Proximity Based Peer-to-Peer Overlay Networks (P3ON) with Load Distribution

Kunwoo Park[1], Sangheon Pack[2], and Taekyoung Kwon[1]

[1] School of Computer Engineering, Seoul National University, Seoul, Korea
kwpark@mmlab.snu.ac.kr, tkkwon@snu.ac.kr
[2] School of Electrical Engineering, Korea University, Seoul, Korea
shpack@korea.ac.kr

Abstract. Construction of overlay networks without any consideration of real network topologies causes inefficient routing in peer-to-peer networks. This paper presents the design and evaluation of a *proximity* based peer-to-peer overlay network (P3ON). P3ON is composed of *two-tier* overlay rings. The high tier ring is a global overlay in which every node participates. Whereas, the low tier ring is a local overlay that consists of nodes in the same autonomous system (AS). Since the low tier ring consists of nearby nodes (in the same AS), the lookup latency can be significantly reduced if the first search within the low tier ring is successful. Also, to cope with skewness of load (of key lookup) distribution, P3ON effectively replicates the popular keys (and results) to neighbor nodes and neighbor ASs. Simulation results reveal that P3ON outperforms the existing ring-based P2P network in terms of lookup time and achieves relatively balanced load distribution.

Keywords: proximity, peer-to-peer, overlay network, load distribution.

1 Introduction

Recently several peer-to-peer (P2P) systems have been proposed to overcome the limitations of the traditional client-server model. P2P systems distribute functionality and share resources among peers. Depending on how to locate resources, P2P systems can be classified into two classes: unstructured and structured. Generally, in unstructured P2P systems, peers are unaware of how resources are located in the overlay networks. Therefore, lookup requests are typically resolved by flooding-like techniques. Gnutella [1] is a well-known unstructured P2P system. Due to the flooding technique, unstructured P2P systems incur a high volume of signaling traffic. On the contrary, in structured P2P systems, peers share the way in which resources are located. Thus, lookup requests can be directed to a specific peer and hence, much fewer lookup messages are needed. However, structured P2P systems require increased maintenance cost incurred by maintaining the overall structure. The most prominent approach in structured P2P systems is to use a distributed hash table (DHT) to locate resources.

In the literature, a number of DHT-based lookup algorithms have been proposed, [2][3][4]. The lookup time that of most of these algorithms is approximately bounded to log(N), where N is the number of nodes. However, the hop

T. Vazão, M.M. Freire, and I. Chong (Eds.): ICOIN 2007, LNCS 5200, pp. 234–243, 2008.

distance between two overlay nodes in the overlay network has nothing to do with the real distance between two nodes. To overcome this inefficient lookup problem, geographical proximity-based routing (i.e. proximity based neighbor selection) is proposed in P2P systems. Pastry [4] is a well-known proximity based algorithm. Since peers build their routing table entries depending on the proximity metric among all nodes in the network, a huge amount of control messages are required to measure proximity especially when a node joins the network.

There are a few attempts to achieve better lookup performance by adding an additional overlay in the system. Brocade [5] utilizes a new layer consists of supernodes, which are powerful nodes close to network access points such as routers. Each supernode manages a group of local nodes and every local nodes access resources via supernodes. The network traffic is reduced but a supernode may become the bottleneck. Plethora [6] organizes nodes into local overlays leveraging autonomous system (AS) information. Using cache in the local overlay significantly reduces the lookup latency. However, local overlay leaders, each of which uniquely exists in each local overlay are responsible for AS merge/split to keep the number of nodes in an AS appropriately.

In this paper, we propose a Proximity based P2P Overlay Network (P3ON). P3ON is composed of two overlays: high tier and low tier. The high tier ring is a global overlay, which includes every node participating in P3ON. In contrast to this ring, the low tier ring is a local overlay, which represents a single autonomous system (AS). That is, all nodes in an AS belong to the same low tier ring. We present a two-phase lookup algorithm to take advantage of local cache. Even if the input query is highly skewed, the overhead at a popular node is effectively distributed by a load distribution mechanism.

The rest of this paper is organized as follows. Section 2 details P3ON. Section 3 shows the numerical results of our system and Section 4 concludes this paper.

2 Proximity Based P2P Networks (P3ON)

If the hop distance in overlay is based on the proximity (e.g. geographical distance) between two nodes and there exists any semantic locality (the popular item will be looked up again by others) among the lookup queries, the lookup time in large-scale P2P networks will be substantially lowered [7]. To accomplish this, we design two decentralized algorithms: the proximity-based ID assignment algorithm and the two-phase lookup algorithm.

2.1 Proximity Based ID Assignment

We first assume that every node in P3ON possesses a unique IP address. By hashing the node's IP address in a collision-resistant manner (e.g. SHA-1, MD5), P2P systems obtain asymptotically almost a unique ID. However, the hash value does not reflect any proximity between the peer nodes. Therefore, P3ON proposes the following hierarchical ID assignment algorithm after mentioning our second assumption.

Our next assumption is that a node is feasible to figure out its AS number (e.g. [8][9]) and identical AS number (ASN) is assigned to all the nodes that belong to the same AS. The AS is usually a group of nodes governed by a single authority and in many cases, nodes in a same AS are closely located. IDs in P3ON are selected from a 176bit namespace. Since every node has a 16 bit AS number representing to which AS it belongs to, the first 16 bits of a node ID are adopted from its own ASN. The remaining 160 bits of the ID are determined by hashing the node's IP address with the SHA-1 algorithm. By concatenating those 16 bits (ASN) and 160 bits (SHA-1 value), we obtain a unique and uniformly distributed node ID, which namespace is 176 bits long. To utilize a distributed hash table (DHT), an item ID must be the same length as a node ID. Therefore an item ID must be also 176 bits long. The latter 160 bits of the item ID are derived from the hashing result of SHA-1 with its item name. As items do not have any similar concepts such as ASN, we prepend additional 16 bits by copying the last 16 bits of the latter 160 bits. Consequently, the node and item IDs are constructed as follows;

$$\text{Node ID} \quad = \quad (\text{ASN}) \mid\mid f(\text{node}'\text{s IP address})$$

$$\text{Item ID} \quad = \quad (\text{last 16 bits of } f(\text{item name})) \mid\mid f(\text{node}'\text{s IP address})$$

where f is a SHA-1 hash function

When constructing the item ID with 176 bits, we have two factors in mind. First, any ID constructed this way is unique. If two distinct items have different item names, the uniqueness is guaranteed by the property of SHA-1. Also, items with the same item name are mapped to an identical ID. Second, item IDs are well distributed over the 176 bit namespace. We perform a simple experiment to verify that item IDs in our scheme are evenly distributed. In our experiment, we first uniformly distributed 500 nodes in 176 bit namespace, and then distributed 1,000,000 items with IDs generated by our scheme. In an ideal case, if K items are uniformly distributed over uniformly distributed N nodes, K/N items will be located at each node.

2.2 Two Tier Ring

Figure 1 illustrates a two-tier ring in P3ON. The high tier ring is the main overlay in P3ON; therefore, every peer node is mapped to a position over the high tier ring. Owing to the proximity-based ID assignment algorithm, nodes, which are closely located in the same AS, are placed adjacently in the high tier ring. The high tier ring is partitioned into AS units. Note that unused ASNs will generate the empty node ID space. The keys corresponding to this empty space will be mapped to an immediate predecessor node. Since nodes in the same AS have the same ASN, the first 16 bits of those node's ID are identical. Therefore, those nodes are placed in the nearby area in the high tier ring naturally.

At the same time, a low tier ring is built by grouping the nodes in the same AS. As a result, the number of low tier rings equals to the number of participating ASs. To form a low tier ring, an additional link per AS is required. This link

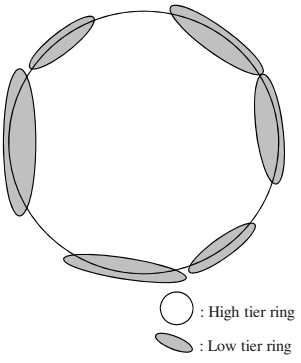


Fig. 1. Two-tier overlay

will connect two peer nodes with the highest and lowest IDs in the AS. A node whose ASN is different from the ASN of its predecessor realizes that it is the node with lowest ID within the AS and it has a responsibility to maintain the link to the node with the highest ID to form the low tier ring. This node queries a node with the highest ID and sets up a connection. As each node belongs to two different rings, high tier and low tier rings, finger tables must be maintained separately. Forming each finger table is identical with the process of Chord [2].

2.3 Two-Phase Lookup Algorithm

Phase I: Low Tier Ring Lookup. At first, a node tries to find the key within its low tier ring. To do this, the node creates a local item ID for the item. The local item ID is created by concatenating the 16 bit ASN of the node and the 160 bit result of SHA-1 with the item name. Initially, keys are located only in the high tier ring. Therefore, a cache miss will occur for the first lookup process searching the item in the low tier ring.

If a peer node that corresponds to the local item ID in the low tier ring (we call this node a local target node) stores the previously queried keys, the local target node can respond to the query from then on. To keep the previously queried keys in the low tier ring, each peer node maintains a local cache. A cache entry consists of item ID and the position of the item, the latest query originator. The latest query originator can give the location information about the item itself, since the originator will possess the item after lookup. The next node search for the same item can download it within the same AS. In the case of cache overflow, a famous replacement policy, such as least recently unused (LRU) can be used. If the local target node does not have any information about the query, the second phase lookup procedure is performed by the local target node.

Phase II: High Tier Ring Lookup. In the second phase lookup procedure, the local target node acts as a query originator. The local target node uses the original item ID described in Section 2.1 instead of the local item ID. By using

this original item ID, the same lookup procedure as Chord is performed at the high tier ring. Since all items and nodes are located at the high tier-overlay ring, there is no possibility of lookup failure. The query response is delivered towards the local target node and the local target node relays the result to the original query initiator. When the local target node returns the query result to the query originator, it stores the local item ID and the IP address of the query originator in each local cache. Therefore, if any other node in the same AS tries to find the same item later, a query can be responded to the query originator with reduced delay.

2.4 Load Distribution

In an ideal P2P condition, all the nodes in the system should experience the same amount of lookup load. Here, the load is two-fold, i) the number of queries that the node receives in a unit time, ii) the number of keys that the node stores. Keys stored at a node occupy the storage space proportional to the number of keys. Incoming query consumes the node's computation power, network bandwidth, etc. In general, nodes with the relatively large number of keys tend to receive more queries than other nodes. Although these two aspects of the load have some correlations, it is feasible to decouple them. The reason is that in a real world, queries are not uniformly distributed among individual items, rather, the distribution of queries are highly skewed [10]. For example, a node with the most popular item can be overloaded by too many incoming queries.

Since recent machines have a sufficient storage space for small sized cache entry, the number of keys is not an issue. In P3ON, to cope with this problem, a dynamic load distribution using two thresholds (δ_1, δ_2) is proposed for load balance. Since each node has different power and link bandwidth, each node uses different thresholds. The central idea is that if the load (the number of queries per unit time) exceeds a certain threshold, the node triggers load distribution. Specifically, a node counts the number of times each key is referred. It is possible by simply modifying the structure of a cache entry by adding an additional field for counting. Keys with frequent access will make the node overloaded. When the number of incoming queries on all the keys of a node exceeds a pre-defined

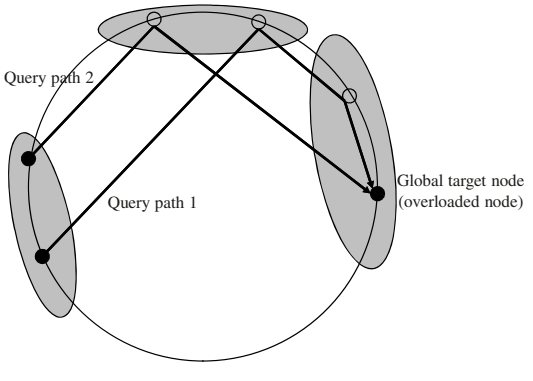


Fig. 2. Load distribution

threshold δ_1 of the node, the node sends out an intra-AS advertisement message and the message is delivered to all the nodes in the local ring (i.e. nodes in the same AS). The message contains the information on the most popular keys of the overloaded node.

When a new lookup process for the key in the above overloaded node is initiated, the key can be found in predecessor nodes before the query reaches the overloaded node. Intra-AS advertisement deals with queries routed over query path1 as shown in the Figure 2 which traverses via the low tier ring. If the size of low tier ring is small (i.e. the number of nodes in the AS is low), the ratio of queries routed over query path2 in Figure 2 will increase. However intra-AS advertisement cannot handle queries over query path2.

Although the overloaded node triggers intra-AS advertisement, the node can still be suffered from too many incoming queries. If the incoming query frequency exceeds the pre-defined threshold, δ_2, inter-AS advertisement is triggered. The intra-AS advertisement is relayed toward the predecessor AS. The message arriving at the predecessor AS will be delivered to all the nodes in that AS as if intra-AS advertisement is triggered. Both inter-AS and intra-AS advertisements distribute queries to mitigate the burden of the overloaded node.

3 Numerical Results

In this section, we evaluate the performance of P3ON in terms of the size of the network (total number of nodes in overlay) and the size of each AS (the number of nodes in each AS). The central idea of P3ON lies in exploiting the use of proximity between nodes. Therefore, the size of AS significantly affects lookup time and load distribution.

3.1 Simulation Parameters

We simulated a network that has up to 10,000 nodes, and in every network layout, 100,000 keys are distributed over the network. The low tier ring has a local cache with the size of 10 slots. That is, there are 10 keys and their locations in the cache of a node. Clearly, the more slots are provided in a cache, the better performance is achieved. However, for the purpose of emphasizing the impact of the local ring, we restrict the cache size to 10. Query initiators are

Table 1. One way delay between transit nodes

Continents	America	Latin	Europe	Asia	Africa	Oceania
Ameria						
Latin	222					
Europe	80	156				
Asia	125	237	159			
Africa	392	326	358	284		
Oceania	136	249	177	198	296	

chosen randomly among the entire nodes, and items to be queried are selected according to a Zipf-like distribution with an input parameter of $\alpha = 1.0$. We use an abstracted world topology. There are 6 continents and one representative transit node exists for each continent. As shown in Table 1, the latency between each transit node is based on the average of measure values during the period between August 2003 and June 2005 measured by IEPM [11]. Each AS belongs to one of the continents, and the delay to the transit node takes $[0 - 50]$ms. Up to 2000 nodes for each AS are deployed and an intra-AS delay of $[0 - 10]$ms is established in our experiments. For the parameters we refer to [11][12]. The stated parameters are used for our results, unless otherwise explicitly stated.

3.2 Lookup Latency

To verify the performance of P3ON, we measured the lookup latency with a virtual network, which varies in size by 250, 500, 1000, 2000, 4000, 8000 and 16000 nodes. Since P3ON is significantly affected by the AS size and the number of ASs, experiments with different numbers of ASs and different numbers of nodes in each AS are as important as experiments with the total number of the nodes being changed. Therefore, we increased the network size with two different ways. First, we fix the number of nodes in each AS to 50, and we increase the number of ASs as follows: 5(250), 10(500), 20(1000), 40(2000), 80(4000), 160(8000) and 320(16000). The values in the parentheses stand for the network size (total number of nodes in overlay). Second, we now fix the numbers of ASs to 50, and change the AS size as follows: 5(250), 10(500), 20(1000), 40(2000), 80(4000), 160(8000) and 320(16000).

Figure 3 shows the result of our experiments. Since P3ON is an enhanced version of the Chord's lookup algorithm, it is natural that P3ON outperforms Chord. We can see the first way of increasing the AS size with the fixed number of ASs maintains the lookup latency around $600 - 800$ ms. When the AS size is too small, i.e. 5 nodes, the total cache slots in the AS is 50, so that P3ON performs worse than Chord due to frequent cache miss. In this situation, the

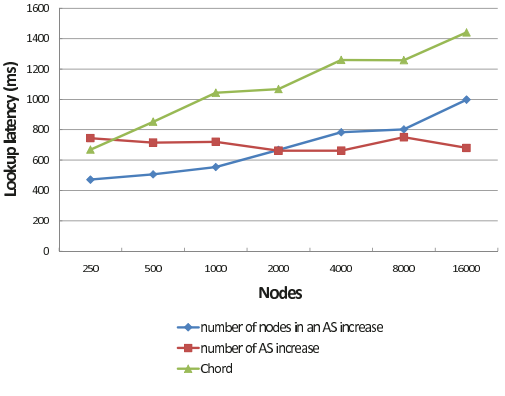
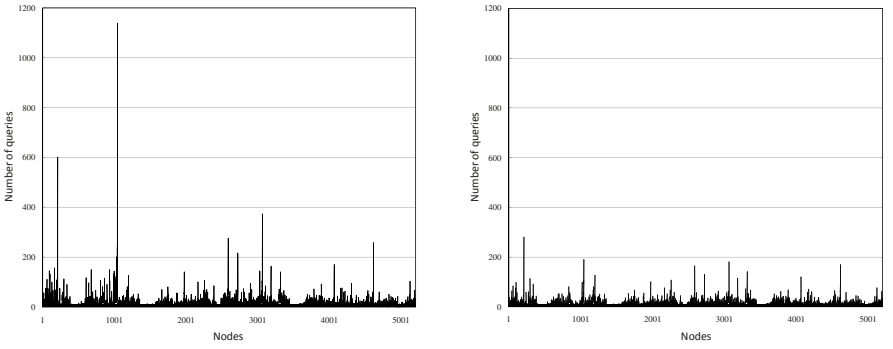


Fig. 3. Lookup latency versus network size (ms)



(a) Before load distribution (b) After load distribution

Fig. 4. Number of queries reached at each node before and after the load distribution

Table 2. Effect of load distribution

	before	after
Total number of query request	8999	6742
Average	1.73	1.30
Maximum	1133	271
Standard deviation	20.50	8.37

delay consumed by the first phase of two-phase lookup algorithm is only a burden to the system. The delay is beneficial when the cache size becomes bigger than 7. The lookup latency fixed around 600-800 is the counterbalance between effect of cache and network size increase. However, the lookup latency of the second way slowly increases as the number of ASs increases, with the fixed AS size. This can be explained as follows. Through the two-phase lookup algorithm, a node is able to utilize the whole information cached at all the nodes in the same low tier ring. Since a low tier ring corresponds to one AS, increasing the number of ASs does not increase the cache hit probability within a single AS.

3.3 Load Distribution

In this experiment, we reveal how the entire workload (the number of lookup requests) is distributed among all the nodes in the overlay. Figure 4 shows the amount of load at every node with the definition of the term 'load' at Section 2.4. Since queries are skewed, a small portion of the nodes dominates the target of the lookup requests before load distribution. The difference in the total number of query requests in Table 2 is due to the cache in each node. If the target key is in its own cache, the node does not initiate the lookup process at all.

Figure 4 shows the impact of load distribution in P3ON. Through the local cache, intra-AS, and inter-AS advertisement messages, the overloaded node's load is significantly reduced. Each local ring reduces a significant amount of load by facilitating the local cache. Query requests that take place due to cache

misses are mostly filtered by the predecessor nodes of the overloaded node. After the load distribution process is in effect, the predecessor nodes experience slightly more load than before. If the predecessor node's capacity is insufficient to handle this additional load, it becomes an overloaded node and triggers its own advertisement to reduce the load. By that domino effect, the lookup load is efficiently distributed in a fully decentralized manner.

The performance of the proposed load distribution algorithm is affected by several factors as follows: the place where the overloaded node is located within the low tier ring, the size of the low tier ring, and the size of predecessor AS. If there are a lot of predecessors of the overloaded node in the low tier ring, the load will be distributed to the predecessors due to the intra-AS advertisement message. Likewise, if the predecessor AS has a number of nodes therein, they will also help mitigate the load of the overloaded node.

4 Conclusion

Peer-to-peer (P2P) overlay networks should be carefully constructed to reduce the lookup latency, especially taking into account real network topologies. Most of P2P protocols and systems have focused on how to reduce the hop distance in lookup operations. However, even a single hop in overlay networks can reach far more than 10 hops in real networking environments. In this paper, we propose a proximity based peer-to-peer overlay network (P3ON), which is a fast, scalable lookup algorithm. P3ON is composed of two overlays. The high tier ring is a global overlay in which every node participates. Whereas, the low tier ring is a local overlay that consists of nodes in the same autonomous system (AS). Since the low tier ring consists of nearby nodes (in the same AS), the lookup latency can be significantly reduced if the first search within the low tier ring is successful. To this end, previously queried keys and results (locations of keys) are stored in the corresponding node of the low tier ring. Also, to cope with skewness of load (of key lookup) distribution, P3ON effectively replicates the popular keys (and results) to neighbor nodes and neighbor ASs. This replication is realized by two advertisement messages: intra-AS advertisement and inter-AS advertisement. Simulation results reveal that P3ON outperforms the existing ring-based P2P network (i.e. Chord) in terms of lookup time and achieves relatively balanced load distribution. Since the real (underlying) network topology is a key issue in determining the performance of P3ON, we are currently working on more realistic experiments.

References

1. Gnutella, `http://www.gnutella.com`
2. Stoica, I., Morris, R., Liben-Nowell, D., Karger, D.R., Kaashoek, M.F., Dabek, F., Balakrishnan, H.: Chord: A Scalable Peer-to-peer Lookup Protocol for Internet Applications. IEEE/ACM Transactions on Networking 11(1), 17–32 (2003)
3. Ratnasamy, S., Francis, P., Handley, M., Karp, R., Shenker, S.: A scalable content-addressable network. In: Proc. ACM SIGCOMM 2001, pp. 161–172 (August 2001)

4. Rowstron, A., Druschel, P.: Pastry: Scalable, distributed object location and routing for large-scale peer-to-peer systems. In: Guerraoui, R. (ed.) Middleware 2001. LNCS, vol. 2218, pp. 329–350. Springer, Heidelberg (2001)
5. Zhao, B.Y., Duan, Y., Huang, L., Joseph, A.D., Kubiatowicz, J.D.: Brocade: landmark routing on overlay networks. In: Druschel, P., Kaashoek, M.F., Rowstron, A. (eds.) IPTPS 2002. LNCS, vol. 2429, pp. 34–44. Springer, Heidelberg (2002)
6. Ferreira, R.A., Grama, A., Jagannathan, S.: Enhancing Locality in Structured Peer-to-Peer Networks. In: Proceedings of Tenth IEEE International Conference on Parallel and Distributed Systems, Newport Beach, CA, July 2004, pp. 25–34 (2004)
7. Gummadi, K.P., Dunn, R.J., Saroiu, S., Gribble, S.D., Levy, H.M., Zahorjan, J.: Measurement, Modeling, and Analysis of a Peer-to-Peer File-Sharing Workload. In: Proc. of the 19th ACM Symposium on Operating Systems Principles, Bolton Landing, NY (October 2003)
8. Mao, Z.M., Rexford, J., Wang, J., Katz, R.H.: Towards an Accurate AS-Level Traceroute Tool. In: Proceedings of the 2003 ACM SIGCOMM Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication, Karlsruhe, Germany (August 2003)
9. Exploiting Autonomous System Information in Structured Peer-to-Peer Networks. In: The 13th IEEE International Conference on Computer Communications and Networks (ICCCN 2004), Chicago, IL, October 11-13 (2004)
10. Ge, Z., Figueiredo, D.R., Jaiswal, S., Kurose, J., Towsley, D.: Modeling peer-peer file sharing systems. In: Proceedings of INFOCOM 2003, Santa Fe, NM (October 2003)
11. Internet End-to-end Performance Monitoring (IEPM), `http://www-iepm.slac.stanford.edu/`
12. Xu, Z., Mahalingam, M., Karlsson, M.: Turning Heterogeneity into an Advantage in Overlay Routing. In: Proceedings of the IEEE INFOCOM 2003, San Francisco, CA (April 2003)