An integrated approach for P2P file sharing on multi-hop wireless networks

Bin Tang, Zongheng Zhou, Anand Kashyap and Tzi-cker Chiueh

***Abstract*— P2P file sharing protocol and Ad Hoc wireless routing protocol share many intriguing similarities even though they are motivated on totally different basis. P2P file sharing systems are based on wireline IP network while mobile Ad Hoc networks (MANET) are multi-hop wireless networks in a much smaller scale. With the advances in wireless technology and mobile computing, the research of P2P file sharing in MANET has gained much momentum in recent years. One natural way is to implement P2P application and ad hoc routing at different layers they belong to. In this paper, we argue that instead of stacking one on the top of the other, more work can be done to make both P2P file sharing protocol and MANET routing protocol interact with each other. We study one representative protocol from each category. We extract their commonalities and design a common query/response framework in which P2P file sharing and Ad Hoc environment are integrated seamlessly. The extensive experiments in $ns2$ show that our strategy performs better than the layered approach in terms of traffic, average query delay and packet delivery ratio.**

I. Introduction

Peer-to-peer (P2P) networks, designed mainly for the purpose of file sharing (among others are P2P communication and distributed computing), compose of autonomous peers connected via the Internet. Its great success is due to its philosophy of migrating the job of information sharing and retrieving from some centralized servers to many hosts (or peers) that possess the content, thus making Internet a truly distributed information storage system. Mobile ad hoc networks (MANET) consist of mobile nodes communicating with each other through multi-hop wireless radio links. With the technological advances of many computing devices such as PDA, laptop and mobile phone with bluetooth, information sharing on short range wireless-connected mobile devices could be as commonly used as the Internet. The fact that people have to pay DSL providers to access Internet while there is simply no charge of utilizing some free radio band for a local P2P file sharing further validates and supports its potential as a promising communication framework in the near future.

Consider the following scenario: John has a mobile phone with 1-2GB of storage and a couple of different radios. While sitting at a Starbucks or strolling in a shopping mall, he can subscribe and join a local P2P network to see what other people around have offered to share in their mobile devices. The shared content could be a MP3 file or a movie clip, or even a multi-player game. Such so called *P2P ad hoc network* can greatly alleviate the real world deployment issues of communication systems to ultimate benefit its users. Some other emerging applications of P2P ad hoc network include information sharing at spontaneous meetings in either close (an office building) or open area (disaster relief environment), where intermediation of a centralized server is not required. Another important application of P2P over MANET could be wireless sensor networks, in which the sharing and aggregation of data sensed at each sensor node is essential for data analysis and environment monitoring. In both P2P file sharing and wireless sensor networks, the files and sensed data information are considered as *first class citizen*, not the nodes storing them – both networks concern with how to locate and disseminate *data* among each other more effectively.

Both P2P networks and mobile ad hoc networks share some fundamental commonalities such as decentralized and self-organizing architectures due to lack of central servers; dynamic topologies due to peer subscription/unsubscription in peer-to-peer networks and mobility or node failure in ad hoc networks. The nodes in both networks can act as both routers and hosts, thus any node may forward packets for other nodes as well as run user applications. Moreover, in this general communication framework, each node can be either client or server or both.

Meanwhile these two networks have substantial differences as well. P2P network is in the scale and context of Internet and it is usually wireline network. Ad hoc networks have so far mainly concerned military and disaster relief applications, in which mobile nodes use wireless interfaces to communicate with each other in a much smaller scale. More fundamental difference lies in their functionalities – P2P network file sharing is an application that deals with how to efficiently locate a set of servers containing a given file and download it; while MANET routing protocols are network layer protocols and concentrate on how to find a optimal route to a remote host.

Current P2P systems are not sufficient to provide file sharing in MANET for the following reasons. First, P2P network is an overlay network based on TCP/IP network, thus after file is located, the following step of file retrieving is done directly using HTTP or FTP, etc. This does not happen for MANET, which needs to be formed anytime and anywhere without requiring an infrastructure. Secondly, in MANET, the location of peers and the links between peers change frequently due to mobility, making structured file indexing such as DHT difficult to manage. Thirdly, in MANET, even though there are various routing protocols proposed for different network scenarios [14], there is no common and widely used standard for routing data in such environment. This paper tries to serve

Department of Computer Science, Stony Brook University. Email: `bintang,zzhou,anand, chiueh@cs.sunysb.edu`

this purpose — to develop a common query and response framework on which both ad hoc network routing and P2 file sharing can be built.

One natural approach is to directly layer file sharing application on top of MANETs, considering P2P file sharing an application on top of MANET network routing layer. However, due to the lack of cooperation and communication between these two layers, there are significant message overhead and communication redundancy. This paper analyzes both protocols carefully and integrates them together such that some common features can be combined and the incurred overhead is reduced to the maximum extent.

We study a representative protocol from each network. We choose FASTTRACK adopted by KaZaA, as the P2P file sharing protocol because among many unstructured P2P file sharing protocols (Napster, Gnutella and KaZaA are typical examples), it strikes a good balance between decentralization and scalability. We choose Ad Hoc On-Demand Distance Vector protocol (AODV) protocol in this paper, other on-demand-driven ad hoc routing protocols such as DSR [6] could potentially be used as well. We will explain later why we adopt on-demand-driven routing protocol instead of table-driven routing protocol in the P2P ad hoc networks. We also adopt some Internet P2P file sharing property and construct supernodes in MANET. We call our approach $integrated$ approach or *data-centric AODV* compared with $layered$ approach as mentioned above.

The rest of the paper is organized as follows: Section II briefly presents the related work. Section III reviews the AODV and FASTTRACK protocols and extracts the synergies existing between them. Section IV proposes our integrated way to implement FASTTRACK on top of AODV. Section V reports the simulation results and Section VI concludes the paper.

II. Related Work

Existing popular P2P file sharing systems, such as Napster [2], Gnutella [3] and KaZaA [1] have been introduced in the context of Internet. Napster [2] uses a centralized file indexing architecture to locate the peers that have the desired files. However, it has the following problems. On one hand, this centralized facility suffers single point of failure. On the other hand, it easily incurs legal challenges of copyright issue, which is the exact reason to force Napster to shut down. Gnutella system adopts decentralized search facility by plain flooding to search for desired files, which eliminates the single point of failure and avoids the legal issue in some way. However, scalability arises as a problem due to the network wide flooding. KaZaA adopts a hybrid file sharing protocol called FASTTRACK, which alleviates the problem of scalability by introducing $supernodes$. Each peer can only communicate to its supernode when trying to find a file. The set of superndoes establish a overlay and the flooding only happens within such overlay. So far, KaZaA is the most popular and widely used P2P system, with over 85 million downloads worldwide and an average of 2 million users online at any given time.

Above P2P systems fall into the category of unstructured P2P networks, in which the files stored in each peer are independent of the overlay topology. Recently, there is also a trend to build the structured P2P file sharing systems on top of distributed hash table for the same reason of scalability [12] [17] [13]. They assign identifiers to nodes and files and specify which node can store which files. These methods put emphasis on more accurate file locating, limiting the searching delay and the size of the file indexing table, aiming at a more scalable P2P protocol in large networks.

In recent years, the synergy between MANET routing and P2P file sharing has been noticed and explored. Schollmeier *et al.* [15] gave a comprehensive study comparing routing in mobile ad hoc and peer-to-peer networks. Hu *et al.* [5] made use of the synergy between DSR (Dynamic Source Routing) [6] and Pastry to design a more scalable DSR, namely DPSR, by integrating the overlay routing table in Pastry with the Route Cache of DSR into one data structure. Specifically, DPSR restricts the number of source routes that each node has to discover. With the goal of designing a more scalable MANET routing protocol, DPSR does not aim at reducing the overhead in the two different layers.

Proem [7] provides middleware support for developing P2P applications in ad hoc environment. It defines four protocols to deal with the reliable transportation, data sharing and synchronization, membership verification, and peer-announcement respectively. The author claims this platform can be adapted to different ad hoc environment, and support any P2P applications by simple extensions to these four existing protocols. Proem aims at providing middleware support for the applications, without taking care of the cross layer overheads. 7DS [8] is a P2P application over mobile ad hoc environment. It exploits the peer mobility to provide Internet connection and data sharing to intermittently connected mobile users. Again, 7DS works in the application layer, thus not able to exploiting the synergy between P2P and MANET routing to reduce the overheads.

The most relevant work to our *i*ntegrated approach is MPP [16] and Ekta [11]. MPP tries to exploit the similarities between P2P and MANET routing to reduce the overhead. It introduces a communication channel between the application and the network layer. The peer registration, searching request, acknowledgement, etc. can be transferred between these two layers through this channel. Ekta tries to include the distributed hash table into the integrated layer composed of the P2P and MANET routing. It combines two parts by providing a one-to-one mapping between the IP addresses of the mobile nodes and their node IDs in the name space. With this integration, the routing structures of DHT and the MANET routing protocol can be expressed into one structure, thus made possible the interaction between each other. These two protocols, MPP and Ekta, both use DSR as the MANET routing protocol. Particularly, in [11] it is mentioned AODV can not be integrated in the same framework. Also, in MPP, the two layers are connected via a communication channel, thus not fully integrated.

In this paper, we are trying to implement P2P file sharing on mobile ad hoc network by analyzing carefully the synergy existing between them. Particularly, we combine FASTTRACK and AODV in a fully integrated way. We also aim to reduce the extra overhead of ad hoc routing to the minimal amount, while still supporting efficient file sharing.

III. Background

In this section, We review the key features of FASTTRACK and AODV protocols and extract their commonalities to develop the frame work of P2P file sharing on MANET. Depending on how the routes are constructed, the ad hoc routing protocols can be categorized into three, namely, table-driven protocols, on-demand-driven protocols and hybrid protocols. In on-demand-driven protocols, each node constructs the route to a destination node when desired. In table-driven protocols, each node maintains the next hop routing information to all other nodes by constantly exchanging with its neighbors about this information. The hybrid protocols view the network as many distinct zones. Nodes in the same zone adopt table-driven protocols while nodes in different zones use on-demand-driven protocol to find route between them when necessary. On the other hand, file sharing tries to locate the file when needed, thus has much common characteristics with on-demand routing. Due to this reason, we adopt one of the on-demand routing protocols, AODV, to integrate with file sharing protocol.

A. FASTTRACK

FASTTRACK is the file sharing protocol used by KaZaA, a popular P2P file sharing application. It provides a decentralized P2P file sharing. Every peer stores some data in its local cache. When a node requires some data which is not stored locally, it broadcasts the query by sending unicast message to each of its neighbors. Any node on receiving this query checks its own cache, if there is a cache hit, it responses to this query with a confirm answer; otherwise, it broadcasts this query again. This network-wide flooding is obviously not scalable. To alleviate this problem, FASTTRACK proposed two solutions. First, all the queries are flooded in a controlled scope, which is restricted by a Time-To-live(TTL) field in each packet. Secondly, FASTTRACK introduces the concept of *supernode*. Supernodes are some nodes with higher bandwidth connectivity and more powerful processing capacity. They form an overlay on top of the peer to peer networks. Therefore, there is a two-level hierarchy - FASTTRACK network is divided into many clusters and each cluster is managed by one supernode. The supernode has all the file information stored in the cluster and only the supernode needs to relay and response to the queries. This reduces the message overhead and saves bandwidth, leading to better enhanced scalability to the P2P system. For each querying peer, once having obtained the IP address of the peer node with the file, a connection (e.g., HTTP) is established directly between the requestor and the file holder to complete the file transportation.

B. AODV

Ad Hoc On-demand Distance Vector protocol (AODV) [9], [10] is a on-demand-driven routing protocol, which means that routes are only created when needed. It has two main components: route discovery and route maintenance. In route discovery, a source node desiring to communicate with a destination node for which it does not have a valid route broadcasts a route request (RREQ) packet to its neighbors. An intermediate node receiving a RREQ first constructs a reverse path to the source. If it knows a valid route to the destination, a route reply (RREQ) is generated and sent back to the source. Otherwise, the same RREQ is broadcast until it reaches the destination node, which sends back a route reply (RREP) packet. To reduce the traffic, duplicate copies of the RREQ packet received at any node are dropped. As the RREQ is routed back towards the source via the reverse path, a forward path to the destination is constructed, which is used for the following data packet routing between source and destination.

Route maintenance is done using route error packets (RERR). In the dynamic topologies in ad hoc network, both link failure and node failure can happen frequently. When a node detects a link failure by a link layer feedback or a neighboring node failure by periodical $hello$ messages, a RERR is sent back to the upstream sequence of nodes to invalidate all the route going through the unavailable link or node.

C. Comparison Between AODV and FASTTRACK

FASTTRACK is an application layer protocol, it specifies file searching and retrieving mechanisms to achieve information sharing among the peers in the network. AODV is a routing protocol that is employed to efficiently search for communication path to a destination node. The main difference between these two protocols is the object item they are after – AODV is a node-centric searching, while FASTTRACK is a data-centric searching. Thus, the routing table of AODV is closely related to the underlying topology. While in FASTTRACK, the underlying topology is not a main issue; instead, it should take more data related issue into consideration when designing the overlay data-centric topology. Another difference is that FASTTRACK introduces supernodes to reduce the flooding traffic so that better scalability is achieved. While there is some research about hierarchical AODV [], the current AODV RFC has not yet adopted this concept. In this paper, we consider AODV with flat topology.

However, these two protocols adopt very similar scenarios to search for either data (in FASTTRACK) or route (in AODV). First, when there is a valid path to the data or to the destination node, the search stops; otherwise, the node broadcasts a message to its neighbors. Secondly, when a intermediate node receives the message, it either sends a confirmative message back or relay the same message to its neighbors, depending on whether it has the copy of the searched data or knows a node with a copy of the data in FASTTRACK or it is the destination node or knows a route to the destination node in AODV. However, in either cases, the flooding is a necessary step due to the decentralized nature of both networks. When implementing FASTTRACK on top of AODV, this dual flooding is considered redundant and some careful design needs to be made. Furthermore, both protocols need some auxiliary mechanisms such as intermediate caching (data or path) for better performance. Note that some issues may arise during this integration. One issue is that AODV is built on flat topology while FASTTRACK introduces two-level hierarchy; another issue is that a specific node has a unique location in the network, while a specific data may have multiple locations. In the following section, we illustrates the details of our integrated protocol of P2P systems over AODV, along with proposed solutions to the two difficulties mentioned above.

IV. Design of P2P over MANET

In this section, we describe in detail our integrated approach to implement FASTTRACK over AODV. Following the illustration above, we propose three frameworks of implementation. Depending on the depth of interaction between FASTTRACK and AODV, they are named as $layered$, $intermediate - integrated$ and $complete - integrated$ approaches. In this paper, we emphasize the comparison of $layered$ and $complete - integrated$ approach.

A. Three Approaches

Layered Approach In the layered approach, the FASTTRACK is implemented as an application on top of AODV routing protocol. The FASTTRACK works as mentioned above. The FASTTRACK routing and AODV routing happens sequentially. When a peer requests a file, if it is not in its local disk nor the pointer (nodeId) of the file is in its local disk, the peer relies on FASTTRACK routing to returns back the first nodeId which stores the desired file. Then, the peer relies on the AODV to find an optimal route to that peer. It is obvious this way has some redundance since both experience the query and response message exchange.

Intermediate-integrated Approach In above layer approach, the supernode selects another supernode as the next overlay hop irrespective of whether there are routes to that node in its routing table, which causes unnecessary route discoveries. In intermediate-integrated approach, a node that has a valid entry in routing table is given preference. Only when no such node exists, a route discovery is initiated. By this way a significant part of the flooding can be prevented. Further, in layer approach, when supernode selects the next hop, it does not consider the freshness of the route, which could possibly leads to an increase in ROUTE ERRORS.

Complete-integrated Approach To eliminate the redundance of messages and improve the system performance such as average query delay is the goal of the complete-integrated approach. The key idea of this approach is that, instead of locating the peer with the requested data and then finding the route to the node sequentially, we propose to find the node with the requested data and establish the route to it at the same time. This seemly simple idea reflects a very important philosophy in P2P file sharing world - it doesn't matter where the file is located; what count is the efficiency of file locating and retrieving. In this integrated model, each mobile node needs to take care of both file sharing and routing. We argue this cross-layer approach integrates the FASTTRACK and AODV in the most complete way and gives the best performance in terms of the delay time for each peer to request and finally retrieve certain file. We think this is a better approach for application implementation on MANET, especially for multimedia application which has strict delay requirements.

Corresponding to the packet type in AODV, we name the different packet types as RDREQ, RDREP, and RDERROR, indicating each packet serving two purposes — locating file and establishing route. Our approach works as follows. If there is a local access miss, the data query of a node is first sent to its supernode. If the supernode can not find the data in its cluster, it issues a RDREQ packet in the overlayer to find the data and the route to it. The next supernode who receives the packet follows the same procedure – checking its cluster, depending on whether it has the desired data, decides further flooding or not. In each step, a routing entry is established to its previous supernode from which the packet was received, so that a reverse path will be established along the way of flooding. The supernode which has a node with the requested data in its cluster, will reply with a RDREP packet. In the RDREP packet, not only the node ID is included, but also the data is piggybacked along the reverse path back. In case of duplicate data are returned back, the one arrived first is accepted and the rest are discarded. Here we would like to emphasize that all these happen within the overlay network.

B. Cache Locating Consideration

First we have some simple assumption. P2P application in Internet is mainly about finding and downloading entertainment materials like MP3 music file or movies, etc. Once a requestor finds the peer with interested data, it will download and store the file in the local machine without worry about too much about the disk space. Following this thought, we assume each node have enough memory space and no cache replacement mechanism is necessary. Since the main goal of our work is to explore the synergy existing between P2P file locating protocol and MANET routing protocol, we leave some interesting part such as cache maintenance and replacement as our future research. However, we do make our scenario as general as possible to accommodate any further addition of ideas such as the effect of cooperative caching to further improve the system performance.

For each data data, there are possibly several copies in the network either since the bootstrap of the network or after several rounds of queries and replies. To better keep track of the data copy/cache information in the MANET, each peer maintains a data structure called *cache routing table*. Each entry of *cache routing table* is a tuple (D_i, N_i), where D_i is the data item Id and the N_i is closest (in terms of number of hops) peer node which contains D_i, to the *best knowledge* of each peer.

The maintenance and update of *cache routing table* is as follows. At the beginning, each peer only knows the data items it stores locally; each supernode has the index of all the data items in its cluster. In the process of the data request and retrieving in the MANET, each node accumulates a better knowledge about its cache routing table by observing its local traffic. By checking the packet header of the data packet passing by, when a peer learns a peer node storing a file it currently doesn't possess, it will add one corresponding entry in its cache routing table. It also closely checks if any passing packet can give it a better (closer) peer than what its current one and updates if necessary. This *cache routing table* is due to the data centric nature of P2P application, in which the intended data files are the concern of each peer, instead of the peer node who has the file. Of course, aside from this cache routing table, each node maintains a regular routing table indicating the next hop towards each destination node. The cache routing table can reduce both network traffic and the user perceived query delay, as we will discuss in more details in the simulation part.

In the world of P2P file sharing, since peer joins and leaves constantly, the ownership of the file is not quite clear, so is the cache consistency. In most scenarios, peers share with each other such as MP3 or photos or movies clips, the data consistency is not a concern at all. So, in our work, we do not address the issue of cache consistency.

C. SuperNode Selection and Implementation

In FASTTRACK, the supernodes assume more important role than the rest of the nodes, acting as local central indexes for data in the same cluster and responding to any query to and from any node in the cluster. Supernodes increase the scalability of a P2P system by providing a hierarchical structure; super-node acts as a proxy to the clients connected to it. The major advantage of supernodes is that it controls the flooding in a P2P network, as the flooding is limited only to the super-nodes. Normally, a node with greater resources than others, or with higher degree than others, is elected as a super-node. We argue the supernode should have the highest number of node degree comparing with neighboring nodes due to the mobility of MANET.This is because more node degree means more neighbors and stronger connectivity, which are critical for data locating and routing in MANET.

Through simulation, we try to get a measure of reduction in message passing overhead when supernode is used, and how the performance varies when we compare the integrated approach querying with layered approach querying. We have simplified the implementation of super-node by a great extent because we are not interested in the super-node functionality, but just the difference in the number of messages due to it, thus the comparison of two approaches. We do not implement a super-node election technique or heuristic since finding the minimum dominating set in an arbitrary graph itself is a NP-complete problem [4]. We fix our super-nodes as some static nodes in an otherwise mobile wireless network. This brings some wireline P2P feature into the mobile environment and is a reasonable assumption.

Specifically, these supernodes are placed symmetrically inside the network so as to cover each region of a virtual grid. Since they do not move, the average number of nodes connected to them over the simulation period will be similar and some positive value. This placement of supernodes also ensures that there is at least one supernode within the range of every node.

Another simplification in our supernode implementation is that our supernode does not keep an index of the data item contained at each node in its domain. In the highly mobile network, this information is very difficult to maintain - the supernode has to beacon periodically to keep track of which nodes are in its cluster. Or when a peer will notify the supernode when it leaves one and moves into another. Each node needs some global positioning system such as GPS to accomplish this. The implementation of supernodes gets simplified because we treat a supernode like any other node. Like supernode in FASTTRACK, we use a simple algorithm to affect the controlled flooding in a network with supernodes - ”only the super-nodes forward a broadcast packet”. Thus, the flooding is limited between the super-nodes. The other nodes receive the packet and reply if they have the data, otherwise they drop the packet.

V. Performance Evaluation

We use $ns2$ to evaluate the performance of the two approaches — $layered$ and $complete-integrated$, with or without supernode considered in each case.

The simulation area is a grid of 1500 meters by 320 meters. We divide this grid into 10 sub-grids of 150 meters by 320 meters and place a super-node at the center of each sub-grid. The ”random waypoint” movement model is used in which 50 nodes move at a speed uniformly distributed between 0-20 m/s. We assume the wireless bandwidth is 2 Mbps and the transmission range is 250m. The run time of the experiment is kept as 500 second.

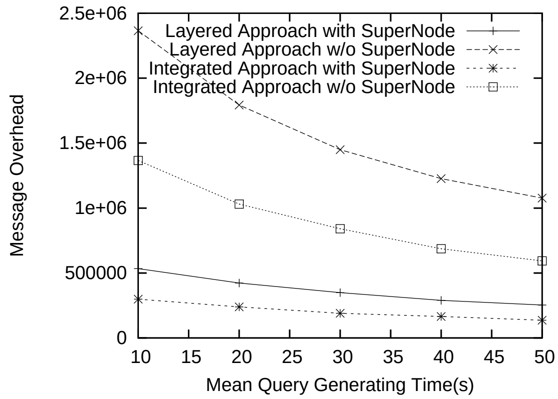
Three metrics are measured: average delay, message overhead and packet delivery ratio (PDR). Average delay is the time elapsed between the query is sent and the data is transmitted back to the requester averaged upon all the queries. Message overhead includes all the query and response messages of locating both data and optimal route. Packet delivery ratio is defined as the percentage of the data queries which receives the requested data. More work and detail plan needs to be done in this respect.

The integration of P2P file sharing and AODV routing is done in current version (2.27) of ns2. There are 100 data items randomly distributed among all 50 nodes. In the P2P model, each peer can play the role of both a client and server.

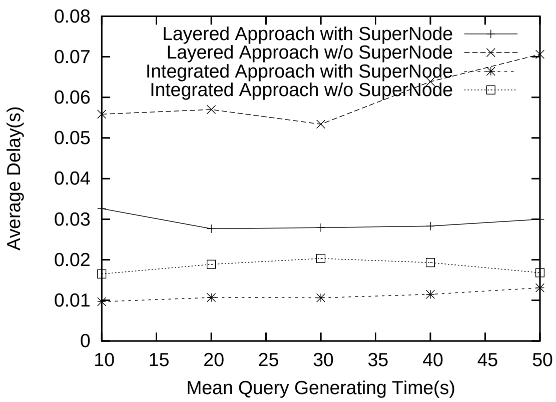
In our client query model, each node sends out queries with the interval of query generate time. The query generate time is uniformly distributed within $(0, T_{query}]$, where T_{query} is the *mean query generate time*. Once a node sends out a query, it doesn't need to wait until the data returned back before launching another data query. In the future, some more traditional data access pattern such as Zipf distribution on the Internet will be considered. However, consider the small scale of MANET as of today, the random access pattern adopted in our experiments is a reasonable choice.

A. Results and Performance Analysis

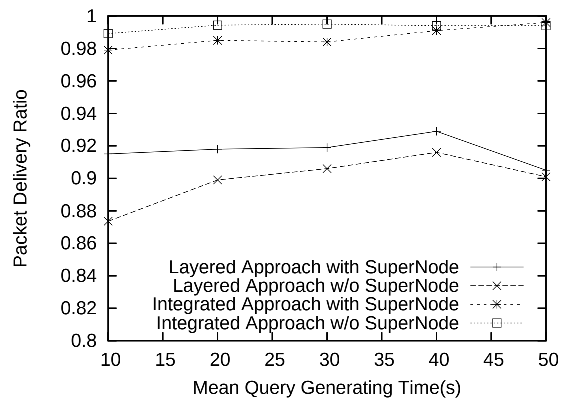
Experiments were run using different workloads and system settings. The performance analysis presented here is designed to compare the effects of different workload parameters such as mean query generate time, node density and node mobility.



(a) Overhead



(b) Average Delay



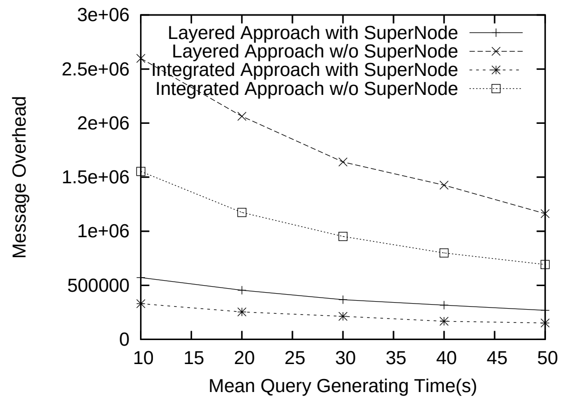
(c) Packet Delivery Ratio

Fig. 1. Static Network

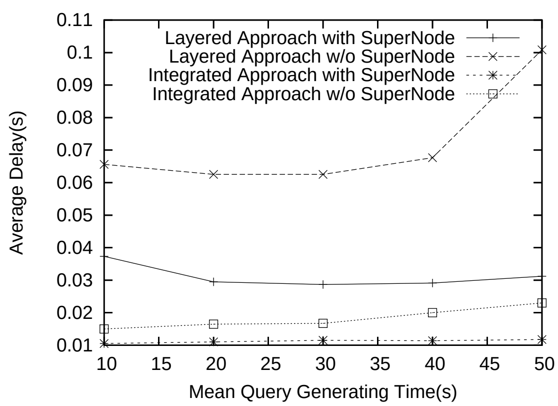
a) Effect of Mean Generating Time:: Figure 1 shows the effect of mean query generate time on the system performance in static networks. The mean query generate time is varied from 10 second to 50 second. Figure 1 (a) shows the overhead comparison. We can see that our integrated approach outperforms the layered approach. One interesting point is that instead of the difference of two times, the integrated approach outperforms the layered approach for 4 5 times. This is because that without a corresponding cache routing entry in the cache routing table, each node from the requester to the source will have to flood its message; the average hops between any pair of peers in our simulation is 5 hops. Furthermore, in either layered approach or integrated approach, the scenario with supernode outperforms the one without supernode, indicating our supernode implementation is a feasible way to achieve the network scalability. Figure 1 (b) shows the average delay comparison. Figure 1 (c) shows the packet delivery ratio comparison. Integrated approach has better PDR than layered approach, with or without supernode. This is largely due to the heavy traffic in layered approach. Furthermore, in integrated approach, the PDR with supernode is better than PDR without supernode. This is because in already heavy traffic, the supernode does play an important role to reduce the traffic. However, in layered approach, when traffic is heavy (less query generating time), the one without supernode has a relatively larger PDR. This can be attributed to the fact that when less query happens, one data item is not heavily cached on different peers compared to heavy query case.

b) Effect of Mobility: Figure 2 shows the same comparison in the mobile environment. We use the ”random waypoint” model, in which each node pauses some time before move the next randomly chosen destination. The pause time is set as 30 seconds in our experiment. Once again, we see the same trend as in static topology.

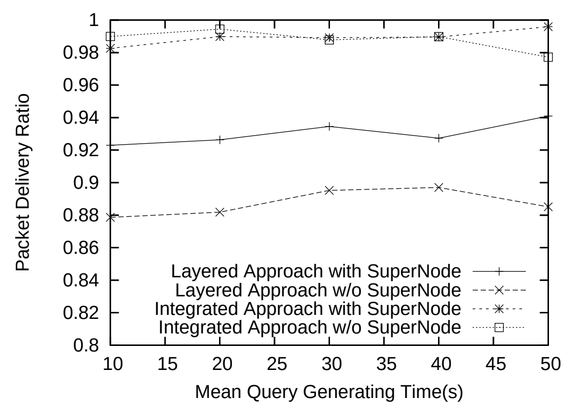
c) Effect of Speed: Figure 3 shows if there is any effect of different mobility speed on our different algorithms. We vary the maximum speed of each node from 5 sec/s to 20 sec/s. It shows the performance does not change much with the change of mobility. It can be attributed to our supernode design, which can make each node to talk with the closest supernode in one hop.



(a) Overhead

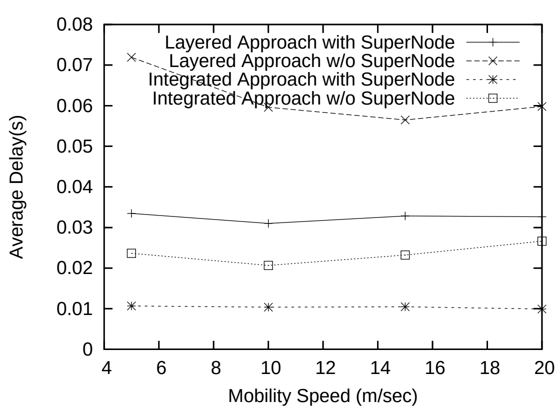


(b) Average Delay

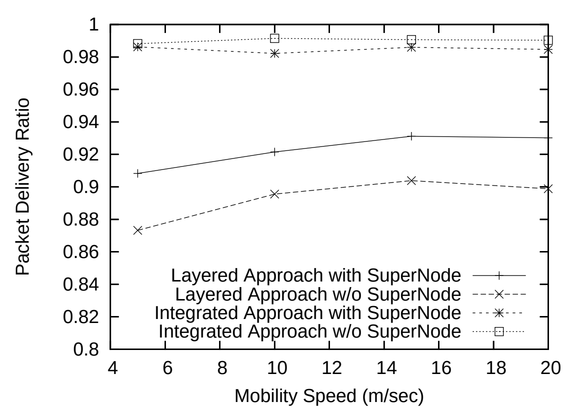


(c) Packet Delivery Ratio

Fig. 2. The effect of mobility on the network performance



(a) Average Delay



(b) Packet Delivery Ratio

Fig. 3. The effect of mobility speed on the network performance

VI. Conclusion and Future Consideration

In this paper we explore how to facilitate P2P file sharing on the environment of wireless ad hoc network. We use AODV as the representative ad hoc routing protocol to integrate with FASTTRACK P2P file sharing protocol. We show our integrate approach not only reduces average delay perceived by each file requester, but also improves the system performance as indicated by reduced overhead messages and increased packet delivery ratio.

We hope the above work can be extended into the research of wireless sensor network. Like P2P file sharing, sensor networks consider the data items as the first class citizen, not the node which store the data items. The main function of sensor network is information extraction and dissemination, which requires a more efficient, integrated application-level routing approach.

References

[1] www.kazaa.com.

[2] www.napster.com.

[3] The gnutella protocol specification v0.4. *http://dss.clip2.com/GnutellaProtocol04.pdf*, 2000.

[4] S. Guha and S. Khuller. Approximation algorithms for connected dominating sets. *Algorithmica*, Apr. 1998.

[5] Y.C. Hu, S.M. Das, and H. Pucha. Exploiting the synergy between peer-to-peer and mobile ad hoc networks. *Proceedings of HotOS-IX: Ninth Workshop on Hot Topics in Operating Systems*, 2003.

[6] D.B. Johnson and D.A. Maltz. Dynamic source routing in ad hoc wireless networks. Kluwer Academic, 1996.

[7] G. Kortuem and J. Schneider. An application platform for mobile ad-hoc networks. *Proceedings of the Workshop on Application Models and Programming Tools for Ubiquitous Computing*, 2001.

[8] M. Papadopouli and H. Schulzrinne. Effects of power conservation, wireless converage and cooperation on data dessemination among mobile device. 2001.

[9] C. E. Perkins and E. M. Royer. Ad hoc on demand distance vector routing. In *Proc. of 2nd IEEE Workshop on Mobile Computing Systems and Applications*, pages 90–100, Feb 1999.

[10] C. E. Perkins, E. M. Royer, and S. R. Das. Ad hoc on demand distance vector routing (aodv), internet rfc 3561. July, 2003.

[11] H. Pucha, S.M. Das, and Y.C. Hu. How to implement dhts in mobile ad hoc networks? *Poster in 10th ACM MOBICOM*, 2004.

[12] S. Ratnasamy, P. Francis, M. Handley, R. Karp, and S. Schenker. A scalable content-addressable network. *Proceedings of ACM SIGCOMM*, 2001.

[13] A. Rowstron and P. Druschel. Pastry: Scalable, distributed object location and routing for large-scale peer-to-peer systems. *Proceedings of MIDDLEWARE*, 2001.

[14] Elizabeth Royer and C.-K. Toh. A review of current routing protocols for ad hoc mobile wireless networks. IEEE Personal Communications, April 1999.

[15] R. Schollmeier, I. Gruber, and M. Finkenzeller. Routing in mobile ad hoc and peer-to-peer networks, a comparison. *International Workshop on Peer-to-Peer Computing*, pages 1–15, 2002.

[16] R. Schollmeier, I. Gruber, and F. Niethammer. Protocol for peer-to-peer networking in mobile environments. *Proceedings of IEEE12th International Conference on Computer Communications and Networks (ICCCN)*, 2003.

[17] I. Stoica, R. Morris, D. Karger, M.F. Kaashoek, and H. Balakrishnan. Chord: A scalable peer-to-peer lookup service for internet applications. *Proceedings of ACM SIGCOMM*, 2001.