

Real World Routing Using Virtual World Information

Pan Hui
Deutsche Telekom Laboratories

Nishanth Sastry
University of Cambridge

Abstract—In this paper, we propose to leverage social graphs from Online Social Networks (OSN) to improve the forwarding efficiency of mobile networks, more particularly Delay Tolerant Networks (DTN). We extract community structures from three popular OSNs, Flickr, LiveJournal, and YouTube, and quantify the clustering features of each network at different levels of hierarchical resolution. We then show how community information can be used for forwarding using hints small enough to store on a mobile device. We also provide a first comparison study of the topological community structures for different types of OSNs with millions of users.

I. INTRODUCTION

The Pocket Switched Network (PSN) paradigm was recently proposed [6] as a means of ferrying data using human social contacts. At its core is the idea that as the storage capacities of mobile devices increase, and support for bluetooth and other short-range data transfer protocols becomes more prevalent, we could use these devices to construct data paths in a store-carry-forward fashion: Various intermediate nodes *store* the data on behalf of the sender and *carry* it to another contact opportunity where they *forward* the data to the destination or another node that can take the data closer to the destination.

Thus, the PSN routes data over humans who are close to each other in real space. We call the graph induced by links formed this way as the *proximity graph*. The unpredictability of human contacts [19] makes stable routing difficult in the proximity graph.

On a small scale, forwarding using community information has been shown to be more effective than other methods [7]. However, inferring community information from the proximity graph on a large scale such as a city is a challenging problem: The proximity graph evolves over time. Observing this over a random time window in the recent past may not reveal all intra-community links. Furthermore, nodes could have chance encounters with strangers, and community identification requires separating these from actual community relationships.

In this work, we show that the *social graph* on online social networks (OSNs) can be used to efficiently infer routes on a large scale. OSNs such as Facebook, Orkut, Flickr, and LiveJournal are attractive because of their

large user base and *explicitly declared, stable* friendship links, which are oriented along common interests, affiliations, hobbies, political stands etc. This, together with their explosive membership growth [11], [8] creates the possibility of a more complete and accurate community identification than can be computed from observing the proximity graph alone.

Note that the social graph captures virtual world friendships, whereas in the proximity graph, we are interested in routing over encounters which happen in the real-world. It has been shown that there is a strong correlation between properties such as paths and links on the social graph and the proximity graph [15]. Thus, the basic strategy we suggest is to compute geographically localized subgraphs of the social graph. We then compute routing hints on these subsets and apply this to the proximity graph.

The principal challenge is that mobile devices have relatively small storage. Thus the routing information stored on the device has to be succinct. Secondly, because of privacy considerations, it is not feasible to store social graph information beyond direct friendship links of the owner of the mobile device.

We address these issues by computing a community digest of the OSN social graph. The digest consists of node memberships in communities at several levels of hierarchy. Community detection is a compute-intensive process, and for the large OSN data sets we examine, the graphs are so massive that most of the common community detection algorithms [16] [3] can not handle them. We use a fast community detection algorithm based on modularity optimization [1].

Back-of-the-envelope calculations show that in the social graphs we examine, the community digests are small enough to be stored on current generation mobile phones. Furthermore, the digest conceals sensitive friendship information about strangers in the OSN, while allowing these strangers to be used as intermediate nodes for store-carry-forwarding.

Orthogonal to the routing concerns of the paper, we perform an analysis of the communities we find in the large social graphs examined. We report several

macroscopic features and sizes of detected community patterns in each of our data sets.

Our contributions may be summarized as follows:

- Based on results in [15], we suggest using the *social* graph of declared online friendships to compute routes on the *proximity* graph of human contacts in real space-time.
- We suggest that routes can be computed using community information and demonstrate that a fast modularity-based community detection scheme will scale for large networks of a similar order of magnitude as current OSNs.
- We calculate that routing information based on communities can be easily stored on small mobile devices.
- We give an analysis of communities found in 3 large-scale OSNs.

The rest of the paper is organised in the following order: leveraging the social graph for forwarding (Section II), the concept of modularity optimization and the fast community detection algorithm (Section III), a brief description of the three online social network datasets (Section IV), statistical properties of the hierarchical community structures of the social graphs (Section V), discussions (Section VII), and conclusions (Section VIII).

II. SOCIAL-BASED FORWARDING

Routing or forwarding issues in DTN and Mobile Ad Hoc Networks (MANET) are important research topics for researchers. Many MANET and some DTN routing algorithms [9] [10] accomplish forwarding by building and updating routing tables whenever mobility occurs. This approach is considered to be cost ineffective for a human mobile network, since human mobility is often unpredictable, and topology changes can be rapid. For instance, Fig 1 shows the contact occurrence distributions in two small-scale proximity graphs created from subsets of data gathered in the MIT Reality mining [4] and UCSD wireless topology discovery [20] projects. It shows that a random pair of nodes are likely to be connected only very rarely. There is a high probability that an edge will occur fewer than 10 times in the trace, and cannot be easily be learned by distributed route computation algorithms. Yet, as shown in [19], these rare edges are important for data delivery.

Rather than exchange much control traffic to create unreliable routing structures, we propose to use social information to choose the next-hop relays, which are less volatile than mobility [7]. It was shown using real human mobility traces that by leveraging social information such as centrality [5] and community [17], the proposed forwarding algorithm, *Bubble*, can significantly improve forwarding efficiency by increasing the fraction of delivered messages at a lower cost.

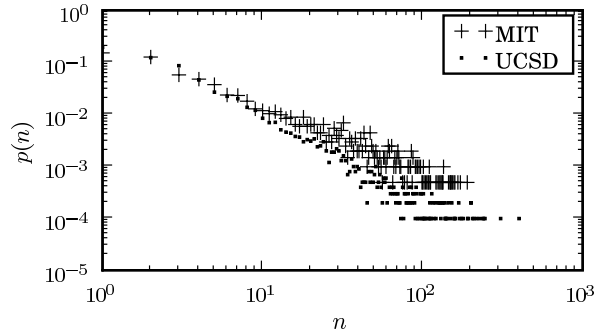


Fig. 1. Contact occurrence distribution (log-log): A contact happens n times with probability $p(n)$

The proximity graphs obtained by mobility experiments are usually small in size due to the limitations (e.g. cost and number of participants) of the experiments. Even moderately large experiments involving a hundred or more nodes (such as the MIT and UCSD data sets discussed above) demonstrate the difficulties of learning appropriate routes directly from the proximity graph.

In this work, we ask whether we can leverage the social graphs from the OSN to serve similar purposes, or at least to bootstrap the process. Considering that a typical OSN topological graph contains tens of millions of nodes (although still tiny compared to billions of mobile users on Earth), it is a valuable resource.

A. Inferring proximity from social graph

Let us denote a social graph from one of the OSN as $G = (V, E)$, where V is the set containing all the nodes in the graph and E is the set of all edges which connect nodes pair (u, v) in the graph. Here a node $v \in V$ is a user on the OSN, and $e \in E$ defines the relationship between two users (e.g. friendship). Considering that G consists of users scattered all over the world, some users connected by an edge may be located in different parts of the world and not useful for city-wide mobile computing. In this case, we can first remove all the edges with the geographical locations not in the same target city. By this process, we can create subgraphs $\{G_i = (V_i, E_i) \mid V_i \subset V, E_i \subset E\}$ for each city i . Our datasets do not have geographical information, so we cannot evaluate the sizes of the geographical subgraphs in this paper. However, Wilson *et al.* have crawled Facebook regional networks, and found that there are more than 2 million Facebook users in London [21]. This number is large enough for bootstrapping many mobile computing applications in a large city like London.

In human society, we can build up hierarchical trees according to the closeness of the relationship between the nodes. For example, we can structure it in such a way that the leaf nodes are the individuals, the first

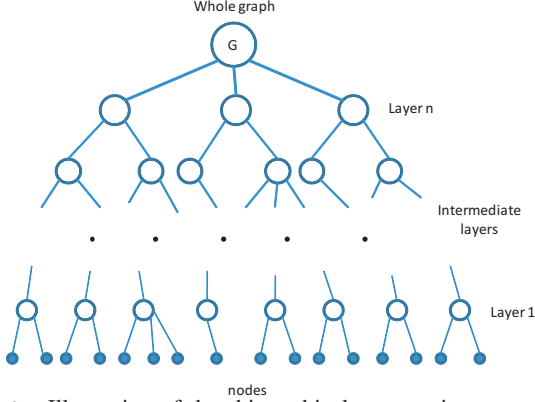


Fig. 2. Illustration of the shierarchical community structures.

layer consists of family members which represent the closest links, followed by relatives in the second layer, and the top layer will be all the members in the society. The nodes in each layer represent the members in each community at that layer. The closeness of the relationship decreases when we go up the tree. There are many other ways of constructing the hierarchies, such as using affiliation instead of kinship. The lower the layer two nodes belonging to the same community, the closer their relationship and the higher chance that they are good relays of messages for each other.

By similar means, we can extract a hierarchical structure, H_s from graph G based on the cohesiveness of the connection of the nodes on the graph. Figure 2 illustrates the process of building up the routing structure from graph G . The leaf nodes are the individual nodes on the original graph. Layer 1 is the first layer of the hierarchy, with each node representing the most densely connected community, and vice versa up the tree.

B. Forwarding using communities

We can then define a probability of forwarding in each community layer as P_l , where l is the layer number. Considering an encounter scenario, where device v_i meets device v_j , and v_i has a message m for device v_k . If v_j and v_k are in the same community at layer l , the probability that v_j is a good relay for v_k is proportional to the probability P_l . For example, we can let $P_1 = 1$. In other words, if the two nodes are in the same community at layer 1, they are good relays for each other. We can then define the probability of choosing v_j as relay for v_k , P_{jk}^l as

$$P_{jk}^l = \frac{|C_k^1|}{|C_{jk}^l|} * P_1 \quad (1)$$

where, v_j is the encountered node, $|C_k^1|$ is the size of the community at layer 1 to which v_k belongs, and $|C_{jk}^l|$ is the size of the community to which both v_k and v_j belong and l is the lowest layer at which v_j and v_k are in the same community. Because of the strict hierarchy, $|C_k^1| \leq |C_{jk}^l|$, so $0 \leq P_{jk}^l \leq 1$ is a valid probability.

1) *Discussion:* The above proposal is based on a probabilistic community based measure of how far two nodes are in the social graph. Of course, there are many other ways of measuring the social distance of two nodes, for example, the shortest path distance between them on the graph. It is difficult for us to assess which definition is better without large scale empirical mobile data.

The advantage of our approach is its simplicity and that each node only needs to know the information of its own communities. For example in the above case, v_j only needs to know the size of its layer 1 community, and whether v_k is in any one of its communities. Succinctness is important since the information will be stored on a mobile device.

Furthermore, as we will find later, community structures in the real OSNs we examine have no more than 6–7 levels of hierarchy. Thus, going up the hierarchy and coming back down to the (destination) leaf node will not cause too many extra hops in the route.

III. FAST COMMUNITY DETECTION

In order to know how the real OSN social graph can be fitted to the requirements of mobile devices, we look at three popular OSNs and see what kind of hierarchical structures we can extract from these social graphs. In order to do this, we use a community detection algorithm to cluster the datasets. The algorithm we used in this paper is a fast community detection algorithm based on modularity optimization [1]. Modularity is defined as the difference between this fraction and the fraction of the edges that would be expected to fall within the communities if the edges were assigned randomly but keeping the degrees of the vertices unchanged.

The fast algorithm [1] is divided into two phases and repeated iteratively. At the beginning, each node is assigned to a different community, then, for each node i and its neighbour node j , we evaluate the gain of modularity by removing i from its community and placing into the community of node j . We place node i into the neighbour community for which the gain is the maximum. This process is repeated until no more increase in modularity can be achieved by replacing nodes. The second phase of the algorithm involves building a new network whose nodes are the communities found during the first phase. The weights of the links between nodes in the new network are now the sum of the weight of the links between nodes in the corresponding two communities, and the links between nodes of the same community become a self-loop on the new node. We call the communities detected with a run consisting of these two phases the communities on that particular hierarchical level. The modularity optimization process from the first phase is repeated on the new network.

These two phases are repeated iteratively until the global modularity maximum is achieved.

IV. OSN DATASETS

The OSNs we study in this paper includes Flickr, LiveJournal, and YouTube. The data were collected by crawling in late 2006 and 2007 [14]. Of course, these networks have evolved since they were crawled, as would be expected for any real social network. Studying a human social network at a certain point in time can still give us some general insight and knowledge into the human anthropology and sociology. More importantly, we believe that the community characteristics of the data sets captured would be representative of the current states of these OSNs.

Details of the data sets are described below and their high-level statistics are summarised in Table I.

OSN	Flickr	LiveJournal	YouTube
Number of users	1,846,198	5,284,457	1,157,827
Fraction crawled (estimate)	26.9%	95.4%	unknown
Number of links	22,613,981	77,402,652	4,945,382
Avg. friends per user	12.24	16.97	4.29
Fraction of links symmetric	62.0%	73.5%	79.1%
Number of groups	103,648	7,489,073	30,087
Mean group membership per user	4.62	21.25	0.25

TABLE I
HIGH-LEVEL STATISTICS OF THE CRAWLED OSN DATASET.

Flickr (www.flickr.com) is an image and video hosting website, web services suite, and online community platform. It was launched in February 2004, and as of November 2008, it claims to host more than 3 billion photos. The data set we used in this paper contains over 1.8 million users and 22 million links.

LiveJournal (www.livejournal.com) is an online social network for bloggers. Users can share their blog, journal, and diary. LiveJournal was started in March 1999. The LiveJournal data in this paper was obtained from a crawl from December 9-11, 2006, with the APIs provided by the website.

YouTube (www.youtube.com) is a video sharing website where users can upload, view, and share video, which includes a social network behind it. The YouTube data set we studied in this paper was obtained on January 15th, 2007, it consists of over 1.1 million users and 4.9 million links, and is believed to cover a large fraction of the whole network.

V. HIERARCHICAL STRUCTURES

We apply the hierarchical modularity optimization algorithm from Section III on the three data sets from Section IV. The algorithm can handle the whole crawled graphs for Flickr, LiveJournal, and Youtube on a computer with 2GB memory. Previous work has studied

OSN	Flickr	LiveJournal	YouTube
No. Nodes	1,861,223	5,284,458	1,157,828
Number of links	22,613,981	77,402,652	4,945,382
Final Stage			
No. Community	199,565	93,013	31,837
Mean Community Size	9.32	56.8	36.38
Modularity	0.6548	0.7370	0.6948
Intermediate Stage			
No. Community	324,104	243,885	120,022
Mean Community Size	5.74	21.66	9.65
Modularity	0.6381	0.7170	0.6477

TABLE II
DESCRIPTIVE STATISTICS OF DETECTED COMMUNITY AT THE FINAL STAGE AND AT AN INTERMEDIATE STAGE.

proximity communities of several OSNs [18], but the scale is only up to thousands of nodes. Here we study networks of millions of nodes and we believe this can give a better overall picture about the clustering and community properties of these social networks.

Figure 3 shows the community size distribution at the final hierarchical level with the global modularity maxima, and Table II (top half) summarises the descriptive statistics of the communities. We can see that the modularity values are all above 0.65, and according to Newman *et al.*[16], the detected communities are considered to be good partitions if the modularity is above 0.4.

We further look at the community size distribution at lower hierarchical levels (i.e. intermediate detection stages). There are 6, 6, and 7 levels of hierarchies respectively in the Flickr, LiveJournal, and Youtube networks. Figure 4 shows again the size distribution of the detected community at layer 1, and the bottom of Table II shows the descriptive statistics. The modularity values at this intermediate stage are not much different from the maximum modularities, and are still above 0.6, indicative of good clustering.

In the LiveJournal and Youtube networks, there is a significant difference in the number of communities at the final and intermediate stages. Flickr is the exception and shows less than a factor of 2 increase in the the number of communities. Since the number of nodes in each network is constant, the mean community size also drops significantly in the intermediate stage, as compared to the final stage.

Observe that for all the data sets, almost all communities sizes are below 100 at both the final and intermediate stages, although there are also a few extremely large communities of hundreds of thousands of nodes. We make use of this highly dense nature of social communities to create succinct community digests suitable for mobile devices.

VI. STORING COMMUNITIES ON MOBILE DEVICES

The routing hints we propose in Section II-B are the community memberships of the person at different levels

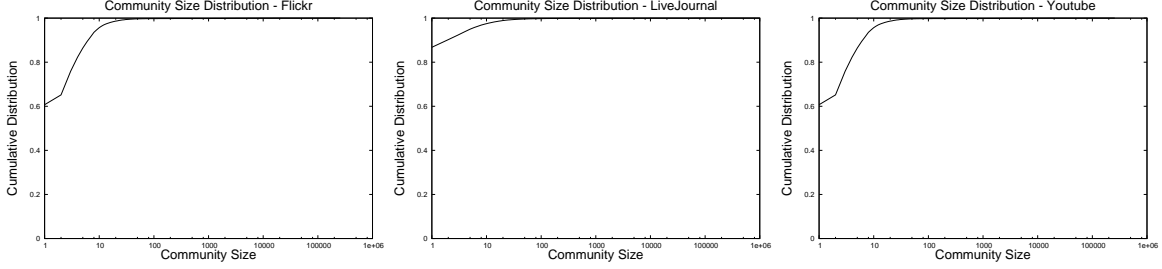


Fig. 3. Size distribution of detected communities at the final hierarchical level.

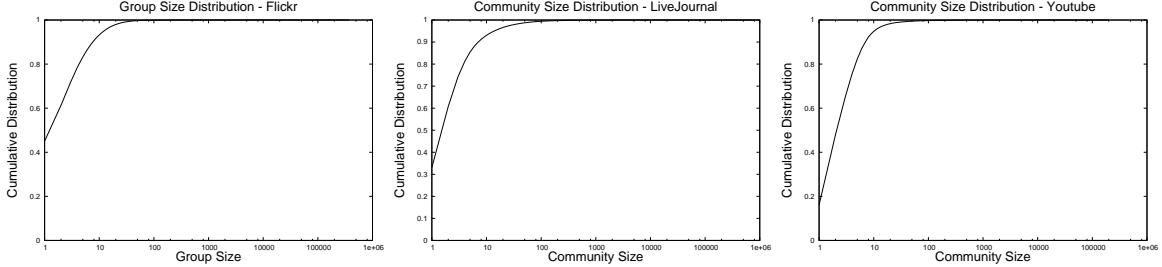


Fig. 4. Size distribution of the detected communities at layer 1.

of hierarchy. Below, we examine whether it is practical to store this information on a mobile device, given the typical communities we find in the previous section.

A. Succinctness

The mean level-1 community size for the LiveJournal network is 21, which can easily fit into the storage of a mobile device. To store the Bluetooth IDs (6 bytes) of each node in the community on the mobile, we need 126 bytes, which is nothing compared to the size of modern flash memory, but of course each user can have multiple Bluetooth IDs. Even for an extremely large community of 100,000 nodes, we only need 0.6M bytes of storage, again within the reach of a modern mobile phone, which can have a few gigabytes of flash memory.

If every user can afford ~ 10 MBytes, we can store nearly two million devices's Bluetooth ID. Based on this estimation, it appears that modern mobile phones leave room for more sophisticated algorithms which use more detailed information than just Bluetooth IDs.

B. Privacy

Even if the community is small enough to store on a mobile device, it may not be preferable to store this information on a low-security device such as a mobile phone, due to privacy concerns. A solution to this is to store a one-way hash of the Bluetooth IDs. For instance, with SHA-1, we will only need 160 bits per ID, leading to further compression.

For larger communities (e.g. at higher levels of the hierarchy), a bloom filter can be used to test membership of a given Bluetooth ID in one of the communities. The bloom filter has the advantage of requiring a fixed amount of storage regardless of community size.

The trade-off is the possibility of false positives with the Bloom Filter. The false-positive is given by $(1 - e^{-kn/m})^k$, where m is the length of the Bloom filter, n is the number of element inserted into the filter, and k is the number of hash functions. The probability of false positives decreases as m increases, and increases as n increases. By adjusting these parameters as well as the probability of forwarding p_l , we can achieve the least possible false positive rate given the storage available on each device.

We show how the lookup works with an example: Suppose each device maintains s bloom filters for each of its communities at s different levels of hierarchy. Using this mechanism, consider the scenario from Section II-B where v_i meets v_j , and device v_i has a message m for v_k . For illustration, assume v_j has $s = 3$ bloom filters for its three layers of community information, F_{C_1} , F_{C_2} , and F_{C_3} , where F_{C_1} is the bloom filter for the community at lowest layer, layer 1, and F_{C_3} is the bloom filter for the community at layer 3. v_j will check whether v_k is in its filters from F_{C_1} to F_{C_3} . Suppose v_j is found in F_{C_2} . The data is forwarded to v_k with probability P_{jk}^2 computed as described in Section II-B. To compute P_{jk}^l , v_j needs to additionally store the layer-1 community size $|C_k^1|$ of each node k in its community at any layer. This is not in itself a huge burden because community sizes, especially at layer 1, are small, requiring no more than 3–4 bits per node.

Note that $P_{jk}^1 \geq P_{jk}^2 \geq P_{jk}^3$. Hence, communities up the hierarchy are preferred with lower probability. Thus the probabilities make it likely for data to move up the community hierarchy when representatives of v_k from lower layers of the hierarchy are not encountered.

Once data reaches a community of v_k at some level, it will preferentially move down the community hierarchy until it reaches the layer 1 representatives, who will meet v_k and deliver data.

The process of probabilistically forwarding data is repeated until the message is finally delivered to node v_k . By using this method, the system can also be scalable even with billions of nodes in the system since each bloom filter will occupy a fixed length.

VII. DISCUSSION

A. Improving Forwarding Efficiency

The Bubble algorithm uses both community and centrality information to improve forwarding efficiency [7]. It is possible to calculate the centrality value of each node on the social graphs using egocentric approximation [12], and merge together with the hierarchical community information to achieve the same performance as Bubble. Our contribution in this paper is not to propose and analyse yet another social-based forwarding algorithm, but to analyse how can we leverage information from OSN social graphs for more stable routing in large-scale PSNs.

We do not claim that the algorithm used in this paper to extract hierarchical information is optimal. It is intended merely as a proof-of-concept. There are other methods in the literature that can be used, for example the algorithm by Clauset *et al.* for extracting hierarchical structure and predicting of missing links on networks [2].

B. Bootstrapping media sharing

The algorithm we have outlined has looked only at delivery efficiency in forwarding and routing. However, given the specialised nature of the different networks that exist, we envision a complete platform for media sharing: The users on the LiveJournal are interesting in blogging, the users on YouTube are interested in watching video, and the users on Flickr consume photos. If we merge the data from all the three networks together, we can create a powerful mobile content sharing system. We can push suitable content to the right users according to their positions and interests in these OSNs. This can be useful during the bootstrapping step for mobile media sharing system such as [13].

VIII. CONCLUSIONS AND FUTURE WORK

We study the hierarchical community structures of three popular online social networks, and propose a scheme to use this multi-scale social graphs for social-based forwarding in delay tolerant networks. Unfortunately, we cannot quantitatively evaluate the improvements in forwarding efficiency, because there is no way to bootstrap human mobility from static social graphs at the moment, at least by the research community. In the future, we are planning to conduct large scale human mobility data collection (www.amillionpeople.net), which are important while lacking at the moment. This

kind of dataset can help to link the dynamic of mobility to the static social graphs and provide insights for better modeling of human mobility.

REFERENCES

- [1] V. D. Blondel, J.-L. Guillaume, R. Lambiotte, and E. Lefebvre. Fast unfolding of communities in large networks. *J.STAT.MECH.*, page P10008, 2008.
- [2] A. Clauset, C. Moore, and M. E. J. Newman. Hierarchical structure and the prediction of missing links in networks. *Nature*, 453:98, 2008.
- [3] L. Danon, J. Duch, A. Diaz-Guilera, and A. Arenas. Comparing community structure identification, 2005.
- [4] N. Eagle and A. S. Pentland. CRAWDAD data set mit/reality (v. 2005-07-01). Downloaded from <http://crawdad.cs.dartmouth.edu/mit/reality>, July 2005.
- [5] L. C. Freeman. A set of measures of centrality based on betweenness. *Sociometry*, 40:35–41, 1977.
- [6] P. Hui, A. Chaintreau, J. Scott, R. Gass, J. Crowcroft, and C. Diot. Pocket switched networks and the consequences of human mobility in conference environments. In *Proceedings of ACM SIGCOMM first workshop on delay tolerant networking and related topics*, 2005.
- [7] P. Hui, J. Crowcroft, and E. Yoneki. Bubble rap: Social-based forwarding in delay tolerant networks. In *MobiHoc '09: Proceedings of the 9th ACM international symposium on Mobile ad hoc networking & computing*, May 2008.
- [8] N. Johnson. Facebook's rapid growth continues with 200 million members, Apr 2009. <http://blog.searchenginewatch.com/090408-132015>.
- [9] E. P. C. Jones, L. Li, and P. A. S. Ward. Practical routing in delay-tolerant networks. In *Proc. WDTN*, 2005.
- [10] A. Lindgren, A. Doria, and O. Schelen. Probabilistic routing in intermittently connected networks. In *Proc. SAPIR*, 2004.
- [11] R. MacManus. Digg overtakes facebook with 1400% growth, 22.6 million uniques, June 2007. http://www.readwriteweb.com/archives/is_digg_really_more_popular_than_facebook.php.
- [12] P. V. Marsden. Egocentric and sociocentric measures of network centrality. *Social Networks*, 24(4):407–422, October 2002.
- [13] L. McNamara, C. Mascolo, and L. Capra. Media sharing based on colocation prediction in urban transport. In *MobiCom '08: Proceedings of the 14th ACM international conference on Mobile computing and networking*, pages 58–69, New York, NY, USA, 2008. ACM.
- [14] A. Mislove, M. Marcon, K. P. Gummadi, P. Druschel, and B. Bhattacharjee. Measurement and analysis of online social networks. In *Proceedings of the 5th ACM/USENIX Internet Measurement Conference (IMC'07)*, October 2007.
- [15] A. Mitbaa, A. Chaintreau, J. LeBrun, E. Oliver, A.-K. Pietilainen, and C. Diot. Are you moved by your social network application? In *WOSN '08: Proceedings of the first workshop on Online social networks*, 2008.
- [16] M. E. J. Newman and M. Girvan. Finding and evaluating community structure in networks. *Physical Review E*, 69, February 2004.
- [17] G. Palla, I. Derenyi, I. Farkas, and T. Vicsek. Uncovering the overlapping community structure of complex networks in nature and society. *Nature*, 435(7043):814–818, 2005.
- [18] B. Saha and L. Getoor. Group proximity measure for recommending groups in online social networks. In *2nd ACM SIGKDD Workshop on Social Network Mining and Analysis*, 2008.
- [19] N. Sastry, K. Sollins, and J. Crowcroft. Delivery properties of human social networks. In *Proc. IEEE INFOCOM Miniconference*, 2009.
- [20] "UCSD". Wireless topology discovery project. <http://sysnet.ucsd.edu/wtd/wtd.html>, 2004.
- [21] C. Wilson, B. Boe, A. Sala, K. P. Puttaswamy, and B. Y. Zhao. User interactions in social networks and their implications. In *EuroSys '09: Proceedings of the fourth ACM european conference on Computer systems*, pages 205–218, New York, NY, USA, 2009. ACM.