# Efficient Delay Aware Peer-to-Peer Overlay Network

Da-lu Zhang and Chen Lin

Department of Computer Science and Technology,
Tongji University, Shanghai 200092, China
daluz@ieee.org
spalding@sohu.com

**Abstract.** In many P2P systems, the path between any two nodes is mainly decided by their topology of overlay network instead of physical one. However, peers are scattered in distributed geography, requests may route across different Autonomous Systems (AS) . So, the inconsistency between overlay network and physical one can not ensure the routing efficiency in the real world. We introduce DAPS (Delay Aware P2P System), a new P2P overlay network that considers both overlay and physical networks. DAPS adopt end-to-end delay as the performance metrics and use "pruning flooding" to achieve efficient routing.

## 1 Introduction

Peer-to-Peer (P2P) networks have become the fast growing and the most popular Internet application in recent years. Traditionally, P2P network can be roughly divided into two aspects according to its organization: unstructured and structured.

Whether being an unstructured or structured network, there is a same point. Peers in P2P network communicate in a logical overlay network. In this way, the path between any two nodes is mostly decided by the hops in the overlay network level, namely the logical distance between two nodes. But, in fact, the routing efficiency of P2P overlay network largely depends on the end-to-end delay in network level, namely the physical distance. Because all nodes are scattered in distributed geography, some requests may route across different Autonomous System (AS), while other requests only hop in the same network. That the actual topology of network is inconsistent with overlay network may lead to a problem: the shortest path in overlay network level can not ensure the efficiency in real world.

We propose a new P2P system model DAPS that consider both overlay and physical networks, and use end-to-end delay as the performance metrics to achieve efficient routing.

The rest of the paper is organized as follows. Section 2 introduces and describes the related work on P2P systems. Section 3 introduces the concepts and design behind DAPS. Section 4 presents early results from our simulation environment and some conclusions are drawn in Section 5.

## 2 Related Work

There exist a series of scalable overlay networks, such as Pastry [1], Plaxton [2], Tapestry [3], Chord [4] and Can [5], all offering DHT service. Their common theme

W. Fan, Z. Wu, and J. Yang (Eds.): WAIM 2005, LNCS 3739, pp. 682 – 687, 2005.

is that they arrange keys (such as lookup items, files, services, etc) and peer nodes in the same identifier space.

Although all these P2P networks work efficiently in the overlay network level, they neglect the real geographic layout more or less. So to some extent, these overlay networks are not really efficient in practice. In general, P2P network is a kind of overlay networks built on existing physical network. Thus, the structure of overlay and physical topology both affect the performance of the whole P2P network. Some P2P overlay network mentioned above doesn't care much about the topology of network.

Some researches [6] have revealed the importance of the topology of physical network, and propose topology-aware overlay network. In order to achieve good performance, the topology-aware overlay network care both the organization of the overlay network and consider some factors of the physical network.

Geographic layout [8] was explored as one topology-aware technique to improve the routing performance in CAN. The technique attempts to map the d-dimensional space onto the physical network such that nodes that are neighbors in the d dimensional space are close in the physical network. This technique can achieve good performance but has the disadvantage that it is not fully self-organizing; it requires a set of well-known landmark servers. In addition, it may cause significant imbalances in the distribution of nodes in the CAN space, leading to hot-spots.

Proximity routing [9] is another kind of topology aware routing. With proximity routing, the overlay is constructed without regard for the physical network topology. The technique exploits the fact that when a message is routed, there are potentially several possible next hop neighbors that are closer to the message's key in the id space. The idea is to select, among the possible next hops, the one that is closest in the physical network or one that represents a good compromise between progress in the id space and proximity. But it will increase the overhead of node maintenance and the size of routing tables. Proximity routing offers some improvement in routing performance, but this improvement is limited by the fact that a small number of nodes sampled from specific portions of the node ID space are not likely to be among the nodes that are closest in the network topology.

Miguel Castro [7] presents a kind of proximity neighbor selection to build structured P2P overlay network with topology-aware routing. Its routing algorithm makes the identifier in routing table point to the close node in physical network. Proximity neighbor selection can improve the performance of the P2P network based on matching, such as Tapestry and Pastry. In Tapestry and Pastry, a message is normally forwarded in each routing step to a nearby node, according to the proximity metric, among all nodes whose node ID shares a longer prefix with the key. Moreover, the expected distance traveled in each consecutive routing step increases exponentially, because the density of nodes decreases exponentially with the length of the prefix match. The routing algorithms in Pastry and Tapestry claim that they allow effective proximity neighbor selection because there is freedom to choose nearby routing table entries from among a large set of nodes.

# 3 DAPS DESIGN

## 3.1 Overview of DAPS

DAPS (Delay Aware P2P System) is not a comprehensive peer-to-peer system, but a kind of solution to the problem of topology-aware routing. Its goal is to reduce the time of L for a lookup request and improve the total performance of the P2P system. The main idea is that routing table is divided into several sectors according to delay from low to high, and source node will define a delay boundary, namely, the pruning factor, $L_t$. Request messages will only send to the nodes whose delay is not more than $L_t$. Compared with traditional flooding, the request message is largely reduced and realize "pruning flood" with the help of its routing table.

The overlay network of DAPS organizes loosely and sends routing messages with being aware of the conditions of the physical network. Considered the factors affect the total performance, "pruning flood" can lower the network traffic, reduce the complexity of locating algorithm and find expected results efficiently.

With clustered entry in the routing table and the loose organization, the overlay network of DAPS is between structured and unstructured. So, it supports partial-match query and nearly does not need to care much about when the nodes join, leave or fail in the network. And its routing table ensures the adoption to the change of network.

## 3.2 Routing Table

Each DAPS node maintains a routing table which indicated where to find the destination of lookup request. The structure of routing table is organized on the base of delay, which is divided into clusters from low to high. Nodes in the same clusters have the same range of delay.

If the range of delay is divided into $\sigma_1$, $\sigma_2 \ldots \sigma_i$, and $0<\sigma_1<\sigma_2<\ldots\sigma_i$ .The *n*th row of the routing table contains the nodes whose delay is between $\sigma_{n-1}$ and $\sigma_n$,.... In current research, the delay range is separated in same interval, $\sigma_i = i\times\sigma_1$. Each entry records the IP address of the node. So, the routing table arranges its neighbor nodes into several sectors according the delay between them.

## 3.3 Locating and Routing

In some unstructured P2P system, flooding is a simple solution to route request messages. The search will be iterative until the request is satisfied, or the maximum depth limit has been reached. Because the number of nodes at each depth grows exponentially, the cost of query will be multiplied. In order to lower the cost and reduce the number of visited nodes, DAPS uses "pruning flood" to route request.

Pruning flood is an iterative deepening, multiple breadth-first search with a pruning boundary. Pruning flood is implemented as follows: first, source node will define a delay boundary, namely, the pruning factor, $L_t$. It means the request must be satisfied in delay of $L_t$, if the results can be found. Then, the source node will only send requests to the nodes in the routing table where $\sigma_i \le L_t$, and then $L_t \leftarrow L_t - \sigma_i$. The

requested node then receives and processes the message. If the node has the expected result, it stops the query immediately and returns results to the source node. Otherwise the node will resend the request to the nodes in its routing table where $\sigma_i \leq L_t$. Just like above, the procedure will repeat until $L_t < \sigma$. Neighbor nodes are organized by the time of delay in the routing table, so the number of visited node can be greatly reduced and locates the destination node in physical network not more than $L_t$ or far less.

Assume that there exist *N* nodes in the network and each node records *N/r* nodes in its routing table averagely. *r* is the scale parameter. It means each node knows *1/r* nodes in the whole network. The routing table is divided into q sectors according to the delay range. So there are *N/ (qr)* nodes in each sectors in average. If $L_t = k\sigma$, in the worst condition, the number of request message that nodes totally send in one lookup procedure will be $(N/qr)\ (N/qr+1)^{k-1}$.

And, M~ O $((N/qr)^{k})$, M is the number of request message. In the traditional unstructured P2P system, such as Gnutella, it sends message by flooding. The number of its request messages will be O ($N^{t}$), t is the TTL of request messages. As it can be seen, if we appropriately choose the parameters of q, r, k, the message number can be largely reduced, and realize "pruning flood".

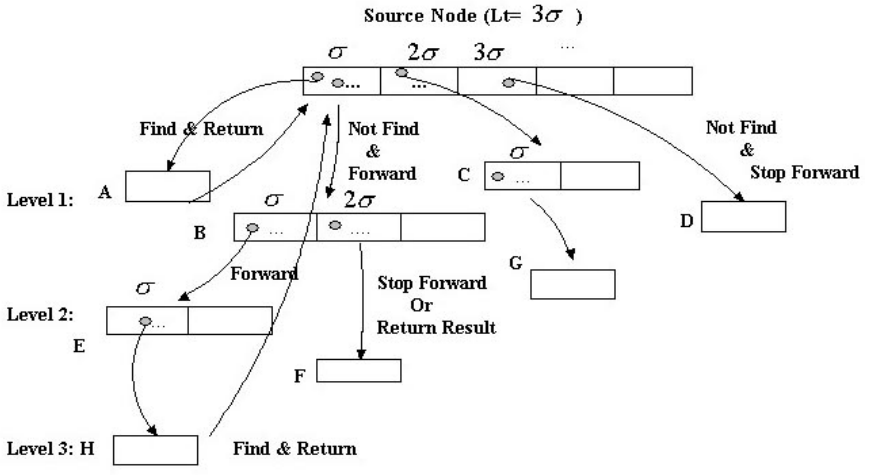


**Fig .1.** Picture describes the procedure of lookup and routing message with a pruning boundary

The source node starts looking up request with boundary $L_t = 3\sigma$. At first, it only sends request message to nodes whose delay is not more than $3\sigma$. When message reaches node A and finds expected results, it will stop searching and return results at once. Node B receives the request but not find the expected result, so it will forward the request to the node whose delay range is not more than $2\sigma$ ($2\sigma = L_t - \sigma$) in its routing table. Again, the request will send to E and F. Not finding result and not exceeding its pruning boundary, E forwards request to H where the result is found and

return it to the source node. As for node F, if there doesn't exist expected results, node F have to exit its request process, as the pruning factor has reached its boundary ( $\sigma + 2\sigma = L_t$). Node C, G and D will process respectively.

The algorithm of DAPS is a kind of proximity routing. It adopts end-to-end delay as the performance metrics, as it is more sensitive to user. When searching a request, DAPS select a set of nodes with low delay. It may increase hops in overlay network, but it delivers the lowest delay to user as much as possible.

## 4 Evaluation

We test the algorithms by simulation on GT-ITM [10] Internet. Figure 2 shows the quality of DAPS routing algorithms. With the increase of pruning factor, the percentage of successful lookup is increasing fast. Nearly about 90% expected results can be found in the delay range (pruning factor) of 800.Further simulation results reveal that we can find nearly all the results with the increase of $L_t$.

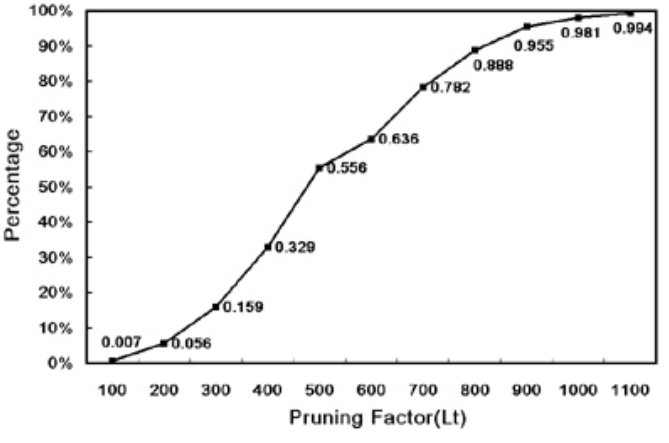


**Fig. 2.** Picture describes the percentage of successful lookup

## 5 Conclusion

In most P2P systems, the topology of physical network is often inconsistent with that of overlay network, which may lower the efficiency the routing algorithm. Based on this reason, we propose DAPS which sends request messages by pruning flooding. Compared with some unstructured P2P system, DAPS can efficiently find expected results in the given delay time and the number of request message is largely reduced.

The overlay of DAPS is loosely structured so that the maintenance is much easier and simpler than that of the structured P2P system. DAPS uses pruning flood to send messages, so the number of request messages is still more than that of structured systems which often use DHT to eliminate flooding. So DAPS is not an ideal solution in the large network currently, but the search is more flexible than structured system,

as it supports partial match. In the future work, we will combine pruning flood and DHT into DAPS so as to reduce the request number and improve its routing efficiency further.

## References


1. A. Rowstron and P. Druschel, Pastry: Scalable, distributed object location and routing for large-scale peer-to-peer systems. Accepted for Middleware, November 2001.
2. C. Greg Plaxton, Rajmohan Rajaraman, Andréa W. Richa. Accessing nearby copies of replicated objects in a distributed environment. ACM Press New York, NY, USA Pages: 311-320 Series-Proceeding-Articles. 1997. ISBN: 0-89791-890-8.
3. John Kubiatowicz, David Bindel, et al. OceanStore: An Architecture for Global-Scale Persistent Storage. In Proceedings of the Ninth international Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS 2000), November 2000.
4. Ion Stoica, Robert Morris, David Karger, M. Frans Kaashoek, and Hari Balakrishnan. *Chord: A Scalable Peer-to-peer Lookup Service for Internet Applications*, ACM SIGCOMM 2001, San Diego, CA, August 2001.
5. Sylvia Ratnasamy, Paul Francis, Mark Handley, Richard Karp, Scott Shenker. *A Scalable Content-Addressable Network*. In Proceedings of the ACM SIGCOMM, 2001.
6. Z. Xu, C. Tang, and Z. Zhang. Building topology-aware overlays using global soft-state. Technical Report HPL-2002-281, HP Labs, September 2002. Submitted for publication, available at http: //www. cs.rochester. edu/u/salTmor.
7. M. Castro, P. Druschel, Y. C. Hu, and A. Rowstron, Topology aware routing in structured peer-to-peer overlay networks.Tech. Rep. MSR-TR-2002-82, Microsoft Research, One Microsoft Way, Redmond, WA 98052, 2002.
8. S. Ratnasamy, M. Handley, R. Karp, and S. Shenker, "Topologically aware overlay construction and server selection," in Proceedings of IEEE INFOCOM'02, New York, NY, June 2002.
9. Roberto Rinaldi. Routing and data location in overlay peer-to-peer networks. Diploma thesis, Institut Eurecom and Universita degli Studi di Milano, June 2002. Also available as IBM Research Report RZ-3433.
10. E. W. Zegura, K. L. Calvert, and S. Bhattacharjee. How to model an internet work. In Proceedings of INFOCOM 96, 1996. Scenario Mig Prob New Part Life Part Source