ISSN (ONLINE): 2395-695X
ISSN (PRINT): 2395-695X

International Journal of Advanced Research in Basic Engineering Sciences and Technology (IJARBEST)
Volume 2, Special Issue 19, October 2016

P2P FILE SHARING USING CACHE AND WEB SERVICES WITH SELF REPLICA APPROACHES IN DISCONNECTED MANETS

P.Chitra
Research scholar,
Department of Computer Science,
Bharathidasan College of Arts and Science,
Erode.
Chitramsc69@gmail.com

Abstract—*This paper proposes a P2P content-based file sharing system, namely Social Network Group (ESPOON), for disconnected MANETs. The system uses an interest extraction algorithm to derive a node's interests from its video files for content-based file searching. For efficient video file searching, Enhanced Social network (ESPOON) groups common-interest nodes that frequently meet with each other as community's network. It takes advantage of node mobility by delegating stable nodes, which have the most frequent video file with community members, as community coordinators for intra community searching, and highly mobile nodes that visit other communities frequently as community ambassadors for intercommunity searching. In addition paper using catch model and web services model used to video file share for mobile ad hoc network environments. In addition paper, the proposed cache approach maintains the user's query in the cache database services. When user submits the query for video file sharing, first the system will look into the cache database thus reduce the execution time. In the web services approach, when query comes from a client node for the video file sharing initially it search with clients and if no results found then it will redirect to the server. The proposed web services methods provides the client to client communication for video file sharing effectively. In a mobile ad hoc network, the mobility and resource constraints of mobile nodes may lead to network partitioning or performance degradation. In reality, however, some nodes may selfishly decide only to cooperate partially, or not at all, with other nodes. These selfish nodes could then reduce the overall data accessibility in the network. This paper analysis the impact of selfish nodes in a mobile ad hoc network from the perspective of replica allocation. This is termed as selfish replica allocation (SRA).*

Keywords— Video File Share, ESPOON, cache and web service, Self Center Friendship Tree, Self Replica Allocation.

I. INTRODUCTION

Today, mobile users interact with each other and share files via an infrastructure formed by geographically distributed base stations. However, users may find themselves in an area without wireless service (e.g., mountain areas and rural areas). Moreover, users may hope to reduce the cost on the expensive infrastructure network data.P2P file sharing model makes large-scale networks a blessing instead of a curse, in which nodes share files directly with each other without a centralized server. Wired P2P file sharing systems like Bit Torrent have already become a popular and successful paradigm for file sharing among millions of users. The successful deployment of P2P file sharing systems and the aforementioned impediments to file sharing in MANETs make the P2P file sharing over MANETs (P2P MANETs in short) a promising complement to current infrastructure model to realize pervasive file sharing for mobile users.

The mobile digital devices are carried by people that usually belong to certain social relationships. So this Paper is focused on the P2P file sharing in a disconnected MANET community consisting of mobile users with social network properties. In such a file sharing system, nodes meet and exchange requests and files in the format of text, short videos, and voice clips in different interest categories. Such scenarios ensure for the most that nodes sharing the same interests (i.e., math), carry corresponding files (i.e., math files), and meet regularly (i.e., attending math classes).

MANETs consisting of digital devices in which nodes are constantly moving, forming disconnected MANETs with opportunistic node encountering. Such transient network connections have posed a challenge for the development of P2P MANETs. MANETs are either flooding-based [11] or advertisement-based [12]. The former methods rely on flooding for file searching. However, they lead to high overhead in broadcast. In the latter methods, nodes advertise their available files, build content tables, and forward files according to these tables. But they have low search efficiency because of expired routes in the content tables caused by transient network connections. Also, advertising can lead to high overhead. Some researchers [1] further proposed to utilize cache/replication to enhance data dissemination/access efficiency in disconnected MANETs. However, nodes in these methods passively wait for contents that they are interested in rather than actively search files, which may lead to a high search delay. Recently, social networks are exploited to facilitate content dissemination/publishing in disconnected MANETs. These methods exploit below property to improve the efficiency of message forwarding:

(P1) nodes (i.e., people) usually exhibit certain movement patterns (e.g., local gathering, diverse centralities, and skewed visiting preferences). However, these methods are only for the dissemination of information to subscribers. They are not specifically designed for file searching. Also, they fail to take into account other properties of social networks revealed by recent studies to facilitate content sharing:

(P2) Users usually have a few file interests that they visit frequently [13] and a user's file visit pattern follows a power-law distribution [14]. (P3) Users with common interests tend to meet with each other more often than with others [15].By leveraging these properties of **S**ocial network-based P2P c**O**ntent-based file sharing is proposed in disconnected m**O**bile ad hoc **N**etworks (SPOON) with four components as shown in **Fig.1.1:**

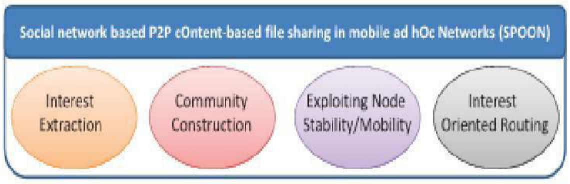

Fig 1.1 Components of SPOON

Based on P2, an interest extraction algorithm is proposed to derive a node's interests from its files. The interest facilitates queries in content-based file sharing and other components of SPOON. A collective of nodes is referred that share common interests and meet frequently as a community. According to P3, a node has high probability to find interested files in its community. If this fails, based on P1, the node can rely on nodes that frequently travel to other communities for file searching. Thus, the community construction algorithm is proposed to build communities to enable efficient file retrieval.

According to P1, a node role assignment algorithm is proposed that takes advantage of node mobility for efficient file searching. The algorithm designates a stable node that has the tightest connections with others in its community as the community coordinator to guide intercommunity searching. For each known foreign community, a node that frequently travels to it is designated as the community ambassador for intercommunity searching. An interest-oriented file searching and retrieval scheme is proposed that utilizes an Interest-oriented Routing Algorithm (IRA) and above three components. Based on P3, IRA selects forwarding node by considering the probability of meeting interest keywords rather than nodes.

The file searching scheme has two phases: Intra-and intercommunity searching. In the former, a node first queries nearby nodes, then relies on coordinator to search the entire home community. If it fails, the intercommunity searching uses an ambassador to send the query to a matched foreign community. A discovered file is sent back through the search path or the IRA if the path breaks. SPOON is novel in that it leverages social network properties of both node interest and movement pattern. First, it classifies common-interest and frequently encountered nodes into social communities. Second, it considers the frequency at which a node meets different interests rather than different nodes in file searching. Communities for file searching. Thus, the community construction algorithm is proposed to build communities to enable efficient file retrieval.

According to P1, a node role assignment algorithm is proposed that takes advantage of node mobility for efficient file searching. The algorithm designates a stable node that has the tightest connections with others in its community as the community coordinator to guide intercommunity searching. For each known foreign community, a node that frequently travels to it is designated as the community ambassador for intercommunity searching. An interest-oriented file searching and retrieval scheme is proposed that utilizes an Interest-oriented Routing Algorithm (IRA) and above three components. Based on P3, IRA selects forwarding node by considering the probability of meeting interest keywords rather than nodes.

The file searching scheme has two phases: Intra-and intercommunity searching. In the former, a node first queries nearby nodes, then relies on coordinator to search the entire home community. If it fails, the intercommunity searching uses an ambassador to send the query to a matched foreign community. A discovered file is sent back through the search path or the IRA if the path breaks. SPOON is novel in that it leverages social network properties of both node interest and movement pattern. First, it classifies common-interest and frequently encountered nodes into social communities. Second, it considers the frequency at which a node meets different interests rather than different nodes in file searching.

Third, it chooses stable nodes in a community as coordinators and highly mobile nodes that travel frequently to foreign communities as ambassadors. Such a structure ensures that a query can be forwarded to the community of the queried file quickly. SPOON also incorporates additional strategies for file perfecting, querying-completion and loop-prevention, and node churn consideration to further enhance file searching efficiency

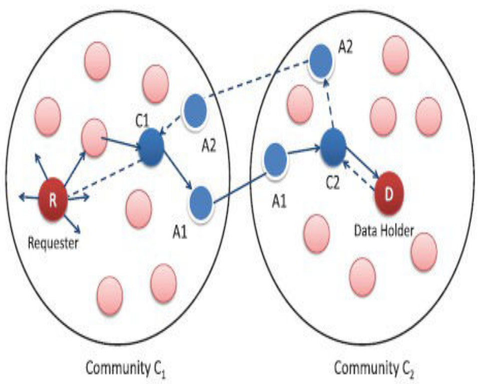

Fig1.2 File Searching in SPOON

II. RELATED WORK

A. P2P File Sharing in MANETs

We first introduce the P2P file sharing algorithms designed in MANETs.

1)Flooding-Based Methods:

In flooding-based methods, 7DS [6] is one of the first approaches to port P2P technology to mobile environments. It exploits the mobility of nodes within a geographic area to disseminate web content among neighbors. Passive distributed indexing (PDI) [8] is a general-purpose distributed file searching algorithm. It uses local broad-casting for content searching and sets up content indexes on nodes along the reply path to guide subsequent searching. Klemm et al. [7] proposed a special-purpose on-demand file searching and transferring algorithm based on an application layer overlay network. The algorithm transparently aggregates query results from other peers to eliminate redundant routing paths. Hayes [9] extended the Gnutella system to mobile environments and proposed the use of a set of keywords to represent user interests. However, these flooding-based methods produce high overhead due to broadcasting.

2)Advertisement-Based Methods:

Tchakarov and Vaidya [10] proposed GCLP for efficient content discovery in location-aware ad hoc networks. It disseminates contents and requests in crossed directions to ensure their encountering. P2PSI [11] combines both advertisement (push) and discovery (pull) processes. It adopts the idea of swarm intelligence by regarding shared files as food sources and routing tables as pheromone. Each file holder regularly broadcasts an advertisement message to inform surrounding nodes about its files. The discovery process locates the desired file and also leaves pheromone to help subsequent search requests. Repantis and Kalogeraki [12] proposed a file sharing mechanism in which nodes use the Bloom filter to build content synopses of their data and adaptively disseminate them to other nodes to guide queries. Though the advertisement-based methods reduce the over-head offlooding-based methods, they still generate high overhead for advertising and cannot guarantee the success of file searching due to node mobility.

B.P2P File Sharing in Disconnected MANETs

The disconnected MANETs are featured by sparse node density and intermittent node connection, which makes previously introduced methods infeasible in such networks. We then further introduce two categories of P2P file sharing methods for disconnected MANETs.

1) Cache/Replication-Based Methods:

Huang et al. [13] proposed a method that considers multiple factors (e.g., node mobility, file popularity, and file server topology) in creating file replicas in file servers to realize optimal file availability in content distribution community. Gao et al. proposed cooperative caching in disruption tolerant networks. It replicas each file to network central locations, which are frequently visited by nodes in the system, to ensure efficient data access. QCR uses file caching to realize effective multimedia content dissemination in opportunistic networks. In addition to node mobility and file popularity, it also considers the impatience of users when creating replicas. Lenders et al. investigated wireless ad hoc podcasting, in which nodes store contents from their neighbors that are interested by themselves or the nodes they have met.
Chen and Shen deduced the optimal file replication strategy in MANETs by further considering nodes' ability to meet nodes as a resource because replicas on these nodes can meet more requesters and, thus, have higher availability. Though these methods improve file availability, nodes in these methods passively wait for contents they are interested in rather than actively search files, which may lead to search delay.

3)Social Network-Based Methods:

Recently, social networks have been utilized in content publishing/dissemination algorithms in opportunistic networks. MOPS provides content-based sub/pub service by utilizing the long-term neighboring relationship between nodes. It groups nodes with frequent contacts and selects nodes that connect different groups as brokers, which are responsible for intercommunity communication. Then, contents and subscriptions are relayed through brokers to reach different communities. MOPS only considers node mobility, while SPOON is more advantageous by considering both node interest and mobility as described previously. Moreover, unlike MOPS that only depends on the meeting of brokers for intercommunity search, SPOON enhances the efficiency of intercommunity search by 1) assigning one ambassadorfor each known foreign community, which helps to forward a query directly to the destination community, and 2) utilizing stable nodes (coordinator) to receive messages from ambassadors.

The work in is similar to MOPS. It selects centrality nodes as brokers and builds them into an overlay, in which brokers use uncast or direct protocols (e.g., WiFi access points) for communication. Then, node publications are first transferred to the broker node responsible for the node's community and then propagated to all brokers to find matched subscribers. Social Cast calculates a node's utility value on an interest based on the node's mobility and collocation with the nodes subscribed to the interest. It publishes contents on an interest to subscribers by forwarding the contents to nodes with the highest utilities on the interest. Content Place defines social relationship-based communities and a set of content caching policies. Specifically, each node calculates a utility value of published data it has met based on the data's destination and its connected communities, and caches the data with the top highest utilities. However, above methods mainly focus on disseminating publications to matched subscribers. Therefore, these methods cannot be applied to file searching directly.

III.PROBLEM DEFINITION

The P2P file sharing model makes large-scale networks a blessing instead of a curse, in which nodes share files directly with each other without a centralized server. Wired P2P file sharing systems have already become a popular and successful paradigm for file sharing among millions of users. The successful deployment of P2P file sharing systems and the aforementioned impediments to file sharing in MANETs make the P2P file sharing over MANETs (P2P MANETs in short) a promising complement to current infrastructure model to realize pervasive file sharing for mobile users.

As the mobile digital devices are carried by people that usually belong to certain social relationships, in this paper, we focus on the P2P file sharing in a disconnected MANET community consisting of mobile users with social network properties. In such a file sharing system, nodes meet and exchange requests and files in the format of text, short videos, and voice clips in different interest categories.

In MANETs consisting of digital devices, nodes are constantly moving, forming disconnected MANETs with opportunistic node encountering. Such transient network connections have posed a challenge for the development of P2P MANETs. Traditional methods supporting P2P MANETs are either flooding-based or advertisement-based the former methods rely on flooding for file searching.

However, they lead to high overhead in broadcast. In the latter methods, nodes advertise their available files, build content tables, and forward files according to these tables. But they have low search efficiency because of expired routes in the content tables caused by transient network connections. Also, advertising can lead to high overhead.

SPOON is novel in that it leverages social network properties of both node interest and movement pattern. First, it classifies common-interest and frequently encountered nodes into social communities. Second, it considers the frequency at which a node meets different interests rather than different nodes in file searching. Third, it chooses stable nodes in a community as coordinators and highly mobile nodes that travel frequently to foreign communities as ambassadors. Such a structure ensures that a query can be forwarded to the community of the queried file quickly.

IV.SYSTEM METHODOLOGY

A. FLOODING BASED METHOD

This module shows the basic structure of the proposed broadcasting format. Each node schedules a broadcast. Clearly, each message is broadcast once at most by a node. A broadcast schedule can be set at any time. For example, a message can be dropped after the first reception but scheduled for broadcast the second time. The node will broadcast the data to maximum number of nodes nearest to it

B. SOCIAL NETWORKING BASED METHOD

Intra-community File Searching and Retrieval

The query message is represented by a query vector. Each query is associated with a counter (count) indicating the number of hops it can travel. The count is decremented by one after each forwarding the query. Since the query is initiated by users, term weights in V_Q are constant values. In the intra-community searching, the destination that a query is sent to is represented by a combination of the V_Q and the node vector of the requester's community coordinator. The requester calculates the similarity between the query vector and the community vector of the community it belongs to. The query is sent to the coordinator of the community directly when the counter is larger than 0.

Intercommunity File Searching and Retrieval

The intercommunity searching algorithm, a coordinator maps a request to the foreign community that is most likely to contain the queried file. Similar to the intercommunity search step, the coordinator also uses the multicity forwarding strategy, i.e., it sends out a query to ambassadors having the highest similarity with the query to enhance the efficiency of the forwarding.

C. SELFISH REPLICA ALLOCATION

In this module, selfish replication allocation is shown. Fig. 1 illustrates an existing replica allocation scheme, DCG, where nodes N1, N2, ... ,N6 maintain their memory space M1, M2, ... ,M6, respectively, with the access frequency information in Table 1 (In Fig. 1, a straight line denotes a wireless link, a gray rectangle denotes an original data item, and a white rectangle denotes a replica allocated.

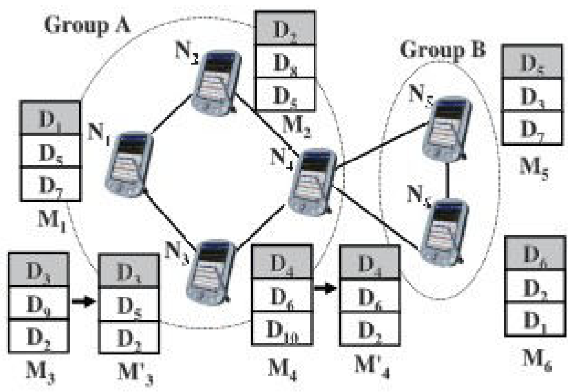

Fig 4.1 Self Replica Allocation

In Fig 4.1, the gray colored area shows three data items that are accessed frequently by N3 and N4). As shown in Fig. 1, DCG seeks to minimize the duplication of data items in a group to achieve high data accessibility. Let us consider the case where N3 behaves "selfishly" by maintaining M'3, instead of M3, to prefer the locally frequently accessed data for low query delay. In the original case, D3, D9, and D2 were allocated to N3.

However, due to the selfish behavior, D3, D5, and D2, the top three most locally frequently accessed items, are instead maintained in local storage. Thus, other nodes in the same group, i.e., N1, N2, and N4, are no longer able to access D9. This showcases degraded data accessibility, since N1, N2, and N4 cannot fully leverage N3's memory space as intended in cooperative replica sharing.As another example, a node may be only "partially selfish" in a MANET. For instance, node N4 may want to locally hold D2, one of the locally frequently accessed data items. In this case, N4 uses only a part of its storage for its own frequently accessed data, while the remaining part is for the benefit of overall data accessibility. Thus, N4 may decide to maintain M'4, instead of M4. Even with only partial selfishness, data accessibility is still degraded, since the other nodes in the same group, i.e., N1, N2, and N3, cannot access D10.

At every relocation period

/* N_i detects selfish nodes with this algorithm */

```
detection(){
for (each connected node Nk ){
 if (nCRi^k < Nk is marked as non-selfish;
else Nk is marked as selfish;}
wait until replica allocation is done;
for (each connected node Nk){
if (Ni has allocated replica to Nk ){
NDi^k = the number of allocated replica;
SSi^k = the total size of allocated replica;}
else{
NDi^k = 1;
SSi^k = the size of a data item;
} } }
```

At every query processing time

```
1: /* When Ni issues a query */
2: update_SF(){
3: while (during the predefined time ω){
4: if (an expected node Nk serves the query)
5: decrease Pi^k;
6: if (an unexpected node Nj serves the query){
7: NDi^j = NDi^j + 1;
8: SSi^j = SSi^j + (the size of a data item);
9: } }
10: if (an expected node Nk does not serve the query){
11: increase Pi^k;
12: NDi^k = NDi^k - 1;
13: SSi^k = SSi^k - (the size of a data item);
14: } }
```

VRESULTS ANDDISCUSSION

The Table **6.1.** describes the search file analysis of SPOON method and self replica allocation. The table contains the input file dataset, search file size and probability of the search files for the SPOON and self replica allocation methodologies. The probability of search file for the SPOON method is computed as:

Search File size $S_{SP} = \sum N_{FS}/N * 2/100$.

And the probability of search file for the self replica allocation method is computed as:

Search File size $S_{SRA} = \sum N_{FS}/N * 2/100$.

The figure 6.1. shows the search file analysis of the SPOON method. The graph values in the x-axis shows the size of the file dataset (N) and the y-axis shows the average probability of the search file.

VI CONCLUSION

The difficulty in distributing the content in the server is eliminated by using this application. It reduces the server bandwidth to consistent amount. The end users need not wait for server in downloading the video file since the P2P application gets the videos from available client nodes. A good documentation of user-friendly features had been incorporated in the system. The system has been introduced to eliminate human error. It minimizes the time consumption and design & development work.

In contrast to the network viewpoint, the project addressed the problem of selfish nodes from the replica allocation perspective. It termed this problem selfish replica allocation The project work was motivated by the fact that a selfish replica allocation could lead to overall poor data accessibility in a MANET. We have proposed a selfish node detection method and novel replica allocation techniques to handle the selfish replica allocation appropriately.

Table 6.1. SPOONS and Self Replica Allocation Search File Analysis

File Dataset (N)	**SPOON search File**		Self Replica Allocation With Search File Size	
	N_{FS} [Size Search File]	**S_{SP} [Probability Search File]**	**N_{FS} [Search File]**	S_{SRA} [Probability Search File]
100	62	0.0124	73	0.0146
200	131	0.0131	153	0.0153
300	233	0.0155	246	0.0164
400	347	0.0173	363	0.0181
500	443	0.0177	462	0.0184

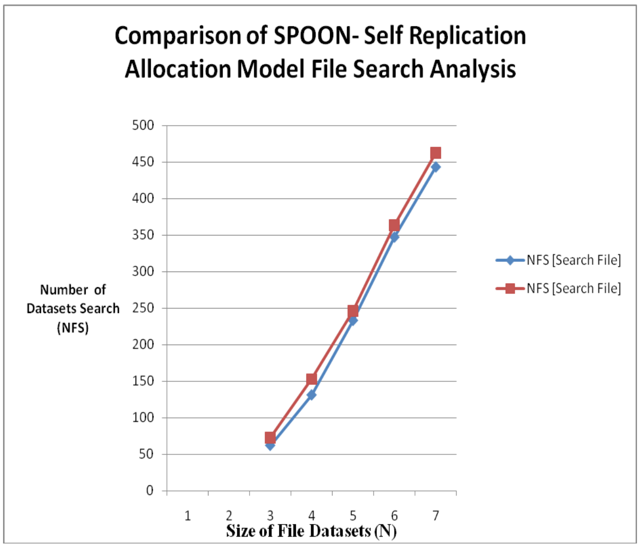

Fig 6.1Comparison of SPOON- Self Replication Allocation Model FileSearch Analysis

The proposed strategies are inspired by the real-world observations in economics in terms of credit risk and in human friendship management in terms of choosing one's friends completely at one's own discretion. We applied the notion of credit risk from economics to detect selfish nodes. Every node in a MANET calculates credit risk information on other connected nodes individually to measure the degree of selfishness. Since traditional replica allocation techniques failed to consider selfish nodes, it also proposed novel replica allocation techniques.

VII FUTURE ENHANCEMENTS

The future strategies may outperform existing representative cooperative replica allocation techniques in terms of data accessibility, communication cost, and query delay. Future work will be on the impact of different mobility patterns. The new system is designed such that those enhancements can be integrated with current modules easily with less integration work.

The process of preparing plans had been a new experience, which was found use full in later phases of the project is completed. Efforts had been taken to make the system user friendly and as simple as possible. However at some points some features may have been missed out which might be considered for further modification of the application. The new system become useful if the below enhancements are made in future.

- The statistical analysis of download data if prepared can be used for further development.
- The P2P application if developed as web site can be used from anywhere.
- The user portal if developed can assist in maintaining download history for end users.

REFERENCE

[1] Y. Huang, Y. Gao, K. Nahrstedt, and W. He, "Optimizing File Retrieval in Delay-Tolerant Content Distribution Community," Proc. IEEE 29th Int'l Conf. Distributed Computing Systems (ICDCS '09), 2009.

[2] J. Reich and A. Chaintreau, "The Age of Impatience: Optimal Replication Schemes for Opportunistic Networks," Proc. Fifth Int'l Conf. Emerging Networking Experiments and Technologies (CoNEXT '09), 2009.

[3] V. Lenders, M. May, G. Karlsson, and C. Wacha, "Wireless Ad Hoc Podcasting," ACM SIGMOBILE Mobile Computing and Comm. Rev., vol. 12, pp. 65-67, 2008.

[4] F. Li and J. Wu, "MOPS: Providing Content-Based Service in Disruption-Tolerant Networks," Proc. IEEE 29th Int'l Conf. Distributed Computing Systems (ICDCS '09), 2009.

[5] C. Boldrini, M. Conti, and A. Passarella, "ContentPlace: Social-Aware Data Dissemination in Opportunistic Networks," Proc. 11th Int'l Symp. Modeling, Analysis and Simulation Wireless and Mobile Systems (MSWiM '08), 2008.

[6] M. Papadopouli and H. Schulzrinne, "A Performance Analysis of 7DS: A Peer-to-Peer Data Dissemination and Prefetching Tool for Mobile Users," Proc. IEEE Sarnoff Symp. Digest Advances in Wired and Wireless Comm., 2001.

[7] J.B. Tchakarov and N.H. Vaidya, "Efficient Content Location in Wireless Ad Hoc Networks," Proc. IEEE Int'l Conf. Mobile Data Management (MDM '04), 2004.

[8] A. Fast, D. Jensen, and B.N. Levine, "Creating Social Networks to Improve Peer-to-Peer Networking," Proc. 11th ACM SIGKDD Int'l Conf. Knowledge Discovery in Data Mining (KDD '05), 2005.

[9] A. Iamnitchi, M. Ripeanu, and I.T. Foster, "Small-World File-Sharing Communities," Proc. IEEE INFOCOM, 2004.

[10] M. Mcpherson, "Birds of a Feather: Homophily in Social Networks," Ann. Rev. Sociology, vol. 27, no. 1, pp. 415-444 , 2001.