Efficient Search Using Hierarchical Clustering in Mobile Peer-to-Peer Networks

Ssu-Wei Liu*, Jonathan Chun-Hsien Lu, San-Yi Hung
Dept. of Computer Science and Information Engineering
Fu Jen Catholic University, Taipei, Taiwan
siwi19@hotmail.com jonlu@csie.fju.edu.tw hungsy95@csie.fju.edu.tw

***Abstract* -- With the quick advance of both ad hoc network technology and peer to peer technology, there have been many applications trying to combine them in a same network. In this paper, we proposed a hierarchical clustering mechanism for peers to do effective file sharing and management and achieve a good performance in a mobile environment. We mapped the overlay network to the physical network correspondingly, and use the Content Addressable Network architecture to do adaptive topology adjustment. Empty blocks were periodically detected and covered by nodes in neighboring blocks to keep the system stabilized. Simulation results showed that our design could achieve very good performance for mobile peer-to-peer networks.**

Keywords: Peer-to-peer networks, mobile ad hoc networks, topology mismatch, file search.

I. INTRODUCTION

Current peer-to-peer (P2P) information sharing systems based on the Distributed Hash Table (DHT) approach can satisfy the needs of the exchange of wide-spread resources and voluminous information among thousands of users. Examples include Chord [1], Pastry [2], Tapestry [3], and CAN [4]. Devices in a mobile ad hoc network (MANET) can self-organize into a communication platform without an infrastructure. Those mobile devices can construct routing information using routing protocols such as DSDV [5], DSR [6], and AODV [7]. P2P networks and MANETs have some common characteristics, and it is natural to apply P2P file sharing over MANETs. In the design of P2P networks the mismatch between the overlay topology and the physical topology can impact the performance significantly. A topology mismatch example is shown in figure 1. Suppose node S wants to send a lookup message that will be routed to node T in the overlay network. The message has to be forwarded through three intermediate nodes (node B, node C, and node D), which travels more hops than the short direct physical path shown in solid line. In this work, we focus on the network topology management of a P2P network over a MANET. We propose a new file search and clustering method to adjust the mobile topology effectively. Our goal is to let peers obtain successful response quickly by searching for the desired file in the MANETs efficiently.

The rest of this paper is organized as follows: Section two reviews the related work on the topology mismatch problem. In section three we present a clustering method used to construct a peer-to-peer network over a MANET, and provide a file searching algorithm. Section four shows the simulation environment and results, and section five gives the conclusion.

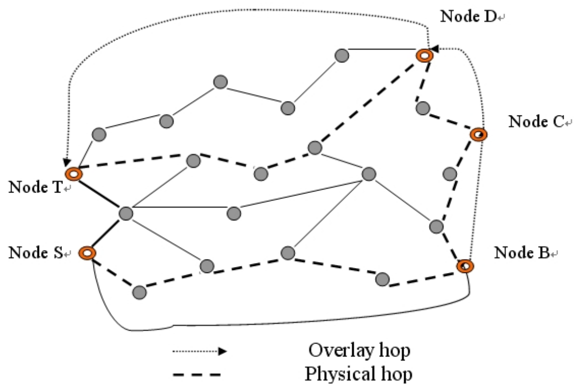


Figure 1. Messages from S to T traveling a longer path due to topology mismatch.

II. RELATED WORK

In a P2P system peers are connected to each other by logical links in the overlay network, where each logical link may correspond to a lengthy path in the physical network [8]. The topology mismatch between the overlay network and the physical network were studied in [9] and [10], which presented solution methods to efficiently build the overlay network and significantly reduce the traffic overhead. Initially the overlay network was constructed to be the same as the underlying physical network. A number of shortcuts were then generated dynamically according to the categories of the links in the overlay network. Those shortcuts were used to create a virtual link to make two nodes directly connected in the overlay network.

Landmarking [11][12] is an approach used to create DHT-based P2P overlays according to physical proximity. Random Landmarking (RLM) based on Pastry selects a set of randomly-chosen landmark keys, each of which is simply an overlay ID. When a node wants to join the overlay network, it is assigned an ID based on its landmark ordering. The could result in that the neighboring nodes physically close are likely to be also assigned numerically close to each other in the overlay space.

In [13] Content-Addressable Network (CAN) was adapted for mobile ad hoc networks. The authors proposed an effective method called SAT-Match, which could tolerate frequent joining and leaving of nodes. The method required dynamic partitioning such that every node “owns” its zone within the overall space. When node A move to node B’s neighborhood physically, it will ask node B to divide its overlay space such that each node will be responsible for half the space to achieve load sharing. This way the overlay topology and physical topology can match each closely. Each node must detect if it has moved to a new area periodically.

Authors of [15] used Chord directly in a mobile ad hoc network. Suitable thresholds could be generated for combining two Chord systems into one large ring or partitioning one Chord system into smaller ones. In [16][17] the network was imagined as a virtual two-dimensional coordinate space, which was partitioned into smaller blocks.

III. MOBILE PEER-TO-PEER CLUSTERING SYSTEM

A. Architecture Concept

There are several assumptions in our system. We assume that our overlay network is based on the structure of CAN because this structure can be dynamically partitioned and merged into zones of various sizes. We also assume that each node is aware of the geographical positions of itself and its immediate neighbors through GPS functionality. In order to have a good topology match, we assume that each node’s overlay coordinate will be calculated in the same relative proportion as its GPS coordinates to the physical space. A node’s ID is denoted as its X and Y coordinates. In a block the node closest to the center of the block will be designated as the leader of the block. Each leader must maintain a list of the other leaders including their node IDs and levels. The underlying ad hoc routing protocol is used to route a packet to the next hop. Figure 2 shows the overlay space.

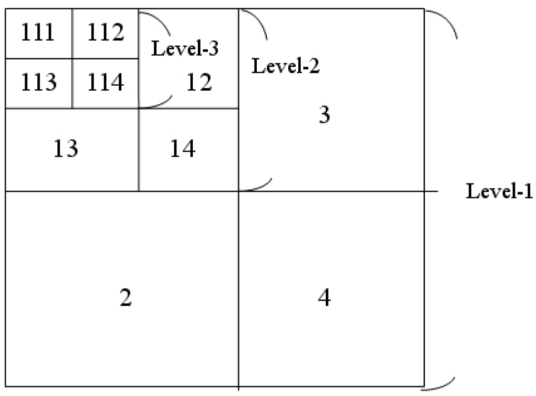


Figure 2. Overlay space and partitioning

Figure 3 shows the hierarchical architecture. A green triangle represents a leader peer and a yellow circle denotes a member peer. Each leader peer frequently exchanges information with other leader peers. A member peer connects to its leader directly, while the leaders connect with each other to form a hierarchical backbone.

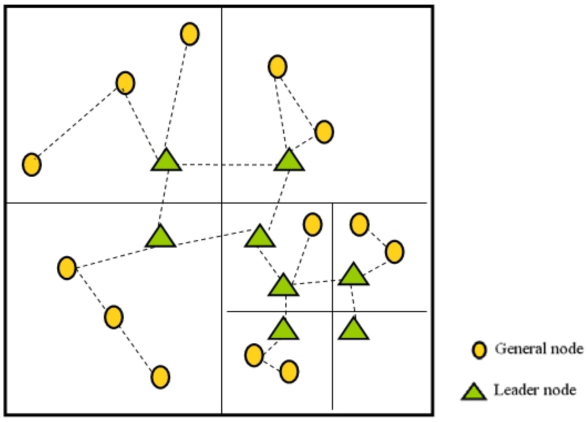


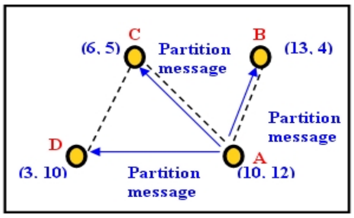
Figure 3. Clustered hierarchical architecture

B. Partitioning and Leader Selection

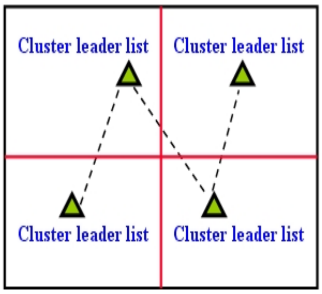
We assume that nodes can sense the existence and locations of other nodes around the area. If there are more than four nodes, they will exchange messages to each other to determine whether they are distributed in all four difference quadrants or not. If yes, this block will be partitioned to four smaller blocks for load balancing purpose. For each new block, the node closest to center of the new block is designated as the leader. Each block can be possibly partitioned into four lower-level quadrants hierarchically, as long as there exists at least one node in each quadrant after the partition. A leader is always selected for each block at every level. Each leader keeps a list of leaders for the other three companion quadrants in the same level, as well as leaders in the direct higher and lower levels. Figure 4 shows an example that when node A senses three nodes in each difference quadrant, it then sends out a partitioning message to the other nodes to perform the partitioning collectively. When a node moves and enters a new block partition, it can become the new leader if it is closer to the center than the current leader. The old leader will send a copy of its leader list to the new leader. If several nodes enter the same block partition and occupy all four quadrants, the block will be partitioned. Each leader periodically broadcasts beacon messages to allow new nodes to contact the leader.

C. Handling of Empty Blocks

When an empty block (called null block) appears in one of the four quadrants, one of the neighboring blocks is chosen to cover the empty block. There can be three different cases as follows: (i) There is only one empty block in the four quadrants. Any of the three neighboring blocks can cover the empty block as shown in figure 5a. (ii) There are two empty blocks, each of which will be covered by one of the non-empty blocks, respectively, as shown in figure 5b. (iii) Three out of four quadrants are empty. In this case the nodes in the non-empty quadrant will merge the four blocks into a larger one at a higher level, and take responsibility for this larger block as shown in figure 5c.



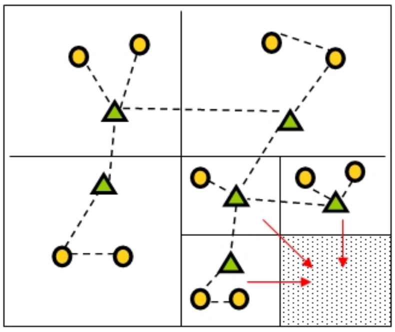
(a)



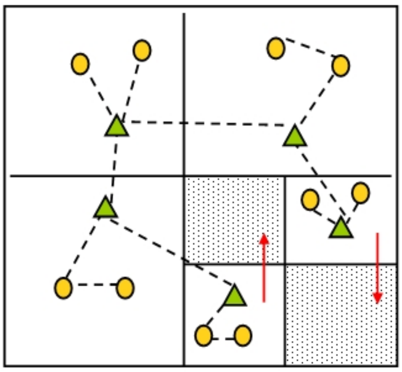
Cluster leader list		
(X,Y)	length	level
(6,5)	3.6	1
(13,4)	6.4	2
(3,10)	5.3	3
(10,12)	4.4	4

(b)

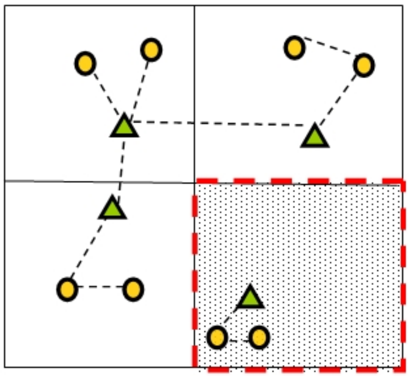
Figure 4: (a) Four nodes in difference quadrants before partitioning. (b) After partitioning.



(a)



(b)



(c)

Figure 5. Covering of empty blocks: (a) one empty block, (b) two empty blocks, and (c) three empty blocks.

D. File Search Process

Each file is first mapped to an X and Y coordinate via a hash function. The owner node them registers the file by sending a registration message to be routed to that coordinate. When the message is forwarded to a node responsible for that coordinate, the node will broadcast the registration to all the other nodes in the same block for the purpose of fault tolerance. When a node wants to look for a file, it will send a query message to its leader. The leader then looks up the leader list to find the level which the leader that manages this file resides in. The search process may go to levels up or down along the hierarchical architecture. An example is shown in figure 6. The source node first calculates the target file's coordinate, which is located in block *A* managed by leader 4. Then the source node consults the leader list in its local leader (namely leader 1), and route the query message through leader 2 in level 2 to leader 3 in level 1, to finally reach leader 4 in level 1.

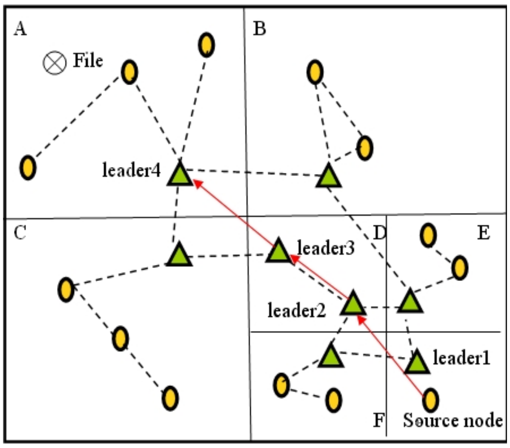


Figure 6. File search through the hierarchical architecture.

IV. PERFORMANCE EVALUATION

A. Simulation Environment

We have implemented a simulation program in Borland C++ Builder 6.0 to evaluate the performance of our design. We assume that all nodes may move around in a 1024*1024m^2 area. A number of nodes ranging from 30 to 300 are generated and randomly distributed in the network. Each node has 10 files randomly selected out of 1024 specific files. Transmission range of every node is set to 361 meters. Stabilization period ranges from 1 to 20 seconds. We compare our method to a random structure, where each node is hashed to an overlay coordinate randomly and there is no hierarchical architecture.

B. Simulation Results

We first study the number of levels formed in our hierarchical architecture. Figure 7 shows that the maximum number of level remains 4 when the number of peers exceeds 150. Figure 8 displays the average number of hops a query message travels in the overlay network. The average number of hops in our method is smaller than the random scheme due to the hierarchical nature. The topology mismatch in the random scheme also results in higher number of hops.

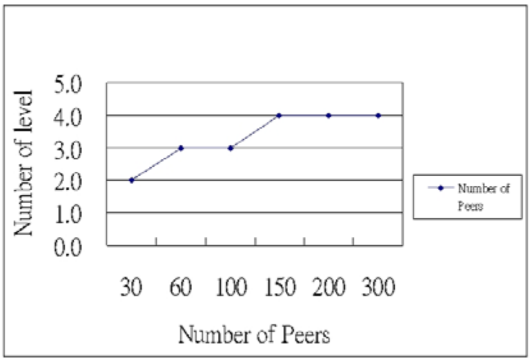


Figure 7. Number of levels in our hierarchical architecture

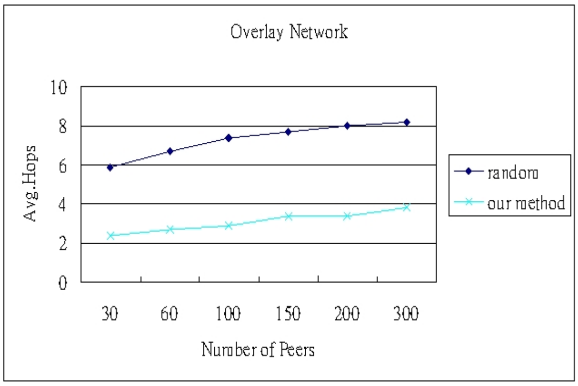


Figure 8. Average number of hops in overlay network

Figure 9 plots the successful rates of search queries. It shows that our method has a better hit rate than the random method. We next study the physical distances that a query message has to travel to find the target file. Figure 10 shows that the distance for our method is much shorter because of the good match between the physical network topology and the overlay network topology. With the random method, one hop in the overlay network may actually span a long distance in the physical network.

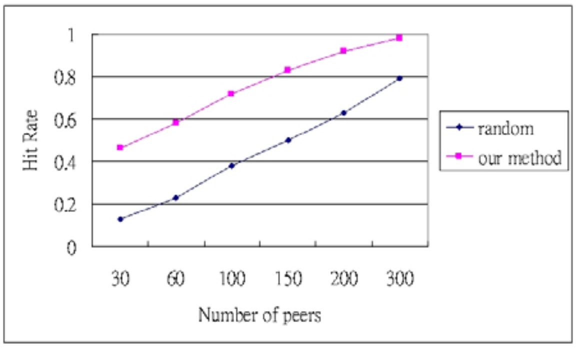


Figure 9. Successful rates of search queries.

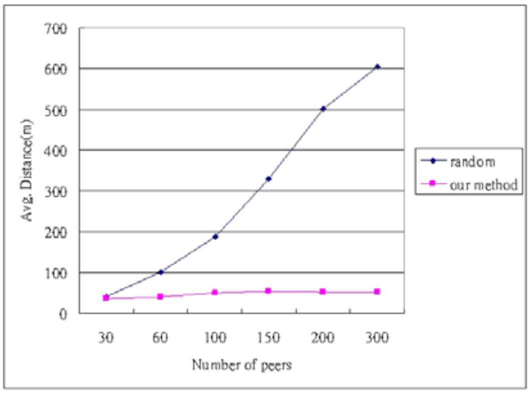


Figure 10. Distance traveled in the physical network

As mentioned before, since nodes may move around, each leader node must periodically detect and try to cover empty blocks in order to stabilize the system. This stabilization will consume network bandwidth. If this stabilization period is set longer to save bandwidth, the hit rate would decrease as well. In cases where there are not enough nodes, less frequent stabilization cause the hit rate to become worse. Figure 11 and 12 plot the results.

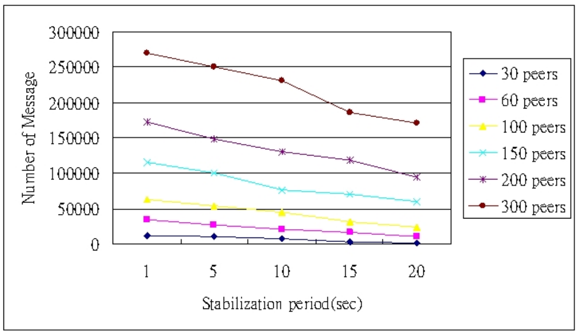


Figure 11. Traffic due to stabilization

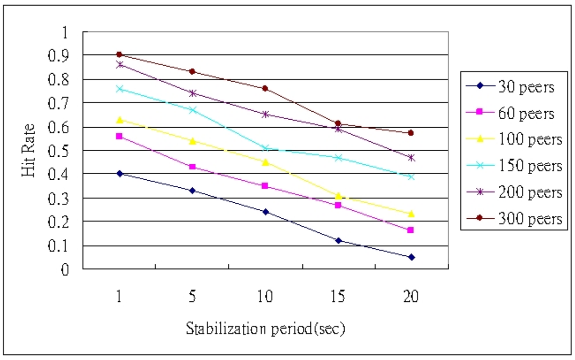


Figure 12. Successful rates vs. various stabilization periods

V. CONCLUSION

Due to the decentralized feature of P2P file sharing systems regarding resource management, its architecture applies naturally to ad hoc networks. In this work we presented a hierarchical clustering architecture to manage and search for files in mobile peer-to-peer networks. We assigned the overlay coordinate of a peer corresponding to its physical location to obtain a good topology match. Peers are organized into a hierarchical structure in order to reduce the number of hops a query may have to travel. Empty blocks were covered by neighboring nodes to keep the hit rate high. Experimental results indicated that our design showed very good performance in both static and mobile environments.

REFERENCES

[1] I. Stoica, et.al., "Chord: A salable peer-to-peer lookup protocol for internet applications", *IEEE/ACM Transactions on Networking*, vol.11, no.1, Feb 2003.

[2] A. Rowstron and P. Druschel, "Pastry: Scalable, decentralized object location and routing for large-scale peer-to-peer systems", *18th IFIP/ACM International Conference on Distributed Systems Platforms*, Nov.2001.

[3] B. Zhao, J. Kubiatowicz, and A. Joseph, "Tapestry: An infrastructure for fault-tolerant wide-area location and routing", *Technical Report UCB/CSD-01-1141, U.C. Berkeley*, April 2001.

[4] S. Ratnasamy, P. Francis, M. Handley, and R. Karp, "A scalable content-addressable network", *Proc. of ACM SIGCOMM Conference*, 2001.

[5] C.E. Perkins and P. Bhagwat, "Highly dynamic destination sequenced distance vector routing (DSDV) for mobile computers", *Proceedings of ACM SIGCOMM 94*, pp. 234-244, 1994.

[6] D.B. Johnson and D.A. Maltz, "Dynamic source routing in ad hoc wireless networking", *Mobile Computing*, pp.153-181, 1996.

[7] C.E. Perkins, E.M. Belding-Royer and S.R. Das, "Ad hoc on-demand distance vector (AODV) routing", *IETF RFC 3561*, Feb. 2003.

[8] D. Doval and D. O'Mahony, "Overlay networks", *IEEE Internet Computing*, pp. 79-82, July/August 2003.

[9] Y. Huang and B. Jin, "Topology mismatching aware p2p overlay construction in wireless ad hoc networks", Available at: http://huangyu.rocklv.net/huangyufiles/paper2005/dtaoc.pdf

[10] Y. Huang, B. Jin, and J. Cao, "A distributed approach to construction of topology mismatching aware p2p overlays in wireless ad hoc networks", *Euromicro International Conference on Parallel, Distributed, and Network-Based Processing*, pp.15-17, Feb. 2006.

[11] R. Winter, T. Zahn, and J. Schiller, "Random landmarking in mobile, topology-aware peer-to-peer networks", *10th IEEE International Workshop on Future Trends of Distributed Computing Systems*, vol. 26-28, pp.319 – 324, May 2004.

[12] T. Zahn, R. Winter, and J. Schiller, "Simple, efficient peer-to-peer overlay clustering in Mobile, ad-hoc networks", *12th IEEE International Conference on Networks, 2004. (ICON 2004)*, vol.2, pp.16-19, Nov.2004.

[13] S. Ren, L. Guo, S. Jiang, and X. Zhang, "Sat-match a self-adaptive topology matching method to achieve low lookup latency in structured p2p overlay networks", *Parallel and Distributed Processing Symposium*, pp.26-30, April 2004.

[14] T. Heer, S. Gotz, S. Rieche, and K. Wehrle, "Adapting distributed hash tables for mobile ad hoc networks", *Fourth Annual IEEE International Conference on Pervasive Computing and Communications Workshops*, pp.13-17, March 2006.

[15] X. Zhichen, T. Chunqiang, and Z. Zheng, "Building topology-aware overlays using global soft-state", *23rd International Conference on Distributed Computing Systems*, pp.19-22, May 2003.

[16] J. Li, J. Jannotti, D.S.J. De Couto, D.R. Karger, and R. Morris, "A scalable location service for geographic ad hoc routing", *the 6th International Conference on Mobile Computing and Networking*, pp.120-130, 2000.

[17] T. Matthias , F. Holger, W. Jörg, M. Martin, and E.Wolfgang "A hierarchical approach to position-based multicast for mobile ad-hoc networks", *Wireless Networks*, vol.13, pp. 447-460, 15 June 2006, *Springer Netherlands.*