International Journal of Soft Computing 9 (5): 285-297, 2014
ISSN: 1816-9503

A Resource Search and Download Protocol for Peer to Peer Ad Hoc Networks

D. Jayashree and P. Yogesh
Department of Information Science and Technology,
Anna University, 600 025 Chennai, India

Abstract: Sharing digital resources among the peers in a virtual overlay network places stringent constrains in the underlying physical network. In this research, the underlying network is a Mobile Ad hoc Network (MANET). Hence, constrains on the network resources are even more stringent. In a mobile Peer to Peer (P2P) network, the resource transaction has to take place through mobile devices. The mobility of the nodes causes higher delay and overhead in the process of route discovery and retrieval of resources in P2P networks. In this research, researchers propose a resource accessing scheme to enable continuous resource retrieval by the mobile devices and the scheme is based on a multicast protocol that is deployed in the overlay network. Groups are created based on the content and a cross layer approach is followed between the network layer and application layer to reduce the search latency and to make the overlay network a proximity aware one.

Key words: Mobile ad hoc network, peer to peer network, gnutella, cross layer, cluster, resource sharing

INTRODUCTION

A peer to peer overlay network is a computer network which is built on top of another network. Nodes in the overlay can be thought of as being connected by virtual or logical links, each of which corresponds to a path, perhaps through many physical links in the underlying network. A mobile peer to peer network is one in which the underlying network is a Mobile Ad hoc Network (MANET) and the resource sharing among the mobile devices can be carried out in a distributed way without any coordinator who controls the transactions in a centralized way. Peer to peer networks and MANETs do share some similarities such as decentralization, self-configuration and self-healing; combining both networks could provide a solution for various purposes such as data storage, data sharing, information retrieval and data dissemination. However, this combination poses great challenges because these networks operate on different layers (peer to peer network on application layer and MANET at the network layer). Moreover, more attention is needed for routing in the overlay network and maintaining the overlay network with adaptation to the underlying network. Hence, the simple deployment of common P2P protocols on top of MANETs is inefficient and does not scale well. The main reason for these limitations is due to the fact that the virtual overlay network does not match the frequently changing physical network (MANET). The absence of clear routing protocol confirms the necessity for cross layer approach for the best interaction between layers.

Mobile peer to peer networks allow multiple users distributed across a large geographical area to share and make use of the resources of one another in a dynamic environment. Accessing the resources in the mobile environment is appealing and challenging since users are permitted to access the resources continuously irrespective of their mobility. The major challenges are increased latency and overhead. In this research, researchers face the challenges by incorporating features like proximity based resource groups, content based group routing and multicast routing mechanism in the overlay network. Moreover, researchers implement these mechanisms in the system in such a way that all these mechanisms are able to co-ordinate among themselves without the need for centralized coordinator. The major resources that are being shared among the peers are text files, audio files, video files, etc. Searching the whole network by flooding would result in unnecessary zigzag routes and congestion in the network. In unstructured peer to peer networks, to reduce the congestion and improve the efficiency of the routing mechanism, researchers propose a selective search mechanism for easy accessing. To further improve the efficiency, content aware cluster is formed by grouping peers of the application layer when constructing the overlay network. After the construction of the overlay network multicast routing algorithm is applied for distributing the resources in a timely and efficient manner. The proposed mobile peer to peer protocol employs efficient resource based search and cross layer communication to decrease the overhead as far as possible and to match the virtual P2P topology with the physical topology of the MANET.

Corresponding Author: D. Jayashree, Department of Information Science and Technology, Anna University, 600 025 Chennai, India

In this research, for routing in the underlying MANET researchers have chosen a modified form of Dynamic Source Routing (DSR) protocol by name Link State DSR (LSDSR) (Hussain, 2006) and the justification for this selection is explained. For the formation of multicast groups and the retrieval of resources we propose a protocol by name Mobile Peer to Peer Resource Sharing Protocol (MPRSP). A part of multicast protocol called Protocol for Unified Multicasting (PUMA) has been adopted in the research for multicast routing. Since, researchers go for a cross layer approach, LSDSR and MPRSP interact between each other to make the formation of multicast groups, retrieval of resources and multicast routing more effectively.

LITERATURE REVIEW

P2P protocols allow peers to connect directly to each other without any interference from a central server. This enables many kinds of P2P applications. First, P2P networks can be used for example for trafic-intensive file sharing (e.g., Gnutella (Schollmeier *et al*., 2002), BitTorrent (Izal *et al*., 2004)). Since, the heavy trafic is between the peers and there is no central server limiting the total trafic in the network. There is P2P Internet telephony (Skype) (Baset and Schulzrinne, 2006) that is analogous to the previous application. Here, the voice or video streams correspond to the files. P2P networks can be used also for less trafic-intensive information search and sharing purposes. Traditional overlay ad hoc network protocol is not compatible to a digital resource search application and cannot be used to build the Gnutella logical network on mobile network for the protocol. Cross-layer approach is implemented in ORION (Klemm *et al*., 2003) the search algorithm dependent on the search locality in the system. The MPP protocol (Schollmeier *et al*., 2003) announces itself to the routing protocol and a search request is broadcasted to all nodes. Any receiving node will process the search request even when it does not satisfy the request. This makes the search overhead increases according to the degree of each peer and transferring behaviour in MPP can involve many nodes in the process including intermediate peers which do not belong to the same group.

The designers of CAN proposed by Saroiu *et al*. (2002), a landmark-based placement for accounting for the underlying topology when creating CAN. Each joining node probes a set of landmark machines, estimates network distances by measuring the round-trip-time and orders the RTTs in increasing order. The quality of the link could not be captured using only the RTT; path optimization is therefore affected.

Note that even though the underlying substrate is accounted for the reflction of the underlying topology on the overlay is partial due to the incomplete knowledge of the underlying network. It has been shown by Royer and Perkins (1999) that the selfish behaviour of overlay routing has significant conflict with the traffic engineering efforts of the ISP.

The multicast protocols which are used in wired networks such as DVMRP (Waitzman *et al*., 1988) and CBT (Ballardie *et al*., 1993) cannot be directly applied in the mobile environment as the multicast tree must be readjusted when node connectivity changes; this leads to partition of network with extra delay. Some protocols are proposed to handle the mobility for example MAODV (Royer and Perkins, 1999), AMRIS (Schollmeier *et al*., 2002), ODMRP (Lee *et al*., 1999) and MCEDAR (Sinha *et al*., 1999). MAODV and AMRIS fall into the tree-based multicast protocol category. In tree-based multicast protocols, only a single path between the source and the receiver exists. It provides increased efficiency and scalability. ODMRP and MCEDAR fall into the mesh-based multicast protocol category. Protocols in this category provide robustness due to the availability of multiple paths between the source and receiver. Mesh-based protocols maintain a high delivery packet ratio. Recently, a number of multicast protocols-AMRoute (Xie *et al*., 2002), ALMA (Ge *et al*., 2004) and PAST-DM (Gui and Mohapatra, 2003) have been proposed in order to perform multicasting in the application layer as an alternative solution to overcome certain network layer multicast drawbacks. AMRoute creates a bidirectional shared multicast tree using unicast tunnelling. Each group has a group leader who maintains all of the pertinent information related to the group. The key characteristic of AMRoute is its use of a logical mesh to create the multicast tree. Because of this, the tree does not need to be readjusted when the topology changes as long as the mesh has a recovery link. AMRoute does not prevent a temporary loop in the tree formation and suffer from non-optimal tree when mobility is present. PAST-DM also creates a logical mesh to connect all group members. It constructs a source-based tree Steiner in a centralized manner which may create a bottleneck in many nodes in the network because the source performs the creation of the multicast tree.

Finally, to the best of the knowledge, the Protocol for Unified Multicasting through Announcements (PUMA) (Hussain, 2006) is the most recent application layer protocol; it is a receiver-driven protocol and creates a logical tree between multicast members without logical mesh support. PUMA was designed to improve efficiency and ensure packet level reliability during tree reconfiguration.

GNUTELLA ARCHITECTURE OVER AD HOC NETWORKS

The success of a data-sharing P2P System lies on the ability to search for and retrieve data efficiently. The best way to search in a given system depends on the needs of the application. Some popular P2P applications are instant messaging, Multimedia Messaging Service, sharing photos MobShare (Sarvas *et al*., 2004) music sharing (Wiberg, 2002). Mobile payments (Zheng and Chen, 2003), etc. Content-based systems such as Gnutella are designed for resource sharing applications with richer queries. In content-based platforms, peers maintain a set of logical links among them which all together form the overlay network. Queries, query results and other control messages are all sent exclusively along overlay links, independently from the physical location of the involved peers. In principle, the graph topology of an overlay network may look significantly different from that of the underlying physical network as shown in Fig. 1. This depends on the way the peers discover themselves and on the policies used to choose and establish logical links. Clearly, this process has direct impact on the workload imposed on the underlying network as each logical link spans across an arbitrary number of physical hops.

Open platform like Gnutella distribute data without the need for central server as in Napster. With this context ad hoc networks can be considered in synergy with traditional P2P Systems which organize themselves into logical network and cooperate independently with each other for the availability of digital content. Hence, Gnutella has been chosen to meet the various requirements such as the ability to search and retrieve the resources and research in dynamic environment which are the basic activities of digital retrieval in movement. The Gnutella V6 network (Schollmeier *et al*., 2003), provides peer hierarchy classification as Leaves and Ultrapeers. Leaves are weak nodes with limited storage, processing power and network bandwidth while ultrapeers are powerful nodes. This study proposes a cross-layer optimization of the protocol to address the compatibility issues that arise due to routing that takes place in physical layer and overlay layer. The key idea is to exploit the topological knowledge collected by on-board routing agents at the network layer to simplify peer discovery procedure and enable a smarter construction of the overlay. For example, peers may favour the establishment of closer relationships, instead of peering with far away entities in forming the overlay. In particular, the main targets addressed by this research are:

- To decrease the network traffic (i.e., overhead) generated for overlay management purposes, making it stable against increasing mobility and node churns
- To improve the quality of the resulting overlay, making it closer to the underlying network topology

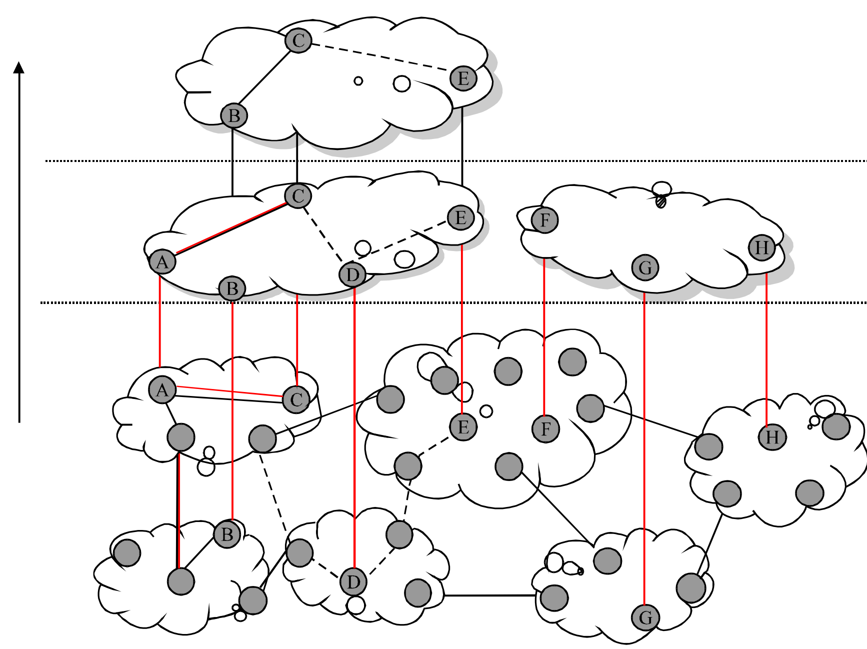

Fig. 1: Overlay with underlay network

The cross layer Gnutella provides the interaction between a peer's application layer and the routing protocol of network layer. The proposed cross layer wireless P2P network supports neighbor peer selection policy, based on the topology knowledge collected on the overlay and exploited at the network layer. Upon the reception of the events, the peer makes decision to add, drop and switch the connections in the multicast tree and finds the optimal path in terms of hop count.

Overlay formation with cross layer interaction: Gnutella was not designed for ad hoc networks and suffers from node mobility causing peers not to achieve minimum connectivity requirements. The protocol generates traffic bursts in correspondence of topological re-configurations. Researchers propose an alternative approach based on cross layering to overcome the incompatibility between overlay layer and network layer. To highlight the cross-layer interaction, researchers base the research from (Borgia *et al*., 2006) where the cross layer (XL) interface work has been introduced to expand protocol's interaction capabilities and possibly implemented in standard protocol stacks without spoiling the clean design (Fig. 2).

Employing a proactive approach, e.g., Destination-Sequenced Distance-Vector (DSDV) would likely involve computing several sets of routes and indexing them by time and the associated resource requirements is also large. On the other hand, reactive routing protocols are preferred in resource sharing systems where it is necessary for a node X to have the routing information about another node Y only when node X is in need of some resources from node Y. Moreover, from the literature also researchers find that due to their simplicity and inherent support for data on demand, reactive routing protocols have been the predominant design choice in wireless peer to peer networks. Barbosa e Oliveira *et al.* (2003) pointed out that DSDV protocol showed to be insensitive to the node speed whereas Dynamic Source Routing (DSR) was very sensitive to the node speed. DSDV was badly affected by the increase in the number of the network nodes as it requires periodic routing updates and broadcasting of triggered beacon messages. DSDV exhibits the most overhead followed by Ad hoc On-Demand Distance Vector (AODV) and then DSR. The proactive protocol for one query, introduced ten times more control packets than reactive protocol. Comparing DSDV to DSR on-demand protocol, the considerable increase in overhead obtained was due to route update messages that are constantly triggered by DSDV.

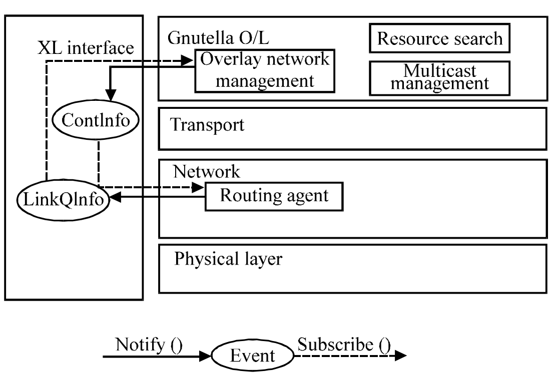

Fig. 2: Cross-layer (XL) architecture for Gnutella above LSDSR

A variant of DSR by name LSDSR could create an on-demand routing protocol by taking link quality into account. As ad hoc nodes are autonomous nodes, the discovery of other nodes in the system is based on the reactive routing protocol of LSDSR which has the mechanism to discover nodes based on their requirement and fill their routing tables based on nodes position. LSDSR is an extension of the DSR routing protocol that implements all the functionalities of DSR. LSDSR uses a link cache instead of path cache to efficiently maintain link information. The key idea behind calculating link-state information was to calculate the physical link delay. Researchers used this link delay as the metric for judging the link quality. Moreover, in LSDSR the resource information is added to the hello and response messages, in order to know the particular service offered by each peer and hence LSDSR provides the route for accessing the desired resources. LSDSR could reduce the routing down time and flooding, Hence, LSDSR not only improves the routing discovery time and link load balance but also optimizes the routing request message and reduces the packet loss rate (Table 1).

A Gnutella peer could exploit the extension on routing protocol and perform peer discovery in conjunction with route discovery at the network layer. Gnutella peers could make their demand through cross layer events along with routing agent. They spread their resource list together with control packets. From the routing agents the application on peers learns about the link quality such as delay, signal strength, liveliness of

Table 1: Cross-layer events

Events	Description
ContInfo (resource details)	Gnutella→LSDSR Message from overlay of local peer with the resourced details
LinkQInfo	LSDSR→Gnutella Message from network layer with local peer's link quality

the link, etc. To outline the link quality of a wireless link, researchers use latency into account for each link. Whenever a peer needs to find a resource peer it makes use of the LSDSR which in turn makes use of the LinkQInfo events to which the routing agent subscribes. The key idea behind calculating link-state information is to calculate the physical link delay. Researchers use this link delay as the metric for judging the link quality. ContInfo events to which the local Gnutella peer subscribes is used to inform about the resources to other peers. A ping messages generated by the peers that are accompanied with the ContInfo event help the peers to know the resource content list available with other peers. These events received from other peers are used to notify the local peer about the resources of these peers, received together with a routing control message. The proposed Gnutella overlay formation method could minimize the overhead since the LSDSR initiate the route search reactively. When mobility is high, the cached routes may not reflect the current topological status. LSDSR routing handles this situation by obtaining fresh routes for each search.

Gnutella logical network formation which defines four node states (full, stable, connecting and idle) that would help to maintain the connecting degree within the bound. Each peer maintains a neighbouring table in order to store the peer to peer connection in the overlay network. In the scheme, researchers enforce limits on the connection established with neighbours. The maximum limit of the neighbour table is denoted by Upper Limit (UL). Researchers also decide to have a lower limit to maintain a minimum number of connections denoted by Lower Limit (LL). Each node's state is decided based on the current size of the neighbouring table. Initially, all peers are in Idle state; the neighbouring table size is zero. Once a node has a neighbouring table size equal to UL, it reaches the full state; maintain its available logical connection. No acceptance of new connection or initiate a peer discovery process. When the neighbouring table size is between one and LL, the node is in connecting state. During the state the peer periodically runs the new connection discovery process and accepting the new connections in order to reach the stable state. If possible, the peer converts the state into full state. If the peer state is stable that is the neighbouring table size is between LL and UL, the peer can decide to move to full state by running the peer connection discovery process. Each peer maintains a minimal amount of connectivity (i.e., a Lower Bound LB) without overdoing it (i.e., exceeding an Upper Bound UB) and consequently not abusing network resources as suggested by Castro *et al*. (2004).

Ping, pong, query and query hit are the crucial messages for Gnutella operation. Ping is used to discover peers on the network. A peer receiving a ping message sends one or more pong messages. A pong message contains information on a peer. When a peer receives a pong message, it stores the obtained peer information in its pong cache and tries to make connection to the peer. Each entry in the pong cache corresponds to one pong message. The number of pong messages generated in response to a ping message is the number of entries in the pong cache of the responding peer. Having random connections with other peers results in routing inefficiency. To address this problem, the system introduces a hierarchical structure with ultrapeers and leaves. A leaf keeps only a small number of connections with ultrapeers. On the other hand an ultrapeer maintains many leaf connections as well as a small number of connections to the other ultrapeers (Singla and Rohrs, 2002). It acts as a gateway to the Gnutella network for the peers connected based on their content and shields these peers from the majority of message traffic.

A pong cache mechanism and link policy selection are integrated in this schema to achieve better performance in the Gnutella logical network formation. The term "node" as used in this study, is peer apart from the overlay formation and the resource discovery processes, researchers leave unaltered the rest of the Gnutella protocol.

Peer discovery and link selection: Ad hoc networks can be considered a computational duality with P2P Systems where peers self-organize in logical networks and cooperate independently to make libraries of digital content available for sharing. Many of the protocols need some initial bootstrapping. This usually means the node wishing to join the P2P network need to know at least one node that already belongs to the P2P overlay. Some protocols use a pre-configured server in the internet to fetch an up-to-date list of nodes in the P2P network. However, MANETs can't guarantee global connectivity and since P2P protocols used in MANETs must offer other means for nodes to join the overlay. When a peer wants to enter the Gnutella network, it initially connects to one of the several existing Bootstrapping peer that would in turn give the list of ultrapeers ID.

Most of the existing P2P networks assume the availability of a list of "well-known" addresses which are usually online or that a cache of existing addresses can be accessed from a centralized source such as a web server. All MANET nodes whether or not they belong to an overlay will cache this response in the event that they wish to join the overlay in the future. After a sufficient time, the original requesting node will have one or more responses to select from. It can then connect to a set of neighboring peers. The process of selecting the specific peer to connect is the function of the overlay formation. In an independent model, an overlay network is viewed as a weighted graph where the edges are overlay links weighted by delay.

The difficulty in designing a neighbour selection method is that each peer only knows information of a certain number (not all) of peers in the system. When peers join and leave it causes traffic overhead to transmit the information which changes according to the movements of nodes. The peers are reshuffled by mobility and peer failure. On receiving the pong with event message the peer fills the resource table. The resource table are updated with pong messages to reduce the need for searching the required content peer. From the periodic updates, the lower layer would provide the accurate delay information. This topology information enhances the resource table and used to form an overlay formation network. This information helps to select the link based on delay and decides the link selection policy. According to the policy, the closer links are favoured than the farthest link and help to build the overlay closer to underlying network. This deterministic selection of overlay links is meaningful in dynamic environment. If a peer is not willing to accept any new connections, it simply ignores the request. The increase in delay could be brought down by reducing the search span within the multicast group. The remaining Gnutella protocol activity is unchanged. The rest of the protocol's link probing and query messages are like the basic protocol.

The new node creates an event message containing the preferred node based on the connectivity with the resource group and sends it to the routing agent through the cross-layer interface. The link delay is measured at the network layer so that the maximum reliability of the connection can be established in the logical network; it must then be selected in the multicast tree. The routing agent notifies the nodes about the length path in a control message. The receiving node selects the optimized path in order to be added to the multicast tree.

Multicast management phase: In this study, researchers present an overview of the Completely Distributed Multicast algorithm for the resource group which is based on proximity tree formation. The message classes in the Gnutella protocol are divided into two major modules. The communication protocol module represents the basic Gnutella messages. The multicast protocol module implements all of the components related to the MPRSP protocol. Multicast group manangement algorithm 1 as:

Algorithm 1: Multicast group manangement:

```
MulticastGroupManagement.(Mem,MessageType,ID)
Switch
Case      // Peer joins Multicast Group
          MessageType =JOIN :
          NewMem ←Mem
          NewMem.Parent ← FINDCLOSESTPEER(NewMem)
          While
          New mem. Parent. Degree = UpperBound
          Do
          ADDCHILDREN (NewMem.Parent.ChildList, NewMem)
          Else
          NewMem.Parent ← FINDCLOSESTPEER(NewMem)
          REFRESHPREDECSSOR (NewMem, ID)
Case  // Peer Leaves Multicast Group
          While
          MessageType =LEAVE:
                    If LeavingMem ← CoreMem
                    then
                              for Child ← 1-N
                              do
                              BROADCASTCOREELECTION (Mem,ID)
                              Else
                              LeavingMem ← Mem
          do
     REMOVECHILDREN (LeavingMem.Parent.ChildList, LeavingMem)
          While (LeavingMem.ChildList)
          do
                    for Child ←1 to N ε LeavingMem.ChildList
                    do
                    Child.Parent ← FINDCLOSESTPARENT
                      (Child.ClosestPredecessor ,Child.Closest
                    Predecessor. ChildList)
                    ADDCHILDLIST   (Child.Parent.ChildList,Child)
                    UPDATEPREDECESSOR (Child,ID)
Case default:
       If (Mem.Hello>TimetoLive)
       then MulticastGroupManagement.(Mem,Leave,ID)
```

A new node selects its parent link in the multicast tree based on the node reliability and proximity. The peer obtains information about all neighbours from the Gnutella logical network. Peer selects preferred connections based on their link stability collected from overlay network and connected as multicast node members. Communication among nodes (peers) is based on PUMA multicast protocol. The data messages pass most likely through peers which belong to that particular resource group. Researchers extended the protocol for Multicast algorithm on Gnutella platform to achieve the following objectives:

- To determine an adjacent link with the best capacity to form the multicast tree even with lack of global topology information
- To restructure a tree in the event of topology change such that only one link leads to one group connectivity so that no cycles are introduced and hence the tree topology is preserved
- To improve features like self-organization and self-healing, especially against typical ad hoc situations like network partitioning

Multicast group formation: In this study, the multicast tree formation and maintenance are discussed. Groups are based on their particular resources and each peer may be a part of more than one group according to its willingness to share many types of resources. The multicast members are communicated within the group in flooded fashion. Each member has to find the parent based on the link quality which has been collected during periodic updating of link status.

The union of the source initiate routes forms the shortest path tree for our overlay multicasting. The core keeps track of the multicast tree state using their Childnode Tables. These tables are used to localize multicast group member ship management protocol to reduce the maintenance overheads and the impact of reconstructing the multicast tree due to node churn. Node churn problem refers to the continuous process of node arrival and departure in distributed application. When node churn occurs, they rely on the local topology maintenance control in the overlay network to repair the overlay. Researchers also invoke a local ancestor recovery protocol to localize repair and self-healing of the multicast tree. In this manner, group membership operations are decentralized and managed efficiently.

Grouping criterion: Digital resources are classified in one or several categories (e.g., text, music, movies, etc.). A peer can share resources classified in various categories. Based on this information, peers self-organize into clusters by joining or leaving different parts of the overlay. A physical node may join several clusters as a logical node. A logical node represents the presence of a physical node inside a cluster. First, researchers want to increase locality of related content by connecting the resource content peer to form a cluster. Researchers frame the group with same type of digital resource to be shared. Forming the group based on certain resources can help to minimize the search latency and reduce the overhead incurred by maintaining the overall network peers.

The peer keeps the delay as the metric whenever the route request has been initialized. Link updation of LSDSR protocol can help to form the link table with delay as link quality. Each group has a group leader called core of that group which would keep group information and guide a receiver to that group. The core node is being selected based on the number of files available for that resource type. The peer holding maximum number of same type of resources will be the core and if the two or more have same count then the smallest ID peer will be the core. Each node members have a maximum number of children depending on their capacity. When node reaches the maximum number of connection (satiated state) it stops accepting multicast connections. Core peer's periodic broadcast announcement helps the peers to receive the multicast messages with control packets. The Ultrapeer which has been connected to the group will have the updates of the core. Ultrapeers can be part of the Gnutella network and act as gateway peers which play a gateway role for mobile device in cluster management and improve the quality of search application. An ultrapeer forwards a query to a cluster member only if it knows the core peer contains the resource holding peer. Peers never replay back the queries to ultrapeers. Each core peer keeps track of the multicast tree state using their childnode tables to reduce the maintenance overheads and help in node churn. In the following study researchers explain the join and leave method which would stabilise the node in node churn.

Peer joining and leaving group: Leaving and joining processes involve the message exchange that happens between the peer and the core. Here, when the peer joins the multicast group, the core has been reached by having the core ID collected from the ultrapeers. The joining peer sends the Join request to the multicast group with group ID. The receiving peer replies with GAct (group accept) or GDcen (group decline) message. The closest group member will be selected in the multicast tree to join. The GAct message carries the acceptance of the receiving peer to allow the requesting peer to establish a link and parent to the new peer and update its childnode tables with this new peer. The receiving peer will obtain the addresses of the list of predecessor nodes from the core to its parent. This way every member will maintain the list of predecessors to reduce the service interruption probability. The link latency property which the overlay network formed allows the peer to select a parent with smaller distance.

Here, researchers establish limitation of upper and lower bound on connectivity. If the peer reached the threshold of upper bound level for degree it sends GDcen message and childnode tables to the new peer. If the new peer receives GDcen message, it moves to the next preferred link to repeat the process. If the requesting peer is interested, it could make use of the child list for link connectivity. Otherwise it starts with other peers as the overlay formation is based on using some constraint to form the tree in optimized manner. All the messages are sent in unicast way to reduce unnecessary traffic. There is a possibility that the new peer may not receive any messages from ultrapeer. This situation conveys two possibilities. One is that would be no group formed based on that particular content. Another factor is that there is a break in the multicast tree which could not be reached by the ultrapeer.

Core peer selection process is informed to ultra peers which are all have communication with the core itself. Each group with a special ID along with their content ID can select a core for its group. With the information contained in such announcements, nodes elect cores and each node in the network learns of one or more routes to the core. Selected core send this CS (Core Selection) message to the ultrapeer. In case the ultrapeer receives two or more core for the same content type it will select the least ID core. In order to resolve group leader conflicts Ultrapeer sends CAcc (Core Accept) message with the selected core ID back to the core. It shows confirmed way of administrating the whole dynamic network.

To leave the multicast group the member sends a leave message to the parent and all the group membership information of this peer will be removed from the parent's childnode list. The parent node updates the information, removes the node from the multicast data and informs the core peer. The affected children will respond to this change by making a choice from its predecessor list. They will find the closest neighbouring peers from the predecessor list to rejoin. These neighbouring predecessors will provide the information about their children from their childnode list. With the latency information about the predecessor and their children, the closest one will be chosen as a parent. The parent childnode list will be updated with the affected children information and the children will update the predecessor list accordingly. This recuperation process will help the children and the parent peer to recover quickly from node churn in a highly dynamic environment.

The group members will be periodically updated. The parent node will periodically send hello message to their children. If there is no response for a period of time the parent assumes that child has moved out or dead. That subsequently initiates the repair process with the leave operation of the child and the repair will be done by the execution of recuperation process.

Whenever, there is a change in the member connection, only few join and leave message would reach the core. The join and leave messages are accounted for the stability of the tree at that particular peer. If this number exceeds certain threshold the multicast tree sourced at the peer is reconstructed. This is because some peers may be struck in unfair and unfavourable situation. The threshold is set by = N/log10 N for N ultrapeers. Maintaining the overhead messages of Join and leave within the limit of O (log N) we divide the ultrapeer with the log10 N. Whenever, the threshold is reached the multicast tree rooted at the core is reconstructed according to the algorithm. The existing trees are replaced by the newly constructed tree for the multicast data delivery. The multicast grouped framed at the Gnutella protocol at the application layer. All multicast communications are done using PUMA protocol.

CONTENT SEARCH AND DELIVERY

A peer which owns the content is called sender peer and that requesting any content is called requesting peer. Each peer establishes membership with their content group according to their content. During the bootstrap the ultrapeer might send the core ID of the content group available under its group table. The peer update the table according to the received core ID. A requesting peer prepares a request message Content Request (CR) for the content required by it. The requesting peer checks the group table for the core peer address of the requested content based group. With the address of the core the requested control message is send to the core of the group for content search. If the content group's core peer ID is not available, request message is sent to the ultrapeer for finding the group core peer ID for that particular content type. The ultrapeer sends the request to all the active ultrapeers for that address. Hence, researchers have chosen higher bandwidth and less mobile peer as ultrapeer. Researchers assume at each time there would be <100 requests that reached the ultrapeer for processing making the links traffic bearable. In case the requesting peer does not get any reply from the peer after some period of time, the CR message is repeated until a reply is received or the maximum retries are reached. The retries could be adjusted suitably.

The ulrapeers which receives CR forward the messages to the content group core for search. The core of the group, flooded the message to the members along with control message. The peers maintain a packet ID cache to drop duplicate data packet. When the owner of the content receives the CR request, it replies back with Content Reply (CRL) message. The ultrapeers do not receive any CRL messages from the requesting peer after forwarding the CR messages. This way the ultrapeer only help to establish the communication between the sender and requesting peer. It shows the ultrapeer does not involve the content transaction. The CRL packet carries metadata regarding the requested peer ID, content name, number of pieces, size of pieces and bitmap. The core peer will decide how the content will be sliced, representing a content image using a bitmap like each bit represent a slice of the content. As researchers are on the process of transferring digital content creating a digital image could be done at sender side. Here, the data content size has been decided by the core of the group. Each group carries the maximum data packet size based on the link characteristic messages received from their group members. It is beyond the scope of the research. Hence, researchers fixed different sizes ranging from 500-1000 kb for different groups for the simulation.

In case CRL packet comes from many requesting peers then the selection should be based on less delay peer. CRL packet carries the shortest path from the sender peer to the requesting peer. From that data, the requesting peer ranks it and selects the least delay peer. Researchers have adopted this approach because the underlying routing is a reactive routing protocol and the delay can reflect the physical distance travelling time at the current period in the dynamic environment. This method would help the requesting peer to consider the next option when there is any communication error happens to the selected peer. It also reduces the overhead incurred by flooding the network with CR and CRL packets.

Once the sending peer receives the request for the content it checks the metadata and set the no of pieces for transmission. The data transmission size is set accordingly. The data is received through multicast announcement from the core node. The latest packet sequence number will keep it fresh. The requesting peer sends Agreement Message (AM) to the sender peer for giving consent for data transmission after receiving the CRL packets. When the selected owner receives the AM packets it starts its transmission and after finishing its transmission it remains silent indicating the stateless characteristics of the protocol. A timer is set for every fragment of content received by the requesting peer. The timeout indicates transmission error has occurred. The requesting peer now waits for another time period for the left out fragment. In case there is no reply the peer goes for recovery mechanism which is explained.

Recovery mechanism: The timely recognition of link failures is necessary to avoid delays and unnecessary data transmissions. It uses feedback from the link layer as the second building block for an efficient file transfer. Feedback can be provided by link-layer notification if available, e.g. as in IEEE 802.11. Otherwise, recovery mechanism should make certain for the content delivery reliability. Normally, peer restarts a search process for the content. However, the peer could get the next sender peer's ID from the reply message. In case the peer rejected the next coming peer address after deciding the first reply peer as its sender peer, resends the content request messages for the remaining pieces of the content. This information comes from the bitmap image of pieces. After the transmission of the requested pieces, node still needing any missing pieces will request them via recovery mechanism once again.

In case the peer stores all the sender peers' address the option of getting the remaining pieces without a search process is possible. The peer could make use of this address to send a direct request to the next sender peer. This might reduce the message overhead caused by traffic generated between the core and the group members. Here, the remaining pieces only requested. Reduction in data packet overhead is certain.

PERFORMANCE EVALUATION

Researchers used the latest version of the Network Simulator (NS2 V.2.29) to examine the performance of the protocol. Researchers implemented MPRSP protocol and PUMA multicast protocol in the Network Simulator (NS-2) and run extensive simulation for typical scenario of Random Waypoint Mobility Model (RWPM) and linear mobility at constant speed. Researchers assume that maximum 100 nodes move in a network area of 1000×1000 m for the simulation. Transmission range considered of each node is 150 m. Each node in the network is initialized with the parameters: Node ID, X-coordinate, Y-coordinate, No. of resources, resource category. For simplicity, node ID is initialized with a unique number between 1 and 500. Initially, X and Y coordinate of each node is randomly set within the network area.

Researchers assume each node can have maximum 4 resources where each resource type is initiated with single capital letter; A: Audio, I: Image, T: Text and V: Video. Researchers are also considering that each category can have maximum 10 file with each file sliced into pieces of 500 and 1000 bytes. Resources are placed randomly chosen destination node for data transmission. The traffic of activated nodes is supplied from a CBR source with a packet rate 0.5 p sec^{-1}. The slow scenario is defined as having a pause time equal to 10 sec and maximum speed of 5 m sec^{-1}. The fast scenario is defined as having a pause time equal to 5 sec and a maximum speed of 20 m sec^{-1}. Pedestrian speed to normal vehicle speed and the MAC layer is set as the 802.11 MAC. The number of join retries is 3 and the maximum logical connection degree is 5. Each node is configured using the 802.11 MAC layer. Performance analysis is provided according to random distribution of five different group addresses between 100 nodes. The duration of the simulation is 500 sec. During this set of experiments simulation researchers use five classes of maximum group size (10, 20, 30, 40 and 50 nodes).

The analysis and measurement of the protocol are based on four primary metrics: multicast group size, average routing overhead for LSDSR, maximum delay, hit ratio and the logical connectivity degree in the tree. Researchers created also three main scenarios based on network mobility (static, slow and fast) in order to study the effect of mobility on MPRSP performance.

Impact of cross layer: The advantages of link delay realization at the application layer are clearly visible in the simulation results we presented. Researchers ran simulations to analyze the message overhead of each of the set of nodes. One of the simulation parameters that would have an impact on the overhead is the forwarding austereness of the route search messages.

In Fig. 3a and b the average overhead measured in kb is experimented for various mobility scenario for different maximum group size under the routing protocol of MPRSP and AODV, respectively. Researchers compare MPRSP result with AODV as underlying routing protocol. Figure 3a indicates the average network overhead

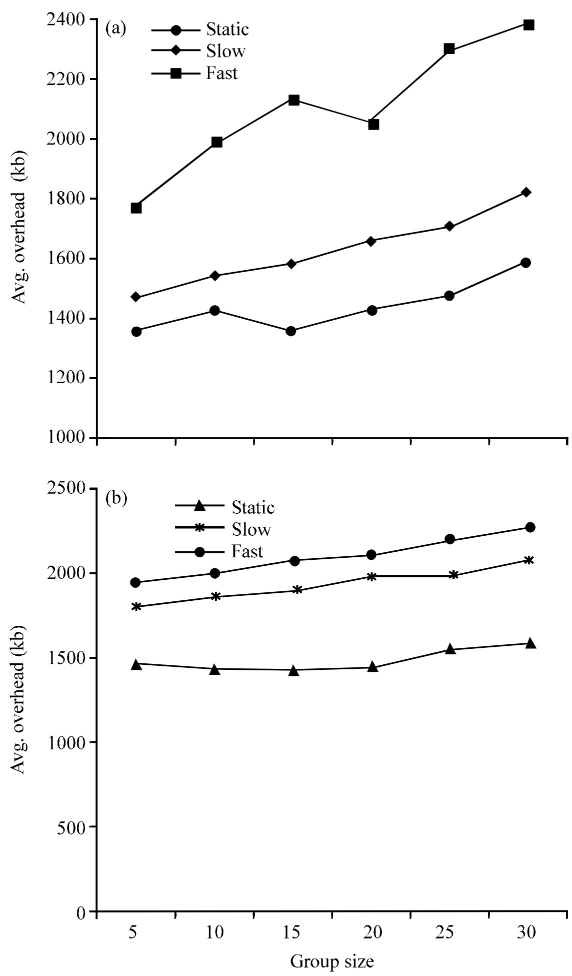

Fig. 3: a) Group size vs. average overhead of LSDS; b) group size vs. average overhead of AODV

produced by the MPRSP in three scenarios (static, slow and fast). The protocol shows increase in routing overhead in high node mobility scenario when the group size is large. This is because the network topology changes frequently, thereby causing extra overhead for link construction. LSDSR reduces the overhead of the Gnutella protocol in scenarios over 20 nodes when compared to AODV. In addition MPRSP performs well during the slow mobility scenario. The protocol provides better results in slow scenarios under an LSDSR routing protocol and shows an increase in overhead under fast mobility. The peer discovery process runs continuously, since the peers with less than lower-bound connection increases due to higher mobility.

MPRSP performance: In order to evaluate the effeteness of the resource aware clustered approach, scenarios that differ in speed and size are simulated. Three metrics are used to determine the efficiency of the MPRSP. The overhead would give the efficiency of finding the resource with less control messages. The overall traffic has to be low in order to enable an efficient search operation and to acquire the services of a peer irrespective of mobility. The ratio of successful query result is a metric to determine how efficient the protocol operates for the retrieval of resources. As the time required for a search and retrieval of requested resources may be important regarding usability, the latency is used as another metric. Therefore, the overall time between sending the request and receiving the object is measured and analyzed.

Control overheads: Figure 4 shows the mean value of control overhead consumed at the access links of the end

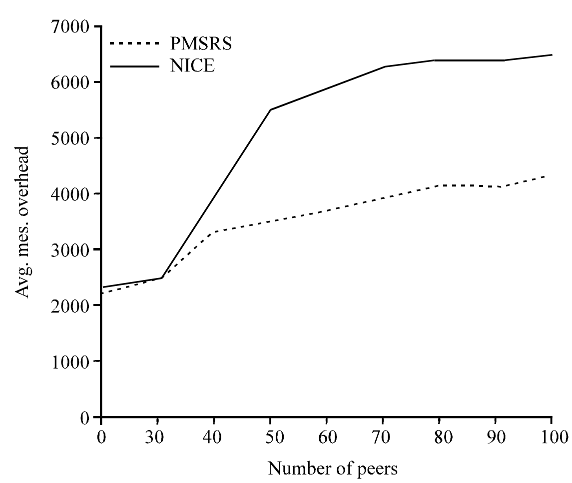

Fig. 4: Average control overhead vs. number of peers

hosts. Each symbol in the plot represents the average value of control traffic in form of kb sent and received by the group members.

The results of Fig. 4 show that the total number of messages increase slightly from around 2000 to around 4500 kb as the number of nodes change from 10-100. Obviously, the control overhead of NICE is higher than PMSRS. The simulation results again identify that NICE is only useful for small-size multicast group. Otherwise, the control overhead to maintain a large group will overwhelm the network resource. As the simulation results indicate, the system efficiently carries out file searches and it is also scalable. Hence, the proposed system becomes more bandwidth-efficient than NICE protocol of the same size. If the group size is very large the control overhead for such an operation is unavoidably high.

Hit ratio: The hit ratio plays an important role in the peer to peer network systems. The higher the hit ratio of a peer to peer network system is, the better the searching performance will be. However, researchers cannot merely consider the hit ratio separately. Users may feel impatient if a high hit ratio peer to peer network system always delays to reply the query hit.

Figure 5 shows how the hit ratio grows with four range of maximum group limitation of the peer to peer network systems. The hit ratio of clustered peer to peer network systems grows steadily with increase in group size. The hit ratio of larger group size of the peer to peer network systems is high at the less mobile speed and subsequently, it greatly improves the searching efficiency. Query messages could be replied in a very short period of time in this network system. The hit ratio of small group is lesser than the larger group in each speed.

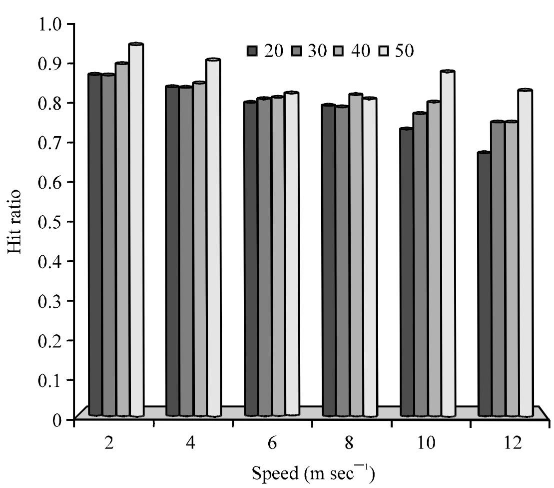

Fig. 5: Hit ratio vs. speed for different group sizes

Peer logical connectivity degree: Peer logical connectivity represents another important issue of the performance analysis. It is defined as the total number of children for each peer. Logical degrees have an important impact on the network overhead because nodes must make only one outgoing connection with nodes that belong to same group in order to avoid any loop formation.

The node stabilization depends on group size. Figure 6a and b indicate the logical node connectivity for each simulation scenario. Figure 6a shows in a slow mobility the degree is almost constant at small group size. It is due to the fact almost the nodes which are connected to the parent are stable. Protocol produces a better degree connection in slow scenario especially for large group size due to the availability of many nodes in the same group. Figure 6b gives the results of the same experiment run under fast scenario. Under the higher speed set-up the bigger group size have better connectivity than their less number of group size. Even though the peers mobility makes the change in their child connectivity, the construction of cluster with higher number of peers makes it possible to have higher number of child.

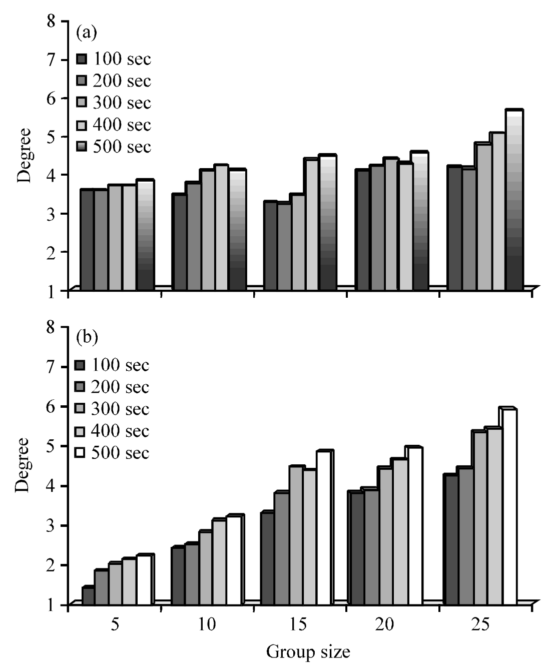

Fig. 6: Degree vs. group size for different simulation time; a) slow mobility scenario and b) fast mobility scenario

Average end to end delay: End to end delay is time that packet takes from source node to destination. In MPRSP, packet relays from several intermediate nodes. So, delay of a path is summation of all the links along that path. Link fluctuates during the mobility of nodes. Some links along path may have high delay comparing to others. Here, group size is limited to certain values to analysis the impact of mobility. Average value gives the value that can be compared with other results. Average end to end delay increases with mobility with or without cross layer implementation.

Protocol should minimize the longest path in the multicast tree in order to reduce the replicated multicast packets and the end to end delay. Figure 7 shows maximum delay for fast scenario and Fig. 7 for slow scenario using different group size. The graphs show a high decrement in delay with a cross-layer implementation compared to an ordinary implementation. This results from the fact that any multicast tree link is selected using low level metric in the network layer. Overall, researchers believe that the model provides satisfactory performance in estimating the routing overhead and brings deeper insight on how mobility affects the routing overhead. Results represent the average over 10 rounds for each scenario.

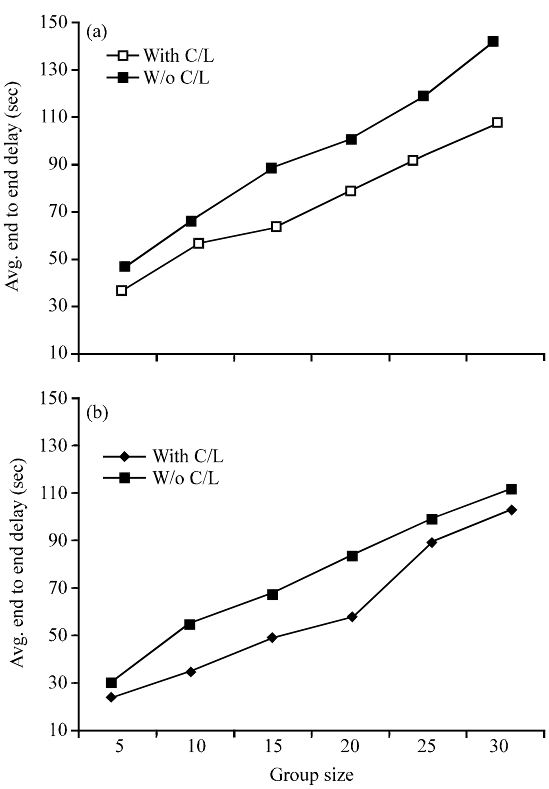

Fig. 7: Group size vs. maximum delay; a) fast mobility scenario and b) slow mobility scenario

CONCLUSION

In this study, researchers proposed a peer to peer system that enables file sharing in wireless ad hoc networks. The novel approach introduced in this research is the unification of resource search and routing functionality which results in a clustered peer to peer scheme. The system keeps track of the link information together with the local resource information which is fully distributed. In achieving this, researchers adapt technique of cross-layer scheme developed for peer to peer network as well as multicast routing approach. Simulation results showed that the system enables efficient access to shared files. However, it may not work efficiently when frequent disconnections occur. Nonetheless, researchers believe that the environment in which a file sharing system would be used is a ad hoc environment where the rate of mobility is not high. A conference room may be given as an example to such an environment in which attendees need to share files.

REFERENCES

Ballardie, T., P. Francis and J. Crowcroft, 1993. Core Based Trees (CBT) an architecture for scalable inter-domain multicast routing. Proceedings of the ACM SIGCOMM, October, 1993, USA., pp: 85-95.

Barbosa e Oliveira, L., I. Guimaraes Siqueira and A.A. Ferreira Loureiro, 2003. Evaluation of ad-hoc routing protocols under a peer-to-peer application. Proceedings of the Wireless Communication and Networking Conference, Volume, 2, March 16-20, 2003, IEEE, pp: 1143-1148.

Baset, S.A. and H.G. Schulzrinne, 2006. An analysis of the Skype peer-to-peer internet telephony protocol. Proceedings of the 25th IEEE International Conference on Computer Communications, April, 2006, Barcelona, Spain, pp: 1-11.

Borgia, E., M. Conti and F. Delmastro, 2006. MobileMAN: Design, integration and experimentation of cross-layer mobile multihop ad hoc networks. IEEE Commun. Mag., 7: 80-85.

Castro, M., M. Costa and A. Rowstron, 2004. Peer-to-peer overlays: Structured, unstructured, or both? Microsoft Research, Cambridge, Technical Report MSR-TR-2004-73. http://research.microsoft.com/apps/pubs/default.aspx?id=70077.

Ge, M., S.V. Krishnamurthy and M. Faloutsos, 2004. Overlay multicasting for ad hoc networks. Proeedings of the 3rd Annual Mediterranean Ad Hoc Networking Workshop, September, 2004, Turkey, pp: 201-209.

Gui, C. and P. Mohapatra, 2003. Efficient overlay multicast for mobile ad hoc networks. Proceedings of the IEEE Wireless Communications and Networking, Volume 2, March, 2003, USA., pp: 623-632.

Hussain, S.M.A., 2006. A link-state dynamic source routing protocol for Wireless Mesh Networks (WMNs). The Department of Computer Science and Software Engineering, The University of Western Australia.

Izal, M., G. Urvoy-Keller, E.W. Biersack, P.A. Felber, A. Al Hamra and L. Garces-Erice, 2004. Dissecting bittorrent: Five months in a torrent's lifetime. Proceedings of the 5th International Workshop on Passive and Active Network Measurement, April 19-20, 2004, Antibes Juan-les-Pins, France, pp: 1-11.

Klemm, A., C. Lindemann and O.P. Waldhorst, 2003. A special-purpose peer-to-peer file sharing system for mobile ad hoc networks. Proceedngs of the IEEE 58th Vehicular Technology Conference, Volume 4, October 6-9, 2003, USA., pp: 2758-2763.

Lee, S.J., M. Gerla and C.C. Chiang, 1999. On-demand multicast routing protocol. Proceedings of the IEEE Wireless Communications and Networking Conference, September 21-24, 199, New Orleans, LA., pp: 1298-1302.

Royer, E.M. and E.C. Perkins, 1999. Multicast operation of the ad-hoc on-demand distance vector routing protocol. Proceedings of the 5th Annual ACM/IEEE International Conference on Mobile Computing and Networking, August 15-19, 1999, ACM, New York, USA., pp: 207-218.

Saroiu, S., P.K. Gummadi and S.D. Gribble, 2002. Measurement study of peer-to-peer file sharing systems. Proceedings of the SPIE Multimedia Computing and Networking, Volume 4673, January 19, 2002, San Jose, CA., pp: 156.

Sarvas, R., M. Viikari, J. Pesonen and H. Nevalinna, 2004. MobShare: Controlled and immediate sharing of mobile images. Proceedings of the 12th Annual ACM International Conference on Multimedia, October 10-16, 2004, New York, USA., Pages: 724-731.

Schollmeier, R., I. Gruber and F. Niethammer, 2003. Protocol for peer-to-peer networking in mobile environments. Proceedings of the 12th International Conference on Computer Communications and Networks, October 20-22, 2003, IEEE, USA., pp: 121-127.

Schollmeier, R., I. Gruber and M. Finkenzeller, 2002. Routing in ad hoc and peer-to-peer networks a comparison. Proceedings of the Networking 2002 Workshops Pisa on Web Engineering and Peer-to-Peer Computing, May 19-24, 2002, Italy, pp: 172-186.

Singla, A. and C. Rohrs, 2002. Ultrapeers: Another step towards gnutella scalability. Lime Wire LLC, 1.0 Edition, November 2002. http://citeseer. uark. edu:8080/citeseerx/showciting;jsessionid=68E3F7AA40E00DE5D55B93A354242644?cid=313522.

Sinha, P., R. Sivakumar and V. Bharghavan, 1999. MCEDAR: Multicast core-extraction distributed ad hoc routing. Proceedings of the IEEE Wireless Communications and Networking Conference, Volume 3, September 1999, New Orleans, LA., pp: 1313-1317.

Waitzman, D., C. Partridge and S.E. Deering, 1988. Distance vector multicast routing protocol. Internet Experimental RFC 1075, November, 1988. http://dl.acm.org/citation.cfm?id=RFC1075.

Wiberg, M., 2002. Mobile peer-to-peer entertainment: Putting folk music back on the streets again when peer-to-peer goes mobile. Proceedings of the Mobile HCI 2002, September 18-20, 2002, Pisa, Italy, pp: 1-2.

Xie, J., R.R. Talpade, A. Mcauley and M. Liu, 2002. AMRoute: Ad hoc multicast routing protocol. ACM/Baltzer Mobile Networks Appl., 7: 429-439.

Zheng, X. and D. Chen, 2003. Study of mobile payments system. Proceedings of the IEEE International Conference on Electronic Commerce, June 24-27, 2003, IEEE Computer Society, Newport Beach, CA, USA., pp: 24-27.