A RESOURCE DISCOVERY STRATEGY FOR MOBILE PEER-TO-PEER NETWORKS

Hao Wu, Chao Cheng and Zhangdui Zhong
State Key Laboratory of Rail Traffic Control and Safety, Beijing Jiaotong University, 100044 Beijing, China

Keywords: Mobile P2P, Resource discovery strategy, M-Tapestry.

Abstract: Resource discovery is one of the key issues in mobile P2P studies. Nonetheless, in existing schemes there are still some deficiencies, such as the lake of flexibility, proportionality and adaptability. In this paper, we present a resource discovery model Mobile-Tapestry (M-Tapestry) for mobile P2P networks based on the P2P Tapestry strategy. The philosophy of this model is to make the network architecture layered , and only a small number of super nodes are chosen instead of all nodes to access the Tapestry system. This measure may reduce the cost of ordinary nodes and solve the reliability problems caused, when the routing table is updated frequently. Additionally, we partition the nodes according to their geographical location, this may avoid the problem that the distance between the nodes having adjacent IDs is actually large due to the Hash operation. Finally, our simulation results and analysis shows that the M-Tapestry strategy has an improved lookup efficiency for mobile P2P networks compared to the Tapestry strategy.

1 INTRODUCTION

With the rapid development of mobile Internet applications, researchers started to pay attention to using P2P technologies in the mobile environment. Resource discovery strategy is one of the key issues for mobile P2P (MP2P) applications (Ponmozhi and Rajesh, 2009; Voitenko, 2009; Tianhao and Nikolaidis, 2004). However, due to the difference between the mobile and Internet transmission environment, bandwidth limitations, node mobility, and multi access interference pose unique challenges to deploying the existing P2P technologies in MP2P networks. Therefore they must be improved in MP2P applications.

In the current researches, (Andersen et al., 2004) built a Mobile eDonkey network for 3G networks based on extending the Internet eDonkey system. Similarly, (Hossfeld et al., 2005; Bakos et al., 2003; Cramer and Fuhrmann, 2005) proposed a type of eDonkey solution which may be used for mobile operational networks with some improvements. Its basic idea is to make use of the resources in one mobile network operator as far as possible, hence it could liminit the network traffic mainly occurred in the local network.

Distributed Hash Table (DHT) is another type of distributed resource discovery strategy, CAN (Ratnasamy et al., 2001), Chord (Stoica et al., 2003), Pastry (Rowstron and Druschel, 2001), Tapestry (Zhao et al., 2001) are typical structured P2P system. Some researchers took into account these strategies used in MP2P networks, for examples, in (Peng et al., 2004) the authors proposed an improved CAN lookup protocol M-CAN which introduced the concept of super node. (Musolesi et al., 2005) studied the scheme that using Chord Ring structure to organize nodes and route between nodes in wireless sensor networks, and also gave the experiments of this scheme. (Pucha et al., 2004) proposed Ekta which integrates a DHT with a multi-hop routing protocol based on geographic location, Ekta divides a mobile P2P network into many subareas with the same range, each one is responsible for a series of hash key values. (He et al., 2005) proposed a P2P video streaming sharing strategy in the the mobile network environment by taking the access point (AP) as a resource index.

In this paper, considering that Tapestry is an ideal resource discovery strategy for its low space complexity and lookup complexity compared to other strategies, we propose a Tapestry-based, efficient, flexible, high-dynamic resource discovery strategy “Mobile-Tapestry” (M-Tapestry) and discuss the performance. The rest of this paper is organized as follows. In Section 2, we present the M-Tapestry system model and propose a resource discovery protocol for mobile P2P networks; In Section 3, we give the lookup efficiency

Wu H., Cheng C. and Zhong Z. (2010).
A RESOURCE DISCOVERY STRATEGY FOR MOBILE PEER-TO-PEER NETWORKS.
In *Proceedings of the International Conference on Wireless Information Networks and Systems*, pages 42-46
DOI: 10.5220/0002987200420046

performance simulation and analysis of proposed M-Tapestry strategy ; Section 4 concludes the paper.

2 SYSTEM MODEL

The resource discovery strategy, M-Tapestry, present in this paper for mobile P2P networks is based on the Tapestry, a existing discovery model for P2P networks. Tapestry (Zhao et al., 2001) is a Plaxton-based distributed query algorithm system, resources and nodes are both identified with the GUIDs (Globally Unique Identifier), which are unrelated with the location and contents and are determined randomly. Tapestry model has the advantages of low complexity and balanced load, consequently it is more suitable to be applied in the mobile P2P networks. On the other hand, the disadvantage is that the node maintenance overheads for neighbor mapping table is very heavy, because nodes join and leave the network frequently in the mobile P2P environment, moreover it will lead to system instability. Therefore, this paper designs a M-Tapestry model for mobile P2P networks combined with the ideas of (Peng et al., 2004) and (Pucha et al., 2004). M-Tapestry model first partitions some subareas according to node geographical location, and introduces the concept of super node, which may take charge of the routing information maintenance. Whereas the shared resource information is maintained by the ordinary nodes, moreover it can reduce the communication overheads, and improve bandwidth efficiency and the lookup speed.

2.1 M-Tapestry Basic Idea

M-Tapestry is layered with two types of nodes: super nodes and ordinary nodes (Peng et al., 2004). The super nodes constitute the higher layer of the network, those nodes with high performance are chosen to be super nodes, such as with a high computing capacity and stability, good reliability, etc. In mobile P2P environment, sink nodes and part of the Internet terminals can be looked as super nodes, because they are very important to the entire network. The rest of nodes are treated as ordinary nodes, which constitute the lower layer of the network. The super nodes are managed through Tapestry routing protocol, and the ordinary nodes are directly registered to the super nodes. M-Tapestry gives full consideration to the network physical architecture. Those ordinary nodes controlled by the super nodes can be divided into groups by geographical characteristics (Pucha et al., 2004), which can ensure ordinary nodes have less hops to the super node, and avoid the problem that the adjacent ID nodes is actually located a far distance due to the hash operation.

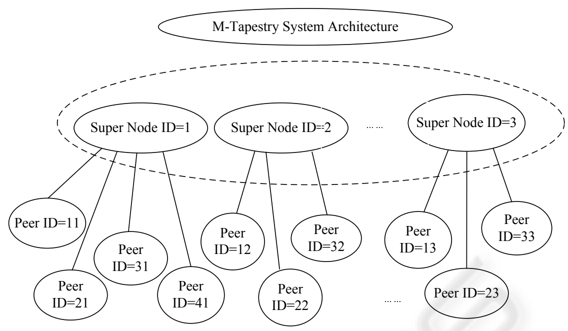

Figure 1: The system architecture of M-Tapestry.

The architecture of M-Tapestry model is shown in Figure 1. Supposed that super nodes are distributed 1 bit for Super Node ID, according to the geographical characteristic, each super node takes charge of managing the registered resources of nodes in a subarea range. Resource IDs can be obtained through the hash digest. The routing between super nodes is completed by means of Tapestry model.

M-Tapestry model involves the following terms:

- Super Node ID: Super node identifier;
- Peer ID: Ordinary identifier. Peer ID = Peer ID(prefix)+ Super Node ID, and Peer ID (prefix) can be randomly obtained;
- Object ID: Shared resource identifier.;
- Message ID: Message identifier;
- Root ID: Root identifier, for each shared resource object, M-Tapestry allocates a relevant node, which called the root of the resource, root (objectID) = the Peer ID closest to the object ID;
- Server ID: Server identifier. Server ID has been introduced to allow for routing more easily in M-Tapestry system.

2.2 M-Tapestry Routing Mechanism

In M-Tapestry system, routing between super nodes is accomplished according to Tapestry routing mechanism, each super node maintains a table of node registered resources within a certain subarea range and a mapping table for its neighbor super nodes in accordance with the routing hierarchy. Meanwhile each super node also maintains a back pointer list, which points to those nodes who look itself as a neighbor. During routing period, the super node looks up the level of neighbor mapping table in turn, and routes the data as far as possible by matching a larger prefix, until it arrives the destination super node.

When the Server Node S plans to release a shared resource O, it first sends a "registered" message to its Super Node S_S, which includes the information of node Server ID and Object ID (GUID is O_G). S_S reports that there is a resource O saved in S through sending a message to the Super Node S_O periodically, S_O is the registered super node of resource O's root node O_R. S_O forwards it and informs the root node O_R that Node S contains the resource O, each super node in this release route stores the resource O's location information pointer $< O_R, S >$, where the location information is only a pointer pointing to S, rather than a copy of the object O.

When the Client Node C needs to request a resource, it first sends a "resource request" message to its Super Node S_C, which includes the information of node Peer ID and the requested resource Object ID. S_C sends a "lookup" message to the Super Node S_O which is the registered node of root node, if a super node in the lookup route has the requested resource location information, it will directly turn to the Super Node S_S which is the registered super node of Server Node, and then the Client Node requests a copy of resource from the Server Node through the Super Node S_S; Otherwise, the message finally reaches the Root Node through the Super Node S_O, it can get the location information pointer of resource O, then the Client Node C may connects to the Server Node S and download the resource.

As shown in Figure 2, it illustrates a description of the actual routing process. Each super node is assigned a Super Node ID number, and maintains a resource table of registered nodes, in which the node ID number and the registered resource information are included. When the node with Peer ID = 0000 requests a resource with Object ID = 0111, it first sends a RootToObject request message to its registered super node with Super Node ID = 00, which includes Object ID and Peer ID information. Receiving this message, super node analyzes the information of Object ID and gets to know its registered super node is with Super Node ID = 11, because these information are contained in the message RootToObject. The message is forwarded by Tapestry routing in the network between super nodes, which is easy to arrive the super node with Super Node ID = 11. Super node with Super Node ID = 11 looks up its own resources table of registered nodes, and transmits message to the node with Peer ID = 0111, node with Peer ID = 0111 returns the Server ID information corresponding to Object ID = 0111 through its own stored information, that is to respond Server ID = 1101. The super node with Super Node ID = 11 returns "Server ID = 1101" to the super node with Super Node ID = 00, then super node with Super Node ID = 00 forwards this message to the client node with Peer ID = 0000. Receiving this message, the client node with Peer ID = 0000 sends a request to the server node with Peer ID = 1101 for resource information, the server node with Peer ID = 1101 returns the copy of Object ID = 0111. Thus, this resource discovery process is completed.

In this example, only one Server ID has been returned, whereas in practical process, when the message forwarded to the node with Super Node ID = 11, it is possible that the Server ID information of the Object ID is found, then it returns the Server ID directly without finding the ultimate Root ID to get the Server ID. In addition, the Super Node ID is only assigned for 2 bits, hence it has found the destination super node in one hop during the message forwarded between nodes, the actual situation may exist multi-hops.

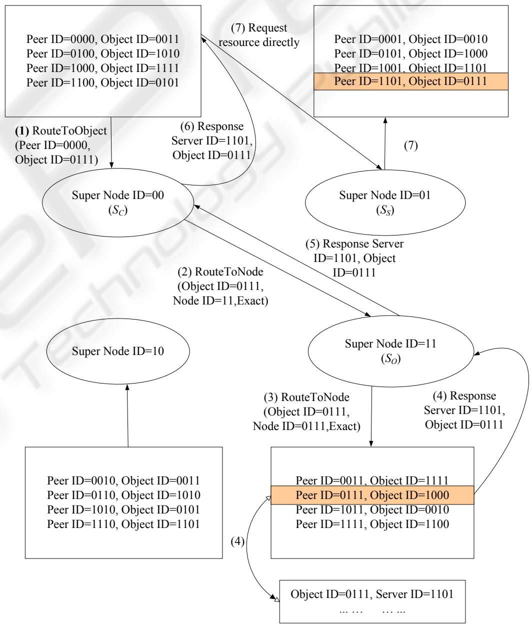

Figure 2: The M-Tapestry routing schematic diagram.

2.3 Node Joining and Leaving

In M-Tapestry model, nodes often join in or leave the network, including ordinary node joining in or leaving a subarea managed by a super node; the exit of super node, and the splitting of super node.

It is very easy for ordinary node to carry out a joining process. First, the ordinary node broadcasts a "joining" message within the entire network, then all super nodes which have received the message return a "response" message, and the ordinary node chooses the nearest super node to register according to the time-delay of received response message. Hence, the joining process is accomplished. The leaving process of ordinary node is classified as two situations: active leaving or proactive leaving. When the super node receives an “exit” message from an ordinary node (active), or monitores the failure of the ordinary node (proactive), the super node will delete the relative information of the ordinary node.

When the number of ordinary nodes managed by the super node beyond the threshold *M*, it will lead to the splitting of super node. During the splitting process, the super node divides those nodes maintained by itself into two groups, one group is still managed by the Original Super Node (OSN), and another group is managed by the New Super Node (NSN). OSN will select an ordinary node as the NSN according to the reliability and stability of the connection between nodes.

Either super node’s failure or leaving actively are resulted in the super node exiting from the network, and it can be looked as a leaving super node (LSN). Ordinary nodes may detect the failure of super node by the “heartbeat” mechanism, and then ordinary nodes change to register the primary neighbor super node, the registration process is same as the joining process of ordinary node. When the leaving super node actively leaves the network, it will transfer those nodes managed by itself to the primary neighbor super node, and informs other neighbor super nodes to update their neighbor mapping table by means of the back pointer.

3 SIMULATION AND ANALYSIS

We use P2PSim software platform in Linux environment to simulate the performance of the M-Tapestry model compared to the Tapestry model. The configure of the parameters is: random network topology; In M-Tapestry model, number of super nodes is 24, number of ordinary nodes is 1000; Each super node can manage up to 100 ordinary nodes, the super nodes and those ordinary nodes are partitioned according to geographical location. Node query interval: lookupmean = 60000ms; the average survival time of super node: lifemean1 = 3600000ms; the average time of death of super-node: deathmean1 = 3600000ms; the average survival time of ordinary node: lifemean2 = 3600ms; the average time of death ordinary node: deathmean1 = 3600ms, time Simulation duration is 200000ms; started statistical time is100000 ms; In Tapestry model, there are total 1024 nodes, which other parameters are same as the above.

We are primarily concerned with the lookup efficiency performance for M-Tapestry. The lookup mean time statistical distribution region of M-Tapestry and Tapestry is shown in Table 1. Processed the simulation results, we can get the lookup mean time probability distribution of M-Tapestry and Tapestry, as shown in Figure 3.

Table 1: Lookup time statistical data in M-Tapestry and Tapestry.

Period (ms)	M-Tapestry(times)	Tapestry(times)
0-100	786	56
100-200	84	264
200-300	46	283
300-400	34	236
> 400	10	121

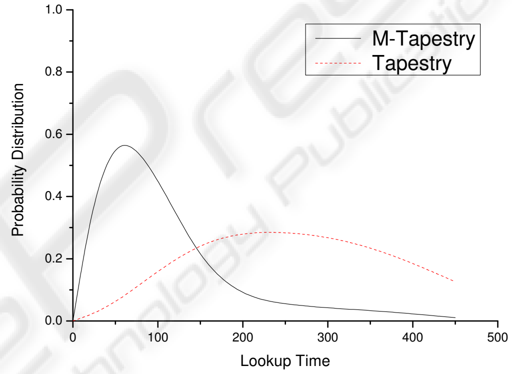

Figure 3: Lookup mean time probability distribution.

Observed from Table 1 and Figure 3 we can conclude that the resource discovery efficiency of the M-Tapestry model is better than Tapestry model, for example, in M-Tapestry, about 81.87% values of lookup time are between 0-100ms, but in Tapestry it is obviously lower than this percent, only 4.79% between this period. It is because the nodes in M-Tapestry model are partitioned at first according to geographical location to ensure that the nodes in neighbor mapping table are also adjacent in actual distance. And the architecture of M-Tapestry model is layered , the distance between ordinary node and its registered super node is short, so when using a direct point to point communication, the propagation delay is very low. As we known, Tapestry routing mechanism are used between super nodes, the lookup complexity is $O(logn)$, in which the value of *n* in M-Tapestry system is obviously much smaller than that in Tapestry. Therefore, the lookup efficiency of M-Tapestry model is much better than that of Tapestry.

4 CONCLUSIONS

This paper addressed various resource discovery strategies designed for mobile P2P networks. Compared to traditional fixed P2P networks, the new challenges of MP2P require improved resource discovery strategies. Therefore we improved the Tapestry model and proposed the novel M-Tapestry model for mobile P2P networks, which is both more efficient and more flexible. We adopted a layered architecture in M-Tapestry, and instead of allowing all nodes to access the Tapestry system, only a small number of super nodes were chosen for this purpose. Thus our solution reduces the cost of ordinary nodes and solves the reliability problems caused by frequent of routing table updates; On the other hand, we partitioned the nodes according to their geographical location, which assists us in avoiding the problem that the distance between ID-adjacent-nodes may be high due to the Hash operation. We analyzed the lookup efficiency of the M-Tapestry model, and verified that it is more suitable for employment in mobile P2P networks.

ACKNOWLEDGEMENTS

The authors thank Prof. Lajos Hanzo for many helpful suggestions. This paper is supported by the Joint State Program of the National Natural Science Foundation of China (NSFC) and the National Railway Ministry of China under project grant number 60830001.This paper is also supported by the State Key Laboratory of Rail Traffic Control and Safety under key project grant number RCS2008ZZ007.

REFERENCES

Andersen, F., deMeer, H., and Dedinski, I. (2004). An architecture concept for mobile p2p file sharing services. In *Proc. of the Workshop at Information 2004 Algorithms and Protocols for Efficient Peer-to-Peer Applications*, pages 229–233.

Bakos, B., Csucs, G., Farkas, L., and Nurminen, J. (2003). Peer-to-peer protocol evaluation in topologies resembling wireless networks. an experiment with gnutella query engine. In *Proc. of the 11th IEEE International Conference on Networks (ICON2003)*, pages 673–680.

Cramer, C. and Fuhrmann, T. (2005). Isprp: A message-efficient protocol for initializing structured p2p networks. In *Proc. of the 24th IEEE Int'l Conf. on Performance, Computing, and Communications*, pages 365–370.

He, Y., Lee, I., Gu, X., and Guan, L. (2005). Centralized peer-to-peer video streaming over hybrid wireless network. In *Proc. of IEEE International Conference on Multimedia and Expo(ICME 2005)*, pages 550–553.

Hossfeld, T., Tutschku, K., and Andersen, F. (2005). Mapping of file-sharing onto mobile environments: Feasibility and performance of edonkey with gprs. In *Proc. of the IEEE Conf. Wireless Communications and Networking*, volume 4, pages 2453–2458.

Musolesi, M., Hailes, S., and Mascolo, C. (2005). Adaptive routing for intermittently connected mobile ad hoc networks. In *Proc. of the Sixth IEEE International Symposium on World of Wireless Mobile and Multimedia Networks (WOWMOM 05)*, pages 183–189, Washington, DC, USA. IEEE Computer Society.

Peng, G., Li, S., Jin, H., and Ma, T. (2004). M-can: a lookup protocol for mobile peer-to-peer environment. *International Symposium on Parallel Architectures, Algorithms, and Networks*, pages 544–549.

Ponmozhi, K. and Rajesh, R. (2009). Applying p2p in manets for resource sharing. In *Proc. of 2009 International Conference on Control, Automation, Communication and Energy Conservation*, pages 1–5.

Pucha, H., Das, S. M., and Hu, Y. C. (2004). Ekta: An efficient dht substrate for distributed applications in mobile ad hoc networks. In *Proc.of the 6th IEEE Workshop on Mobile Computing Systems and Applications (WMCSA 2004), English Lake District*, pages 163–173.

Ratnasamy, S., Francis, P., Handley, M., Karp, R., and Shenker, S. (2001). A scalable content addressable network. In *Proc. of ACM SIGCOMM 2001*, pages 161–172.

Rowstron, A. and Druschel, P. (2001). Pastry: a large-scale, persistent peer-to-peer storage utility. In *Proc. of the 8th Workshop on Hot Topics in Operating Systems*.

Stoica, I., Morris, R., Liben-Nowell, D., Karger, D., Kaashoek, M., Dabek, F., and Balakrishnan, H. (2003). Chord: A scalable peer-to-peer lookup protocol for internet applications. *IEEE/ACM Trans. on Networking*, 11(1):17–32.

Tianhao, Q. and Nikolaidis, I. (2004). On the performance and policies of mobile peer-to-peer network protocols. In *Proc. of Second Annual Conference on Communication Networks and Services Research*, pages 208–217.

Voitenko, I. (2009). On the implementation problems of mobile peer-to-peer networks. In *Proc. of 29th IEEE International Conference on Distributed Computing Systems Workshops, ICDCS Workshops '09*, pages 455–458.

Zhao, B. Y., Kubiatowicz, J. D., and Joseph, A. D. (2001). Tapestry: An infrastructure for fault-tolerant wide-area location and. Technical report, University of California at Berkeley, Berkeley, CA, USA.