury, J.-L. Guillaume, and C. Robardet. Description and simulation of dynamic mobility networks. *Comput. Netw.*, 52(15):2842–2858, 2008.

[28] W.-K. Sung. Phylogenetic tree reconstruction: distance based. http://www.comp.nus.edu.sg/ bioinfo/, 2003, lecture notes.

[29] C. Tantipathananandh, T. Berger-Wolf, and D. Kempe. A framework for community identification in dynamic social networks. In *Proc. of the Thirteenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 717–726. ACM, 2007.

[30] P. Uetz et al. A comprehensive analysis of proteincprotein interactions in saccharomyces cerevisiae. *Nature*, 403:623–627, February 2000.

[31] F. Vega-Redondo. *Complex social networks*. Econometric Society Monographs. Cambridge University Press, January 2007.

[32] W. J. Xiao and I. Gutman. Resistance distance and laplacian spectrum. *Theoretical Chemistry Accounts: Theory, Computation, and Modeling*, 110:284–289, 2003.

[33] K. Xu, P. Hui, V. O.-K. Li, J. Crowcroft, V. Latora, and P. Lio. Impact of altruism on opportunistic communications. In *The First International Conference on Ubiquitous and Future Networks*, Hong Kong, China, June 2009. IEEE.