Social Network Based File Sharing System in Mobile Peer-to-peer Networks

Haibo Zhang and Haiying Shen
Department of Computer Science and Computer Engineering
University of Arkansas, Fayetteville
Fayetteville, Arkansas 72701
Email: {hxz009, hshen}@uark.edu

***Abstract*—Peer-to-peer file sharing applications are increasingly popular and enable users to share various files with high scalability. However, in intermittently connected mobile network environment, existing file sharing systems are hindered by lack of connectivity and constant network topology change. We propose a file sharing system that takes advantage of node mobility and builds communities based on social network. Instead of being hindered by the mobility of the network, we utilize the node mobility to facilitate communications and provide message routing and file retrieval services.**

I. Introduction

In the past few years, the improvement of microelectronics results in dramatic increase of the availability of personal mobile devices, such as laptops, powerful PDAs and smart mobile phones. With the convenient devices, they can share the user-generated contents and interested resources through computer networks. They can interact with the users and share the files via Internet connections. However, there are occasions when the Internet infrastructures are unavailable, so they should share the files through mobile connections. The environment can be regarded as a mobile peer-to-peer (P2P) network in which nodes are constantly changing. In mobile P2P network, the peers are intermittently connected, which makes traditional approaches of file searching and file sharing between two distant peers almost infeasible. It is a challenging problem to build the file-sharing systems in such mobile P2P network environment with frequent disconnections of nodes and dynamic topology.

In this paper, we design a social network based file sharing system in mobile P2P networks. It leads to low network overhead and offers satisfactory transmission success rate. Our contribution in this paper is two-fold. First, we cluster the peers in the mobile P2P network to different communities based on their interests and enable users to retrieve files in their communities, which reduces traffic cost without reducing file searching success rate. Second, we exploit different mobility of the nodes for file searching and inter-community communication. Peers in the mobile P2P networks are assigned different tasks based on their characteristics of stability or mobility, so that peers can better utilize their resources and make more contributions to file searching and file retrieving. The social network based community, routing algorithm and exploitation of stability and mobility bring about significantly reduced network traffic and enhanced file retrieval success rate.

The remainder of this paper is organized as follows. Section 2 provides an overview of P2P file sharing systems and social networks. Section 3 presents the design of node functions and file retrieving processes. In section 4 the performance of our proposed system is evaluated in the simulation. The last section gives the concluding remarks and future work.

II. Related Work

P2P systems provide a distributed file sharing environment without the assistance and surveillance of the central servers. More and more users are attracted to use the file sharing applications over the Internet, and intense research work has been conducted in this area. Napster [1] has a central server to keep an index of files to assist peers to search their interested files. Gnutella [2] employs flooding for fast data retrieving at the cost of heavy communication traffic in network. CAN [3], Chord [4], Pastry implemented distributed hash table to hash file names or peers to keys that map the resources to corresponding peers.

A number of file sharing systems have been proposed in mobile P2P networks. Passive Distributed Indexing (PDI) [5] is a general-purpose distributed search service and it provides a simple approach for file searching in mobile network. Klemm *et al.* [6] proposed a special-purpose on-demand searching and file transferring algorithm based on an application layer overlay network. It transparently aggregates the query results and eliminates redundant routing paths and it enables efficient file transfers. Nevertheless, both of them use simple flooding strategy for file searching in local region and can only retrieve the list of files that are available in local part of the network. The index of files that are stored in nodes in other partitions cannot be retrieved even if nodes have infrequent contact with them in the past or may contact them in the near future. Ding *et al.* [7] evaluated and compared five routing basic approaches for P2P file sharing over mobile ad hoc networks. Anna Hayes *et al.* [8] extended Gnutella for mobile environment and proposed to use a set of keywords to represent the user's interest.

Since the device carriers in the network are human beings, whose movements are strongly affected by their social activities, recurring patterns can be observed from their movements and behaviors [9] [10][11]. Previous research has applied social network in data forwarding and data retrieving through a shorter routing path. Small-world phenomenon has been observed in social network from the observation that the individuals are often likely connected through a short chain of acquaintances. The phenomenon was confirmed in the small world experiment [12] conducted by Milgram and his co-workers. The result of the experiment showed that the average number of intermediate forwarders in a successful chain was found to be approximately 6, known as the "Six degrees of separation" principle. Recent studies have revealed that the phenomenon is found in mobile P2P networks with intermittent connections where humans carries the mobile devices [13], and Internet applications such as YouTube where human users are intensively involved [14]. CAR [15] and HiBOp [16] use the context information based on social network models [17] for context-aware routing in intermittently connected mobile P2P network. Conti *et al.* [15] proposed a mobile file-sharing framework that uses the same mechanism to route data and use the extra storage capacity to cache data for other potentially interested users. Boldrini *et al.* [18] further included other factors such as resource limitation and access probability.

III. The Design of File Sharing System

A. Node Functions

In our file sharing system, we assign a number of responsibilities or functions for the nodes in the mobile P2P network to facilitate file sharing.

The underlying function file caching & index management module provides persistent storage for users' interested files and cached files for potential access (for the local user or the user's neighbors). If mobile nodes find files in other communities that may be interested in by nodes from the original community, and there is enough storage, it is replicated to mobile nodes and finally carried back to stable nodes and store in them. If the node is interested in the file or the neighbors may potentially be interested in it, it will be cached in local repository. In addition, the nodes also recommend to others the files that its neighbors are potentially interested in. This is an intelligent part of the system that receives the files even before the user sends out the request. Index server in each community keeps a complete local index for all the files inside the community and a partial global index for popular files in other communities.

Keyword extraction can infer the best keywords to describe the users' interests from the local file list based on the word importance. The keywords extracted from the meta data of the files are divided into several groups to represent different categories of interests. Each category of interest are assigned a weight based on the access frequency of the related files. Community construction is based on the groups of keywords that represent the characteristics of the nodes. Since the files contain much descriptive information, extracting keywords that best describe the interests of individual nodes is crucial. Communities are constructed in our system to help users to locate the file in their communities and prevent unnecessary query messages flooding to other communities.

Routing selection determines the path to forward the requests or the files. Although nodes in the same community have similar interests, they have different weight for each of the keywords. Each node on the routing path computes the interest similarity between the destination node and the neighbors and chooses the best one as the next relaying node.

A subscription for the file will be created on the index server chosen from the stable nodes if the file cannot be found in the local community. The index server will disseminate the requests to other communities to search the file through mobile nodes.

B. Community Construction

The keyword list provided by interest keyword extractor is exploited to explore the common interest and similarity of a group of nodes. Each interest-based community can be represented by a number of groups of keywords. A node with similar interests and frequent contacts with the members of the community can be added to the community.

We explain here the processes of interest information exchange and community construction and maintenance. When two nodes meet, they exchange their keyword collections, each of which consists of a list of keywords and corresponding percentage of file numbers in the local reservoir. After comparing the keyword collections of the two users' interests, the similarity values would be recorded and they would construct a community if the contact frequency exceeds a threshold. The similarity Sim between arbitrary interest collections of the two nodes is evaluated as the function of the percentage of files in the local repository and the word semantics distance, *i.e.*:

$$Sim(G_{ik}, G_{jl}) = \sum_{p \epsilon G_{ik}} \sum_{q \epsilon G_{jl}} \frac{w_{ik} * w_{jl}}{SD(K_{ikp}, K_{jlq})},$$

where G_{ik} denotes the k^{th} interest keyword group of the node N_i. K_{ikp} denotes p^{th} keywords of the k^{th} keyword collection on node N_i. w_{ik} is the weight for this keyword collection, i.e., the percentage of the related files in local repository. SD is a function to compute the semantic distance of two words, so that even if the words don't match exactly but are highly relevant, we can still get a high similarity score. If the two users have multiple irrelevant common interests, multiple communities will be created. The community information including community ID and community keywords is stored separately in each member. For new comers that potentially have similar interests with the community and frequent contact with the community, community matching value CM is computed to evaluate his/her membership for the community.

$$CM(N_i, C) = \sum_{j \epsilon M} \sum_{k=0}^{n_{KG}} Sim(G_{ik}, G_C) * EF(i, j),$$

where N_i denotes node i, C denote the community and EF is the encountering frequency of node i and node j. M denotes the members in the community C. G_{ik} denotes the k^{th} interest keyword group of the new comer itself.G_C denotes the interest keyword group of the community. n_{KG} is the number of keyword groups. If it exceeds a threshold, it is granted membership to the community; otherwise, the similarity value would be stored in table called node similarity table in the new comer node for future evaluation.

C. Routing Algorithm

Social network theory discloses the phenomenon that people with similar social properties tend to cluster and people in the same community have strong links to each other. Small world phenomenon [12] in social network indicates that messages can be delivered between two persons who dont know each other through intermediate forwarders within a small number of hops. In reality, people with the same interests tend to communicate more frequently with each other and share things that both of them are interested. If we exploit this social behavior model and build communities based on their various interests, it would significantly help route messages in mobile p2p network.

In the proposed system, various communities are constructed based on a group of people's common interests if they have a certain degree of contact frequency with other members of the community. Distant visitors with similar interests are excluded from the local community for convenience of membership management. The users have multiple categories of interests and therefore belong to multiple communities. Given the destination node's interest keyword list, the optimized approach that we proposed in this paper is to route messages based on the similarity of interests. We always choose the nodes that are the most similar with the destination node, instead of flooding the messages into the network.

For every neighbor the node has encountered, it receives the interest keyword list and records each of the keywords and the corresponding encountering frequency. HiBOp[16] routing algorithm is a context-aware approach, while our approach is more specific but it focuses on interests only. We further improve the algorithm to include the history of indirect encountering. The directly and indirectly encountered nodes are included in the history table by exchanging history tables when two nodes meet and the probability of the encountering is represented as a weight in exponentially decaying fashion based on the time to travel (TTV) from a node j to the local node i. The history weight H(i,j) is computed as $H(i,j) = \lambda * e^{-\theta TTV}$, where TTV means the time slots it takes to travel from the specific node to the local node. It indicates the probability change for node i to encounter the destination node j directly or indirectly and the probability decreases when TTV decreases. With the destination node ID and its interest keyword list, the node is able to estimate the probability that each neighbor might encounter the destination node directly or indirectly and choose the most possible nodes to forward the messages.

The evaluation score of encountering the destination node is computed as:

$$S(N_i, N_j) = \frac{(\sum_{k=0}^{n_{KG_i}} \sum_{j=0}^{n_{KG_j}} Sim(G_{ik}, G_{jl})) * EF(i,j) H(i,j)}{n_{KG_i} n_{KG_j}},$$

where $i \epsilon M$, N_i denotes current node i, N_j denotes destination node j, $EF(i,j)$ is the nodes' encountering frequency, G_{ik} denotes the k^{th} interest keyword group of the new comer itself, and n_{KG_i} denotes the count of keywords in destination node. For each of the keyword groups in destination node's interest keywords list, we choose the largest similarity value weighted by encountering frequency and history weight.

D. Exploitation of Stability and Mobility

In mobile P2P environment, the devices carried by people move around with them. The mobility of devices results in intermittent connectivity between them and constantly changing network topology. In this environment, the traditional routing algorithms and file-sharing strategies in mobile P2P environment become infeasible. In this paper, we take advantage of different mobility of the nodes to searching and retrieve files. We assign one of the most stable nodes in a community as the index server, which keeps an index of all the files in the community and maintains a stack of file queries that should be transferred to other communities. Several other nodes with high mobility and frequent connections to the index server are assigned as the communicator nodes, which handle inter-community communications. They download the request stack from index server and travel from the index server in home community to external communities. On the traveling path between home community and external communities, if the communicator nodes happen to find the neighbors have the files requested or files that home community members may be potentially interested in, they cache the files on local repository. The probability that some members in the home community will need to access the files in the short future can be evaluated by the similarity formula Sim.

1) Index Server Node Selection: The index server is chosen from the most stable nodes, because intuitively the stable nodes tend to move less and stay in a certain scope of area, where others may communicate with it after entering the area. We define stable nodes as the nodes that have high frequency to communicate with other community members, instead of those remain in a physical area, due to lack of GPS and infrastructures. The criteria we use to select the index server is the tightness of the relationship with other members in the community, which means that we choose the node with the most important or popular role in the community. In graph theory and network analysis, centrality is used to determine the relative importance of a vertex within the graph (e.g., how important a person is in a community). There are several widely used measures of centrality to evaluate the nodes' importance: degree centrality, betweenness, closeness [19], and eigenvector centrality.

- Degree centrality [20] is measured as the number of direct links that a node has. The node p_i's degree centrality in the network with a size of N is calculated as:
  $C_D(p_i) = \sum_{k=1}^{N} a(p_i, p_k)$,
  where $a(p_i, p_k)$ is the connectivity between node p_i and node p_k and $a(p_i, p_k) = 1$.

- Closeness centrality is measured as the reciprocal of the mean shortest path length linking nodes p_i and p_k among all the possible approaches. A node's closeness centrality is calculated as:
$C_C(P_i) = \frac{N-1}{\sum\limits_{k-1}^{N} d(p_i,p_k)}$,
where N is the size of the network and $d(p_i, p_k)$ is the geodesic distance between nodes p_i and p_k.
- Betweenness centrality is measured as the number of path that a node indirectly links 2 other nodes. In the network with N nodes, betweenness centrality is calculated as:
$C_B(p_i) = \sum\limits_{j=1}^{N}\sum\limits_{k=1}^{j-1} \frac{g_{jk}(p_i)}{g_{jk}}$,
where g_{jk} is the count of all geodesic paths linking g_j and g_k, and $g_{jk}(p_i)$ is the number of those geodesic paths that involves the node.
- Eigenvector centrality [21] assigns relative scores to all nodes in the network. The scoring principle is that connections to high-scoring nodes result in higher score of the node than equal connections to low-scoring nodes. Eigenvector centrality is calculated as:
$x_i = \frac{1}{\lambda} \sum\limits_{j\epsilon M(i)} x_j$,
where x_j is the score of node j, $M(i)$ is the set of nodes that are connected to node i and λ is a constant.

However, these metrics are difficult to measure in mobile P2P network due to lack of global knowledge of the network topology. The concept of egocentric network, defined as the network consisted of a single node and other nodes that it is connected, is introduced to handle this situation. In this mobile P2P network, when calculating the centralities, we assume they have connection if two nodes have contacted with each other in the last period of time. Moreover, a weight value is assigned for each edge linking two nodes based on the contact frequency. In egocentric network, the method to calculate degree centrality is the same as that mentioned above, since it is the count of the node's connections. Degree centrality and eigenvector centrality consider only the relationship between the node and its directly connected neighbors, which is not enough for evaluating the importance of the node. Closeness centrality cannot be calculated in egocentric network due to lack of information about the distance from the ego node to all other nodes in the network, because it only contains some local information. Although egocentric betweenness values are not the same with the sociocentric betweenness values, egocentric betweenness is shown [19] to be an effective measure to evaluate the importance of the node.

As a result, we choose improved egocentric betweenness to evaluate the importance of the nodes in the network and choose one from the N most important nodes as the index server. In the initial phase of index server discovery, the nodes in the community collect contact information from the neighbors and then calculate egocentric as:

[width=240pt]DataFlowChart.png

Fig. 1. File sharing system data flow

$C(p_i) = \sum\limits_{j=1}^{N}\sum\limits_{k=1}^{j-1} \frac{w_{ji}*w_{ik}*g_{jk}(p_i)}{g_{jk}}$,
where w_{ji} and w_{ik} are the encountering frequencies between node j and node i and that between node i and node k.

2) Communicator Node Selection: Stable and mobile nodes are defined in the view of social network. Stable nodes are important and popular leaders in their communities, meaning they have frequent contacts with the members in the communities. Mobile nodes are those who have wide contact with people both in the communities and outside the communities. Communicators are chosen from the nodes with highest mobility scores. In the last period of time, a mobile node's egocentric network should consists of some nodes in the home community and many external nodes from other communities.

In the mobile P2P network, communicators should serve as carriers of messages between the index server in home community and other communities. It is required that communicators should have tight connections with the data server. There is large probability for a node to be promoted as community communicator if it frequently encounters the index server or other nodes in the community that have strong connection to the index server. Eigenvector centrality assigns relative scores to all nodes in the network by assigning higher scores to nodes with connections to high-scoring nodes than that with connections to low-scoring nodes. It is a great metrics to evaluate the nodes for communicator selection.
$x_i = \frac{1}{\lambda} \sum\limits_{j\epsilon M(i)} x_j$,
where x_j is the score of node j, $M(i)$ is the set of nodes that are connected to node i and λ is a constant.

Upon every encountering of two nodes, they exchange their scoring information and re-calculate the score. Since the communicator nodes are supposed to have strong connections to index server, they report the score to the index server when they meet. In this way, after the instability period of the community, the index server may choose the best ones as communicators and assign the responsibilities to them. Other factors should also be taken into consideration, like storage capacity, the percentage of personal interested files versus community interested files (storage vacancy).But is's out of the scope of the paper.

E. Processes of File Retrieval

In this section, we will describe the details how the file requests are forwarded from the file requestor to the file holder and how the files are sent back to the requestor. The nodes will search the file in their community first and then search the global network if the file doesn't exist in the local community.

Initially, when a user needs a file, the local node checks the local repository. If the file can't be found, it sends out a request to the currently connected neighbors and looks up it in their file repositories. If these neighbors report to the requestor that the file doesn't exist, the requestor sends out a request addressed to the local index server and it is forwarded by the nodes that have the most similar interest with the destination. After receiving the request, the local index server looks up the file in the file index, which contains all the file information in this community. If the data holder address is found, the request will be forwarded to the data holder and the file will be sent back to the requestor. Otherwise, if the file can't be found in the index, it is indicated that no node in this community holds the file and we should refer to other communities for it. A subscription is generated in the index server and the request will be carried away to outside communities by the communicators, which have high mobility and travel between different communities.

The index server in a foreign community checks if this community has the file. If the file isn't held by any node, a failure message will be created and sent back to original index server by communicators. However, if the community happens to hold the requested file, the index server will send the request addressed to the data holder. The file will be sent out from the data holder to the index server, which will send the file back to the original community through the communicators. After arriving in the original community, the file is forwarded hop by hop to the file requestor and the file retrieval processes are done.

Figure 1 depicts how file searching and file retrieving are done locally and globally. In this graph, requestor node R in community C1 sends out a file request. Since its neighbors don't have the file, it is forwarded to the index server IS, which checks the community file index but still can't find it. The community communicators marked with M approach the index server and take the request to distant communities, where the index server IS searches the file in the community and then sends it back.

IV. Experiment and Analysis

To evaluate the performances of our proposed file sharing system in mobile P2P network environment, we conduct simulation studies based on the ONE [22] simulator, which allows users to create mobile P2P network environment based on various movement models and real-world traces. We developed three routing algorithms for the file sharing application simulation on the ONE framework based on the map based movement. The algorithms include normal flooding based routing algorithm, community-based flooding routing algorithm and our social network based routing algorithm.

In the movement model, we assume that every node is a mobile device carried by a person for sharing files they are interested in. We choose a part of the map of a small town as the paths that the nodes move on. The virtual area for the experiment is $4000m \times 5000m$. The device holders have three major categories of interests and they have much higher frequency to go to the related clubs, parties and lectures, so that nodes with the similar interest have higher frequency to contact each other. According to the mobility model, each node starts at a randomly chosen position and moves to another location that is randomly chosen based on his/her interest. They move at a random speed of [0, s_{max}], where s_{max} varies based on different types of the nodes. Each node store 50 files they are interested in within the local repository.

[width=200pt]image001.png [width=200pt]image003.png

(a) Delivery probability (b) Traffic overhead

[width=200pt]image005.png

(c) Number of hops

Fig. 2. Performance evaluation results

We test the three different file sharing systems with different routing algorithms and get some results about delivery probability, traffic overhead and average number of hops per request as shown in the graphs.

Figure 2(a) shows the result of successful delivery rate in the mobile P2P network. Epidemic routing algorithm achieves the best delivery performance as expected in a reasonably limited period. When the size of the network is small, e.g., small than 300, the social network based approach is inferior to the other two algorithms but it get closer as the network grows. It may be due to heavy network traffic in Epidemic approaches and thus more traffic jam and longer transmission delay, which affect adversely the delivery probability. Nevertheless, the social network based routing algorithm generates much smaller traffic and is more efficient. When the size increases, there're more nodes to relay the requests and files to the destination, which leads to larger delivery probability.

Figure 2(b) indicates social network based approach results in much less traffic in the network. Epidemic approaches do not generate much traffic when the network size is small, but it grows quickly as the network size increases. Social network based approach is more efficient because it only choose the most probable routing path to forward the messages.

We found that in our proposed system, the requests need to travel more hops than Epidemic approaches, as shown in figure 2(c). Since the nodes choose the relaying node most similar to the destination and not every possible routing path is tried, the routing algorithm should be able to predict future contacts in order to reduce the number of hops. We need to optimize the social network based routing algorithms in the future work.

V. Conclusion

In this paper, we have described a social network based file sharing system in mobile P2P network, proposed system for file searching and file retrieval in mobile P2P network environment. We designed the social network based routing approach to forward the messages to the destination with specific groups of interests in the proposed file sharing system. Although nodes' disconnections occur frequently due to mobility, we exploited the device holders' mobility in social network to take advantage of it. An important node with frequent contacts with others is elected in an community as index server, which handles file index for files in local community. Several highly mobile nodes that contact external communities are chosen as the communicator when requesting files from foreign communities. We exhibit from the experiment that social network based routing would dramatically reduce the traffic cost per file requested due to avoiding flooding. Taking advantage of node clustering and assigning responsibilities for the index server and communicators can increase the file retrieval success rate while reduce network traffic cost.

ACKNOWLEDGMENT

This research was supported in part by U.S. NSF grants CNS-0834592 and CNS-0832109.

REFERENCES

[1] Napster, "$http://www.napster.com$," 2000.

[2] "Gnutella development forum," 2002.

[3] S. Ratnasamy, P. Francis, S. Shenker, and M. Handley, "A scalable content-addressable network," in *Proceedings of ACM SIGCOMM*, 2001, pp. 161–172.

[4] I. Stoica, R. Morris, D. Karger, M. F. Kaashoek, and H. Balakrishnan, "Chord: A scalable peer-to-peer lookup service for internet applications," 2001, pp. 149–160.

[5] C. Lindemann and O. P. Waldhorst, "A distributed search service for peer-to-peer file sharing," in *Proceedings of the second International Conference on Peer-to-Peer Computing*, 2002.

[6] A. Klemm, C. Lindemann, and O. P. Waldhorst, "A special-purpose peer-to-peer file sharing system for mobile ad hoc networks," 2003.

[7] B. B. Gang Ding, "Peer-to-peer file-sharing over mobile ad hoc networks," in *Pervasive Computing and Communications Workshops, 2004. Proceedings of the Second IEEE Annual Conference on*, 2004, pp. 104–108.

[8] D. W. Anna Hayes, "Peer-to-peer information sharing in a mobile ad hoc environment," in *Mobile Computing Systems and Applications, 2004. WMCSA 2004. Sixth IEEE Workshop on*, 2004, pp. 104– 108.

[9] M. E. J. Newman, "The structure and function of complex networks," *SIAM Review*, vol. 45, pp. 167–256, 2003.

[10] R. Z. Albert and A. lszl Barabsi Director, "Statistical mechanics of complex networks," 2001.

[11] A.-L. Barabasi and J. Frangos, "Linked: New science of networks," 2002.

[12] S. Milgram, "The small world problem," in *Psychology Today 1 (May 1967*, 1967.

[13] A. Chaintreau, P. Hui, J. Crowcroft, C. Diot, R. Gass, and J. Scott, "Impact of human mobility on the design of opportunistic forwarding algorithms," in *in Proceedings of IEEE INFOCOM*, 2006.

[14] J. L. X. Cheng, C. Dale, "Statistics and social network of youtube videos," in *IEEE IWQoS08, Enschede, The Netherlands*, June 2-4, 2008.

[15] A. P. Marco Conti, Franca Delmastro, "Context-aware file sharing for opportunistic networks," in *Mobile Adhoc and Sensor Systems, 2007. MASS 2007. IEEE Internatonal Conference on*, 2007, pp. 1–3.

[16] M. J. J. P. A. Boldrini, Chiara Conti, "Hibop: a history based routing protocol for opportunistic networks," in *World of Wireless, Mobile and Multimedia Networks, 2007. WoWMoM 2007. IEEE International Symposium on a*, 2007, pp. 1–12.

[17] D. Watts, "Small worlds: The dynamics of networks between order and randomness." Princeton University Press, 1999.

[18] M. C. Chiara Boldrini and A. Passarella, "Context and resource awareness in opportunistic network data dissemination," in *World of Wireless, Mobile and Multimedia Networks, 2008. WoWMoM 2008. 2008 International Symposium on a*, 2008, pp. 1–6.

[19] P. V. Marsden, "Egocentric and sociocentric measures of network centrality," in *Social Networks*, October 2002, pp. Vol. 24, No. 4. pp. 407–422.

[20] L. C. Freeman, "Centrality in social networks: Conceptual clarification," in *Social Networks*, 1979, pp. Vol. 1, No. 3. pp. 215–239.

[21] P. Bonacich, "Factoring and weighting approaches to status scores and clique identification," in *Journal of Mathematical Sociology*. Princeton University Press, 1972, pp. 2, 113–120.

[22] T. K. Ari Keranen, Jorg Ott, "The one simulator for dtn protocol evaluation," in *SIMUTools, Rome, Italy*, 2009.