Middle-East Journal of Scientific Research 22 (7): 1063-1071, 2014
ISSN 1990-9233
© IDOSI Publications, 2014
DOI: 10.5829/idosi.mejsr.2014.22.07.21967

# Spoon-P2P Content-Based File Sharing in Disconnected Manets

*[1]N. Srinivasan, [2]A. Raisha Banu, [3]R. Srinath, [3]B. Subhasini and [3]M. AntoBennet*

[1]Department of EEE, VELTECH, Chennai- 600062, India
[2]Department of IT, Rajalakshmi Engineering College, Chennai-602105, India
[3]Department of ECE, VELTECH, Chennai-600062, India

**Abstract:** Present peer-to-peer (P2P) file distribution methods in mobile ad hoc networks (MANETs) can be divided into three categories: local broadcasting based advertisement (push) and discovery (pull)-based and contact-based. The first two techniques can simply be time consuming and low ability to accommodate when the demand grows higher. They are mainly developed for linked MANETs, in which end-to-end relativity among nodes is preserved. The contact-based methods adjust to the adaptable nature of disconnected MANETs but fail to regard the social contents of portable nodes, which can be subjugated to advance the file searching effectiveness. In this paper, we suggest a P2P content-based file distribution system, namely SPOON, for disconnected MANETs. The system uses an interest mining algorithm to derive a node's concern from its files for content-based file searching. For competent file searching, SPOON assembles similar-interest nodes that frequently gather with each other as a set. It takes the benefit of node portability by designating constant nodes, which has the most common contact with neighborhood members, as community coordinator for hunt within the community and highly-mobile nodes that visit other communities frequently as community ambassador for search in other communities. An interest-oriented file searching scheme is projected for high file searching efficiency. Supplementary strategies for file transfer, request-completion and avoiding occurrence of loops and node churn consideration are measured to further enhance the file searching competence. We developed a virtual environment using JAVAFX and MYSQL to test our system. The test outcome show that our system considerably lowers communication cost and improves file searching success rate compared to present methods.

**Key words:** MANETs · Content-based file sharing · Mobility · Coordinator · Ambassador · Community

## INTRODUCTION

Since in the former years, personal mobile devices such as Laptops, PDAs and smartphones have been more and more widespread. The mobile phone users are expected to grow in enormous number in the recent years [1]. The mobile users are growing in enormous number worldwide therefore, there is a need for effective file sharing among them. The number of smartphone users has gradually amplified by 72% in the year 2007 [2]. The incredible rapid growth of mobile users is leading to a hopeful future, in which they can freely share files between each other whenever and wherever.The number of Smartphones shipped will change over a six year period to 2013, as well as what proportion of total mobile devices will be Smartphones. To successfully aid the mobile users to interact with each other when they are present in the remote regions by some appropriate techniques [3]. Currently, mobile users interact with each other and share files via an infrastructure formed by geographically distributed base stations. But, users may find themselves in an extent deprived of wireless provision (e.g. hilly areas and countryside areas) [4]. Moreover, users may expect to reduce the cost on the costly infrastructure network data. BitTorrent is one of the most prevalent ways of partaking and downloading files. Multimedia files which include a mixture of text, video, still images, animation, audio, or interactivity content forms are often distributed with its. The Bit Torrent [5] is a protocol which is meant for disseminating hefty amount of data in a decentralized manner. The KaZaA Overlay study was conducted to overcome the complications in sharing the file, when number of participating user gets increased tremendously [6]. The measurement provides a design principle for web

**Corresponding Author:** N. Srinivasan, Department of EEE, VELTECH, Chennai- 600062, India.

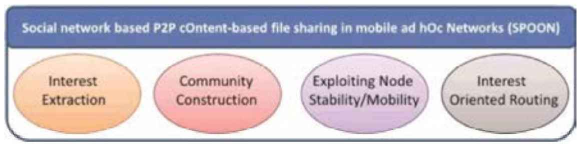


Fig. 1: Components of SPOON.

designer and to engineers in managing the ISP. In Peer to peer data transmission, the data can disseminated either by using Info-station or without any access point [7]. DS is an architecture or set of protocol stating of how the data can be shared among several peers [8].

In MANETs consisting of digital devices, hosts are continuously moving, making disconnected MANETs with opportunistic node encountering. Such temporary network links have posed a challenge for the development of P2P MANETs [1].

The former methods rely on flooding for file searching. However, they lead to high overhead in transmission. In the latter methods, nodes publicize their available files; build content tables and transfer files according to these tables. But they have low search efficacy as of expired routes in the content tables caused by transient network connections. Also, publicizing can lead to high overhead [9].

These properties are exploited to improve the efficiency of message forwarding:

- (P1) nodes (i.e., people) frequently display certain mobility patterns (e.g., local get-together, varied centralities and tilted visiting favorites).

Though, these methods are only for the distribution of data to subscribers. They are not definitely intended for file searching.

Similarly, they fail to take into account additional properties of social networks revealed by recent studies to enable content sharing:

- (P2) Users usually have a few file interests that they visit frequently and a user's file visit pattern follows a power-law distribution.
- (P3) Users with common interests tend to meet with each other more often than with others.

By leveraging these properties of social networks, we propose Social network-based P2P cOntent-based file sharing in disconnected mObile ad hoc Networks (SPOON) with four components as shown in Fig. 1 [10]:

- Based on P2, we propose an interest extraction algorithm to derive a node's interests from its files. The interest facilitates queries in content-based file sharing and other components of SPOON.
- We refer to a collective of nodes 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, we propose the community construction algorithm to build communities to enable efficient file retrieval.

- According to P1, we propose a node role assignment algorithm that takes advantage of node mobility for efficient file searching. The algorithm designates stable node that has the tightest connections with others in its community as the community coordinator to guide intracommunity searching. For each known foreign community, a node that frequently travels to it is designated as the community ambassador for intercommunity searching.4. We propose an interest-oriented file searching and retrieval scheme 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.

## Related Works

**P2P File Sharing in MANETs:** We have mention the former approach of the P2P file sharing algorithms designed for connected MANETs

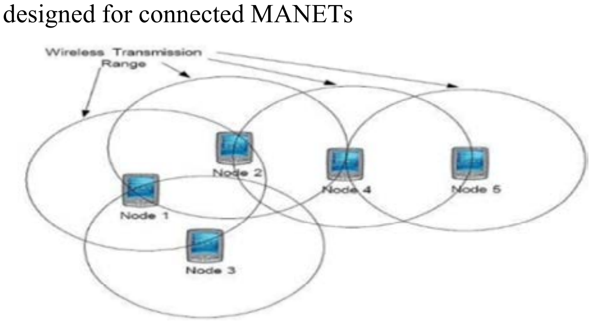


Fig. 2: Data transmission in Connected MANETs

**Flooding-Based Methods:** In flooding-based methods, 7DS [9] 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) is a general-purpose distributed file searching algorithm. It uses local broadcasting for content searching and sets up content indexes on nodes along the reply path to guide subsequent searching. A special-purpose on demand file searching and transferring algorithm was used that transparently aggregates query results from other peers to eliminate redundant routing paths. However, these flooding-based methods produce high overhead due to broadcasting.

**Advertisement-Based Methods:** Tchakarov and Vaidya [8] 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 [10] 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 its surrounding nodes about its holed files. The discovery process locates the desired file and also leaves pheromone to help subsequent search requests. Though the advertisement-based methods reduce the overhead of flooding-based methods, they still generate high overhead for advertising and cannot guarantee the success of file searching due to node mobility.

**Hurdles in Former Approach:** In the former approach of file sharing system in MANETs, the nodes may have high overhead and low scalability.

If any intermediate node is crash the data may be lost on reaching the destination. In order to overcome these difficulties in the former approach we introduce the following approach.

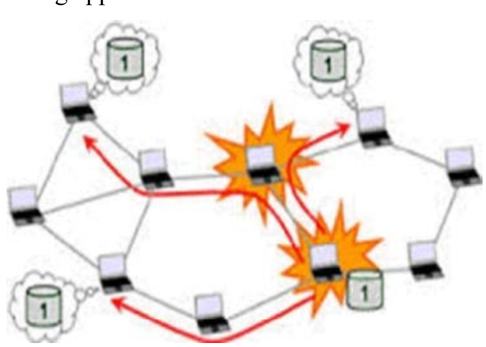

Fig. 3: Data loss in MANETs

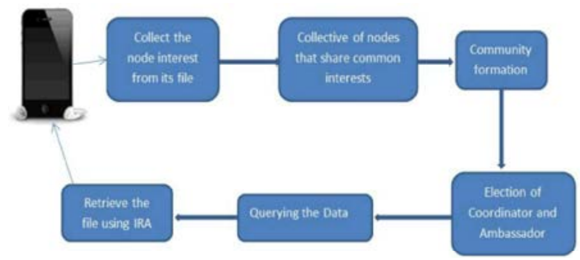


Fig. 4: Structural Design

**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. So, we further present the approach of P2P file sharing methods for disconnected MANETs is Social Network-Based Method.

**Social Network-Based Methods:** This modus provides a content based services utilizing the long term neighboring relationship between nodes. It groups nodes with frequent contacts and selects nodes that connect different groups as ambassadors, which are responsible for intercommunity communication. The one which has the least mobility is elected as the community coordinator responsible for intra-community file searching and the node which has the highest mobility is elected as the community ambassador responsible for intercommunity file searching. The ambassador of one community communicates with the coordinator of other community for querying and fetching of data. The retrieval of data is based on the node's interest and mobility. SPOON enhances the efficiency of intercommunity search by 1) assigning one ambassador for 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 [6] is similar to MOPS. It selects centrality nodes as brokers and builds them into an overlay, in which brokers use unicast or direct protocols (e.g., Wi-Fi 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.

**Spoon Strategy:** In this section, we concentrate on the design of SPOON.SPOON contains portrayal (1) *Interest Extraction Algorithm* to extract the information the mobile node and thereby to create community. (2) *Community Construction* to form a community based on the contents it possess. (3)*Node Role Assignment* to

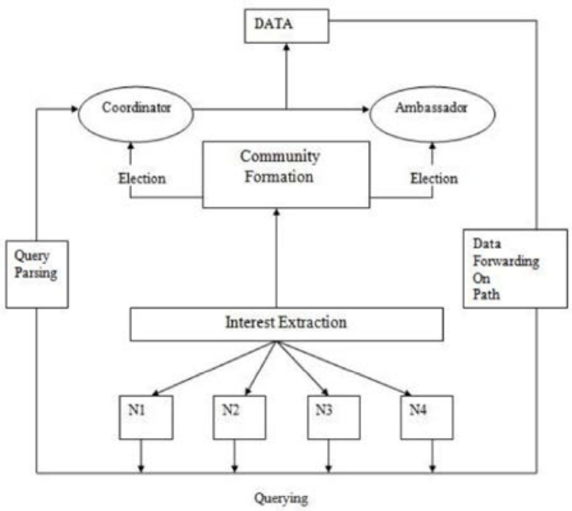


Fig. 5: Detailed Design

assign the responsibilities to the node to query and extract the content which the node possess. (4) *Algorithm used* to manipulate the node

**Interest Extraction:** We assume that node contents can be classified to different interest categories. It was found that users usually have a few file categories that they query for files frequently in a file sharing system. First we derive a node's interests from its files. The interest facilitates queries in content-based file sharing and other components of SPOON. Collective of nodes that share common interests and meet frequently is grouped as a community in which a node having high probability to find interested files in its community. The probability of similar interested nodes meeting together and sharing is high. If this fails the node can rely on nodes that frequently travel to other communities for file searching. We build the community for efficient file searching.

**Community Construction:** Social network theory reveals that people with the same interest tend to meet frequently [7]. By exploiting this property, SPOON classifies nodes with common interests and frequent contacts into a community to facilitate interest-based file searching. Nodes with multiple interests belong to multiple communities. The community construction can easily be conducted in a centralized manner by collecting node interests and contact frequencies from all nodes to central node.

First we derive a node's interests from its files. The interest facilitates queries in content-based file sharing and other components of SPOON. Collective nodes that share common interests and meet frequently are

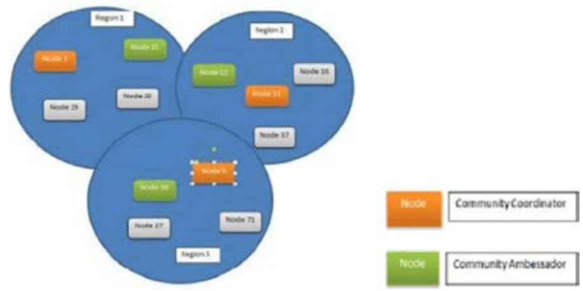


Fig. 6: Data transmission in Disconnected MANETs

categorized as a community. The search is made by the node which has the high probability to find interested files in its community. If this fails the node can rely on nodes that frequently travel to other communities for file searching. We build the community for efficient file searching.

**Node Role Assignment:** From the past we came to know that in a social network consisting of several nodes. So, we find the nodes based on some traits like node's mobility, strength and distance and based on this the nodes are assigned a responsibilities to query the data from the mobile node. We assign two roles namely

*Community Coordinator t*o maintain the index of similar interested files of mobile nodes.(2)*Community Ambassador* to query and retrieve the content from dissimilar community [10].

**Community Coordinator:** We define a stable node that has tight contact frequency with other community members as the community coordinator. In network analysis, centrality is often used to determine the relative importance of a vertex within the network. We then adopt the improved degree centrality, which assigns weight to each link based on the contact frequency, for coordinator selection because it reflects the tightness of a node with other community members. In the initial phase of coordinator discovery, each node, say node Ni, in a community collects contact information from its neighbors in the same community and then calculates its degree centrality by

$$D(pi) = \frac{N}{\sum wij\ J=1} \qquad (1)$$

where wij is the link weight between Ni and Nj and N is the number of neighbors in the same community. To reflect the property that the coordinator has the most connections with all community members, wij equals 1 if the contact frequency between Ni and Nj is larger than a threshold and 0 otherwise. Though such a method cannot ensure its connection to every community member, it ensures that the coordinator has the tightest overall connection to all community members.

Each node periodically checks its degree centrality and broadcasts such information to all community members. If a node receives no larger centrality score than its own centrality for three consecutive periods, it claims itself as the potential coordinator. The potential coordinator would confirm its status as the coordinator when meets the previous one. If it is confirmed, it then requests the community information from the old coordinator. Also, when the new coordinator meets community members, they exchange information for group vector update and ambassador selection, as well as request routing.

**Community Ambassador:** An ambassador is used to bridge the coordinator in its home community and a foreign community. We use the product of a node's contact frequency with its coordinator and that with the foreign community for ambassador selection. Each node i calculate its utility value for foreign community k by

$$Uik = F(Ni;Ck) *F(Ni;Nc), \quad (2)$$

where Ck represents foreign community k, Nc is the coordinator in its home community and F (.) denotes the meeting frequency. Each node reports its utility values for foreign communities it has met to the coordinator in its home community. Then, the community coordinator chooses one ambassador for each known foreign community. Also, the node that has the highest overall contact frequency with all foreign communities is selected as the default ambassador.

In case that a request fails to find a matched ambassador, the default ambassador can carry the request and seek for potential forwarders in foreign communities. If an ambassador loses the connection with the coordinator for a certain period of time, a new ambassador that satisfies above requirements is selected. This arrangement facilitates interest-oriented file searching by enabling a coordinator to send file requests to matched foreign communities quickly. In above design, ambassadors are the key to connect different communities efficiently. Coordinators achieve balance between the centralized and distributed searching by checking whether a community can satisfy a query quickly, which is important in disconnected MANETs.

Also, though broadcast is used in coordinator selection, the cost is limited because 1) it is only among community members and 2) we can set a long inter broadcast period because nodes usually have stable degree centrality. To select ambassadors, each node just reports its utility values to the coordinator, which can be piggybacked on the beacon messages. Therefore, this step does not incur significant extra costs.

**Querying and Rendering the Data:** The interest-oriented file searching scheme has two steps: intra- Community and inter-community searching. A node first searches files in its home community. If the coordinator finds that the home community cannot satisfy a request, it launches the inter-community searching and forwards the request to an ambassador that will travel to the foreign community that matches the request's interest. A request is deleted when its TTL (Time To Live) expires. During the search, a node sends a message to another node using the interest-oriented routing algorithm (IRA), in which a message is always forwarded to the node that is likely to hold or to meet the queried keywords. The retrieved file is routed along the search path or through IRA if the route expires.

**Algorithm Used:** The following are the algorithm used to manipulate the content of the mobile node.

**Interest-Oriented File Searching and Retrieval:** In social networks, people usually have a few file interests and their file visit pattern generally follows a certain Distribution. Also, people with the same interest tend to contact each other frequently. Considering the relation among node movement pattern, individuals' common interests and their contact frequencies, we can route file requests to file holders based on nodes' frequencies of meeting different interests.

Then, the interest-oriented file searching scheme has two steps: intercommunity and intercommunity searching. Anode first searches files in its home community. If the coordinator finds that the home community cannot satisfy a request, it launches the intercommunity searching and forwards the request to an ambassador that will travel to the foreign community that matches the request's interest. During the search, a node sends a message to another node using the interest-oriented routing algorithm, in which a message is always forwarded to the node that is likely to hold or to meet the queried keywords.

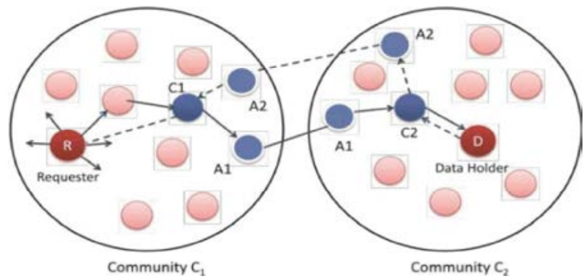


Fig. 7: File searching in SPOON

**Interest-Oriented Routing Algorithm:** This section concentrates on how the data is queried and rendered from the mobile node based in the community constructed on the condition of content of the mobile node.

**Intracommunity File Searching and Retrieval:** 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. Since the query is initiated by users, term weights in vQ are constant values. In the intracommunity searching, the destination that a query is sent to is represented by a combination of the vQ and the node vector of the requester's community coordinator (vNC), represented by

$$vdest = \lambda vQ + (1 - \lambda)vNc. \quad (3)$$

**Algorithm 1:** Pseudocode of intracommunity file searching for query Q conducted by node Ni.

```
Procedure intraSearchForQ ()
     If a neighbor nb of Ni matches query Q then
     Ni.sendQeuryTo(Q, nb)

 else if Q.src = Ni then
     if Sim(vQ;vC) < Ts then Q.vdest? vNC
Ni.sendThroughIRATo(Q, NC)
else

Q.vdest ?vQ
Ni.rankNbByFitness() overallF? 0
     for each neighbor nb of node Ni do overallF gets
     overallF +F(Q;nb) Ni.sendQeuryTo(Q, nb)
     if overallF >β then break
else
     if Q.hops < MaxHop then Q.vdest? vQ
     Ni.rankNbByFitNess()
     nb ?the neighbor with maximal fitness
Ni.sendQeuryTo(Q, nb)
else
     Q.vdest ?vNC Ni.sendThroughIRATo(Q, NC)
```

**Intercommunity File Searching and Retrieval:** In 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 intracommunity search step, the coordinator also uses the multicopy 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.

**Algorithm 2:** Pseudocode of intercommunity file searching for query Q conducted by node Ni.

```
     Procedure interSearchForQ () if Ni is a
     coordinator then
      bContain ?Ni.checkContainFile(Q) if bContain
     Ni. sendQeuryToDes(Q)
else

     Ni. rankAmByMatch() overallS? 0
     for each ambassador NA of Ni's community do
     n.sendQeuryTo(q, NA) overall S?overall S + Sim
     (q:VQ,NA.VC) if overallS > α break
     if Ni is an ambassador then when Ni meets
     another node Nj
     if Nj. homeCommunity = Ni. foreignCommunity
     then
     Ni. sendQeuryTo(Q, Nj)
     Nj. sendThroughIRATo(Q, NC)
```

**Information Exchange among Nodes:** We summarize the information exchanged among nodes in SPOON. In the community construction phase, two encountered nodes exchange their interest vectors and community vectors, if any, for community construction. In the role assignment phase, nodes broadcast their degree centrality within their communities for coordinator selection. When the coordinator is selected, the coordinator ID is also broadcasted to all nodes in the community. Then, each node reports its contact frequencies with foreign communities to the coordinator for ambassador selection. Besides, when a node meets a coordinator of its community, the node also sends its updated node vector to the coordinator to update the community vector and retrieves the updated community vector from the coordinator. When an ambassador meets the coordinator of its community, it reports the community vectors of foreign communities to the coordinator. After above information exchange, two encountered nodes exchange their node vectors and history vectors for packet routing. Each node Checks packets in it sequentially to decide which packets should be forwarded to the other node based on the file searching algorithm. Further, when network turns to be stable, the frequency of information exchange for community construction and node role assignment can be reduced to save costs. A query in a local community for a file residing in a remote community is forwarded through the coordinator of the local community. A community ambassador aids in retrieving the file that is in other community.

**Request-Completion and Avoiding Occurrence of Loops:** Given a file query, there may exist a number of matching files in the system. A node can associate a parameter Smax with its query to specify the number of files that it wishes to find. A challenge we need to handle is to ensure that the querying process stops when Smax matching files are discovered when multicopy forwarding is used. To solve this problem, we let a query carry Smax when it is generated. When a query finds a file that matches the query and is not discovered before, it decreases its Smax by one. Also, if this query is replicated to another node, Smax is evenly split to the two nodes. A query stops searching files when its Smax equals0. When a query needs to find more than one file, it is likely that IRA would forward a query to the same node repeatedly. To avoid this phenomenon, SPOON incorporates two strategies. First, the query holder inserts its ID to the query before forwarding the query to the next node. Second, a node records the queries it has received within a certain period of time. The former method avoids sending a packet to nodes it has visited before, while the latter method prevents sending different replicas of the same query to the same node. Specifically, when a node, say Ni, needs to forward a query to a newly met node Nj based on IRA, Ni checks whether the query's record of traversed nodes contains Nj. If yes, Ni does not forward the query to Nj. Also, when a node receives a query, if the query exists in its record of received queries, the node sends the query back to the sender. These two strategies effectively avoid searching loops by simply preventing a node from forwarding the same query to nodes that have received the query before.

**Node Churn Consideration:** In SPOON, when a node joins in the system, it first finds the communities it belongs to and learns the IDs of community coordinators and then reports its files and utility values to the community coordinator when encountering it. This enables the coordinator to maintain updated information of the community members. A node may leave the system voluntarily when users manually stop the SPOON application on their devices. In this case, a leaving node informs its community coordinator about its departure through IRA. If the leaving node is an ambassador, the coordinator then chooses a new ambassador. If the leaving node is a coordinator, it uses broadcast to notify other community members to select a new coordinator. A node may also leave the system abruptly due to various reasons. Simply relying on the periodical beacon message, a node cannot tell whether a neighbor is left or is just isolated from itself, which is a usual case in MANETs. To handle this problem, each node records the time stamps when it meets other nodes and sends it to the coordinator through IRA. The coordinator receives this information and updates the most recent time stamp of each node seen by other nodes. If the coordinator finds that a node's timestamp is more than

Tx seconds ago, it considers this node as a departed node. When a node finds that more than half of community members have found that the coordinator has left, it broadcasts a coordinator reelection message to select a new coordinator instead of the node which has left.

**Evaluation:** We first evaluated the system using JAVAFX and MYSQL. A virtual environment is created using JAVAFX. SPOON categorizes nodes having similar interests and frequent contacts into a community to facilitate interest-based file searching. A node is created by the user by inputting its distance and mobility. The nodes that have similar type of keywords based on a particular domain are categorized as a community. A node can have more than one file belonging to different categories, in such cases we have categorized them by considering the highest probable category as its community. Our system assigns the coordinator and ambassador position to a node based on its mobility. This can change accordingly based on the available nodes at that instant. The node which is in need of a file inputs 4 keywords and selects the type of file and queries for it. The query is forwarded to the coordinator of its community. If the coordinator finds the file within its community, the file is retrieved back to the requested node. If not, the query is forwarded to the ambassador of the community. The ambassador searches for the matching files based on the keywords given in other communities. If it finds the file, it is retrieved back to the requested node. The user node that has requested can either save or view the file which it has requested. If both the coordinator and ambassador could not find the file, then the query fails after retries.

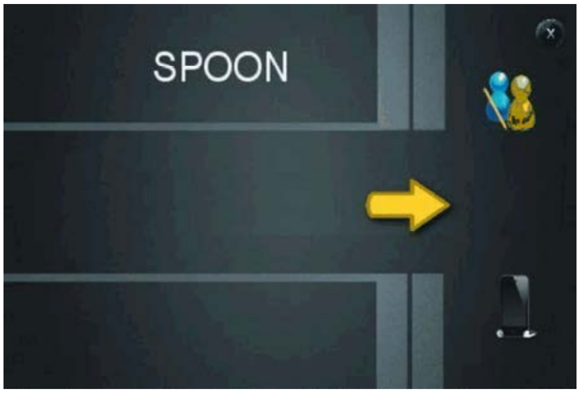


Fig. 8: Main page

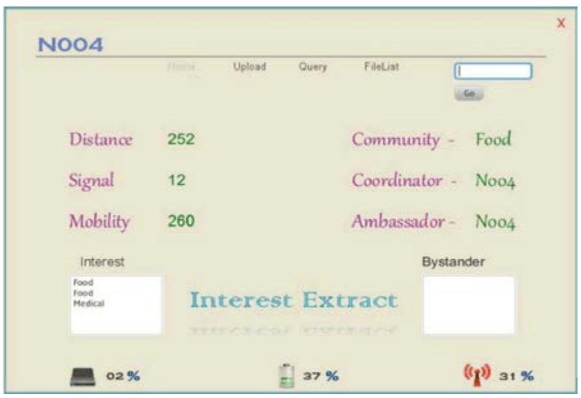


Fig. 9: Node Created

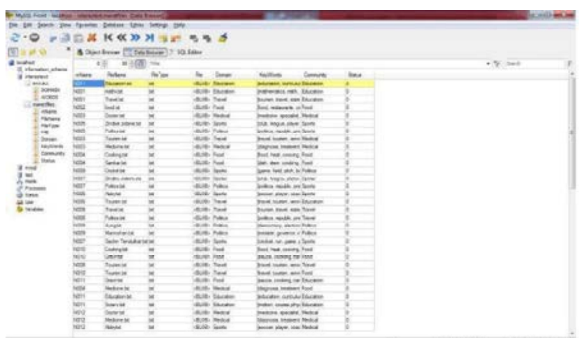

Fig. 10: Database


Fig. 11: Monitoring Screen

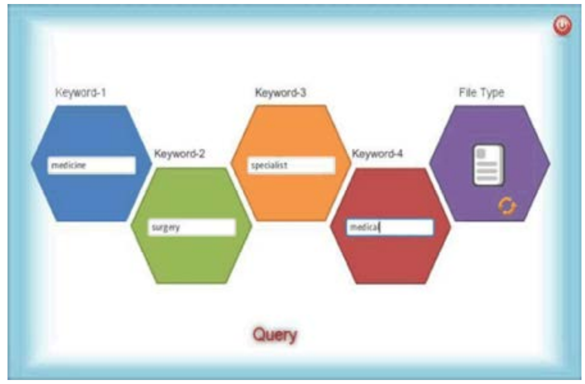


Fig. 12: Searching through keywords


Fig. 13: Query forwarded to Coordinator and Ambassador

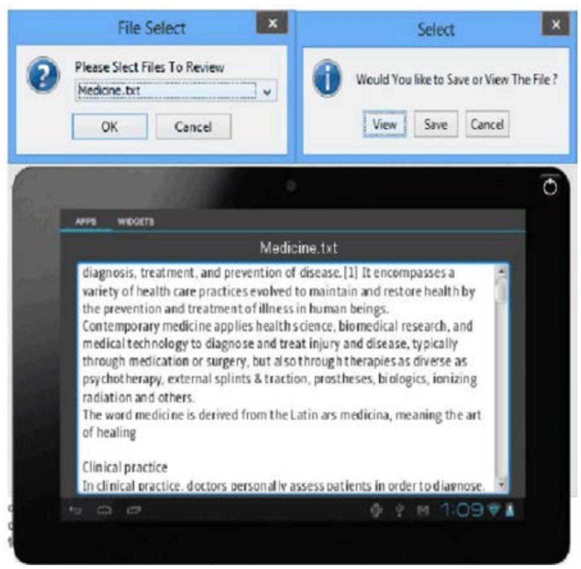


Fig. 14: Content extracted from node

## CONCLUSION

We explored a Social network based P2P content file sharing system in disconnected mobile adhoc Networks(SPOON). SPOON considers both node interest and contact frequency for efficient file sharing. We introduce four main components of SPOON: Interest extraction identifies nodes' interests; Community construction builds common-interest nodes with frequent contacts into communities. The node role assignment component exploits nodes with tight connection with community members for intra-community file searching and highly mobile nodes that visit external communities frequently for inter-community file searching; the interest-oriented file searching scheme selects forwarding nodes for queries based on interest similarities. SPOON also incorporates additional strategies for file prefetching, querying-completion and loop-prevention and node churn consideration to further enhance file searching efficiency.

## REFERENCES


1. Kang Chen, 2014. Leveraging Social Networks for P2P Content-Based File Sharing in Mobile Adhoc Networks, pp: 235-249.
2. A Report on "The State of the Smartphone Market," Retrieved on Feb 4, 2014 from http://www. allaboutsymbian. com/news/item/6671_The_State_of_the_Smartphone_Ma.php.
3. A Report on "Next Generation Smartphones Players, Opportunities & Forecasts 2008-2013, Juniper Research, 2009, RetrievedonFeb11,2014from http://www.juniperresearch.com/shop/products/whitepaper/pdf/Whitepaper%20~%20Smartphones.pdf.
4. A Report on A Market Overview and Introduction to GyPSii, Retrieved on Feb 12, 2014 from http://corporate.gypsii.com/docs/MarketOverview, 2013.
5. Technical Description of Bit Torrent protocol retrieved on Feb 4, 2014 from "http://www. bittorrent. com, 2013.
6. Technical Description of Kazaa retrieved on Feb 4, 2014 from "Kazaa," http://www.kazaa.com, 2013.
7. McPherson, M., 2001. Birds of a Feather: Homophily in Social Networks, Ann. Rev. Sociology, 27(1): 415-444.
8. Tchakarov, J.B. and N.H. Vaidya, 2004. Efficient Content Location in Wireless Ad Hoc Networks, Proc. IEEE Int'l Conf. Mobile Data Management (MDM '04).
9. Papadopouli, M. and H. Schulzrinne, 2001. 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.
10. Hoh, C. and R. Hwang, 2007. P2P File Sharing System over MANET based on Swarm Intelligence: A Cross-Layer Design, Proc. IEEE Wireless Comm. and Networking Conf. (WCNC '07), 2674-2679.