Constructing Searchable P2P Network with Randomly Selected Long-Distance Connections

Jingbo Shen, Jinlong Li, and Xufa Wang

Department of Computer Science and Technology,
University of Science and Technology of China,
Hefei, Anhui, 230026, P.R. China
zxfsjb@mail.ustc.edu.cn, {jlli,xfwang}@ustc.edu.cn

Abstract. Object lookup is a basic problem in P2P network. Long-distance connections have been used to construct a searchable P2P network according to the small-world paradigm. Long-distance connections based on distance can achieve a searchable P2P network in theorem. However, it is hard to measure the lattice distance between two peers in real P2P network. On the other hand, it is easy to construct randomly selected long-distance connections with low overhead. We increase the number of randomly selected long-distance connections *k* to improve the performance of object lookup. Simulation results show there is some relation among *k*, the network size *N* and the average path length. The lower bound of *k* to achieve a searchable P2P network is still an open question.

Keywords: Searchable P2P Network, Small-World Paradigm, Long-Distance Connection, Random Selection, Object Lookup.

1 P2P Network with Long-Distance Connections

Peer-to-peer (P2P) overlay networks are used for file sharing in recent years [1][2]. In P2P network, each peer only keeps in connection with some peers in its routing table without any information of the global network, and uses these peers to locate target peers and finds desired content. So P2P networks can be thought as complex networks. The topology of P2P networks can be modeled by directed graph, where nodes represent peers and edges represent connections between peers. Each node only has some short-range edges and a few long-range edges.

A basic problem in P2P network is how to find the desired content from a large number of peers rapidly and efficiently [3]. When a peer receives a search request, the peer has to forward the request to some neighbor using only local information, if it is not the target. The process is called *resource discovery* or *object lookup*. Greedy algorithm is always used for object lookup in P2P network. Each peer forwards the search request to its neighbor which is the nearest to the target. If any random peer can be found in *O(logN)* hops using a decentralized algorithm, the network is called a "*searchable network*", where *N* is the network size.

J. Zhou (Ed.): Complex 2009, Part II, LNICST 5, pp. 1860 – 1864, 2009.

In order to improve the performance of object lookup, researchers have paid their attention to complex network and found the small-world phenomenon which flashed into their minds. The small-world network has two attractive properties [4]: (1) a low average hop distance between any two randomly chosen peers, and (2) a high clustering coefficient. These properties can be used in P2P network to improve the performance of object lookup by building long-distance connections as shortcut. In recent years, there are many different kinds of methods to choose long-distance connections, such as random selection [5], selection based on distance [6], selection based on semantics [7], selection based on social dynamics and behaviors [8], and so on. Except for the method of selection based on distance, other methods can all be thought as random selection from topology.

Kleinberg proposed an algorithmic perspective about the small-world phenomenon in 2001 [9]. For a *d*-dimensional grid with *N* points, the *lattice distance* of lattice points *i* and *j* is defined as the number of "*lattice steps*" separating them, written as *d(i,j)*. Kleinberg proposed that there is a decentralized algorithm with the time complexity $O(\log^2 N)$, if each peer *u* has a long-distance connection to point *v* based on distance with the probability proportional to $[d(u,v)]^{-d}$.

CAN-SW (CAN using Small-World model) [10] used Kleinberg's theory to improve the performance of object lookup by build a long-distance connection based on distance for each peer on CAN [11] overlay. In order to choose long-distance connections, CAN-SW needs to know the network size and the topology of basic CAN overlay. However, it is hard to select the long-distance connections based on distance accurately in real P2P network for the following reasons.

(1) When a peer just joins a P2P network, it will only connect with its neighbors and can not know the distances from itself to all other peers. So it is hard for the peer to select a peer to build a long-distance connection based on distance.
(2) Though CAN-SW can choose the remote peer by using the network size and the topology of basic CAN overlay in simulation, it is not operable and accurate in a real P2P network. Because peers often join and leave P2P network frequently, the network size and the basic topology of a real P2P network are always in change.
(3) Besides, as the cost, additional messages are sent to discover remote peers, measure the distances and keep the connection with the selected remote peer. Whenever the selected remote peer failed, the same process has to be done again.

Above all, it is not a wise idea to build long-distance connections based on distance for a real P2P network. On the other hand, it is easy to build randomly selected long-distance connections for a real P2P network. So we want to construct a searchable P2P network by randomly selected long-distance connections.

2 Randomly Selected Long-Distance Connections

We can use search history to build a P2P network with randomly selected long-distance connections, which called *NLCH (P2P Network with Long-distance*

Connections based on History). We also use CAN as the basic grid. For simplicity, we assume that the topology is a *d*-dimensional torus. We build the long-distance connection after each successful search and use the search result to build a long-distance connection from the search source to the search target. Each peer can keep *k* long-distance connections at most. Each peer uses the *Least Recently Used (LRU)* Algorithm to update its long-distance connections.

Each peer uses a decentralized greedy routing algorithm for object lookup. Whenever received a search request, the peer will check all its *remote neighbors* which are connected by long-distance connections and all *nearby neighbors* which are connected by CAN connections, and choose the peer which is the nearest to the target as the next hop, as shown in Fig. 1.

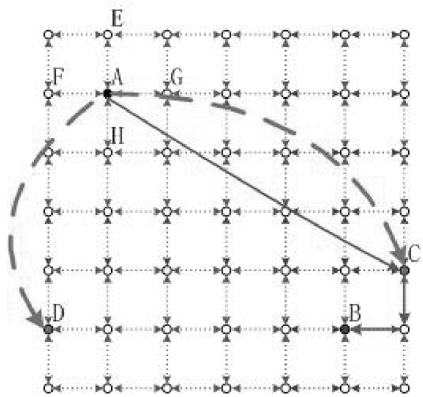
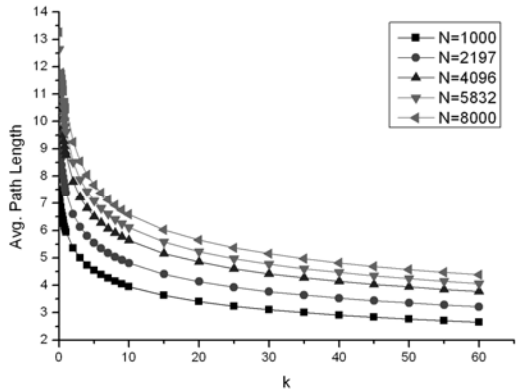


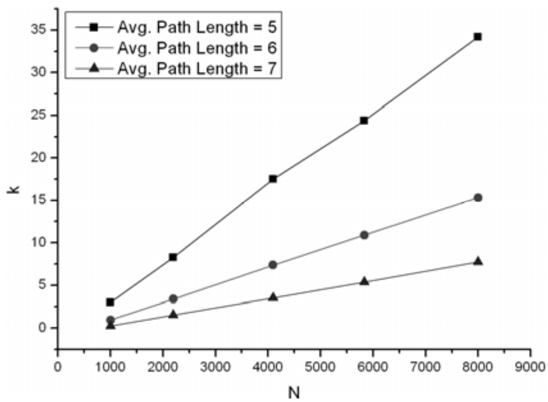
Fig. 1. This is an illustration of the topology in NLCH with *d=2*. Peer *E, F, G* and *H* are four nearby neighbors of Peer *A*, while Peer *C* and *D* are two remote neighbors of Peer *A*. When Peer *A* wants to search for Peer *B*, it will find that Peer *C* is the nearest to Peer *B* in all its neighbors, and forward the search request to Peer *C*.

When a peer just enters the network, it will first use the CAN connections for object lookup. After each successful search, the peer can build or replace its long-distance connections automatically. If some of its long-distance connections failed, the peer can also replace the failed link with a new search result after the next successful search. So no additional traffic is needed to construct and maintain the topology.

We measure the average path length with different *k* and *N* in NLCH while *d=3*, as shown in Fig. 2(a). With the increase of *k*, results show the average path length can decrease rapidly in the first, and then slow down for different *N*. In order to achieve the same performance of object lookup, different *k* is needed for different *N*, as shown in Fig. 2(b). It seems the average path length can be kept in *O(1)* with a linear relation between *k* and *N*.



(a)



(b)

Fig. 2. This is the relation among *k*, *N* and the average path length in NLCH with *d=3*

3 Conclusions and Open Questions

We want to construct a searchable P2P network by adding some long-distance connections to regular graph, such as a grid or a torus.

A searchable P2P network can be achieved by building a long-distance connection based on distance for each peer, according to Kleinberg's theory. The time complexity of object lookup is $O(log^2N)$. However, it is hard to select the long-distance connections based on distance exactly. Because the network size and the topology of a real P2P network are always in change, it is not operable and accurate to measure the lattice distance between any two random selected peers, with only local knowledge for each peer. Besides, even using the approximate distance to choose long-distance connections, additional overhead is needed to measure the distance, construct and maintain long-distance connections.

For above reasons, we prefer to use randomly selected long-distance connections to improve the performance of object lookup in real P2P network, and propose a P2P network with randomly selected long-distance connections called NLCH.

However, if the number of randomly selected long-distance connections k is a *fixed constant*, the time complexity of object lookup is $\Omega(N^{2/3d})$ in theorem according to Kleinberg's theory. Because it is easy to add randomly selected long-distance connections in real P2P network, we increase k to decrease the average path length. Simulation results show the average path length of object lookup can be kept as a constant, if k increases with N. Obviously, if k is large enough, the average path length can achieve $O(1)$; while if k is only a fixed constant, the expected minimum path length is $\Omega(N^{2/3d})$. There should be a critical value of k in $[1,\infty)$ called k_0, from which the average path length can decrease from $O(N^{2/3d})$ to $O(\log^{\alpha} N)$, while α is a constant. If $k>k_0$, we can construct a searchable P2P network by building randomly selected long-distance connections on a regular graph.

There are still some open questions which we are looking for solutions in complex network.

(1) If k is in $\Omega(N)$, the average path length can keep in $O(1)$. But is $\Omega(N)$ the tight lower bound of k to achieve the average path length of $O(1)$?
(2) How to set the lower bound of k in order to achieve a searchable P2P network by using randomly selected long-distance connections?

References

1. Gnutella, `http://gnutella.wego.com`
2. Freenet, `http://freenet.sourceforge.net`
3. Vanthournout, K., Deconinck, G., Belmans, R.: A taxonomy for resource discovery. Personal and Ubiquitous Computing 9, 81–89 (2005)
4. Newman, M.E.J., Watts, D.J.: Scaling and percolation in the small-world network model. Physical Review E 60, 7332–7342 (1999)
5. Hui, K.Y.K., Lui, J.C.S., Yau, D.K.Y.: Small-world overlay P2P networks: construction, management and handling of dynamic flash crowds. Computer Networks 50, 2727–2746 (2006)
6. Zhuge, H., Chen, X., Sun, X., Yao, E.: HRing: A Structured P2P Overlay Based on Harmonic Series. IEEE Transactions on Parallel and Distributed Systems 19, 145–158 (2008)
7. Li, M., Lee, W.C., Sivasubramaniam, A., Zhao, J.: SSW: A Small-World-Based Overlay for Peer-to-Peer Search. IEEE Transactions on Parallel and Distributed Systems 19, 735–749 (2008)
8. Carchiolo, V., Malgeri, M., Mangioni, G., Nicosia, V.: Emerging structures of P2P networks induced by social relationships. Computer Communications 31, 620–628 (2008)
9. Kleinberg, J.: The small-world phenomenon: an algorithm perspective. In: 32th Annual ACM Symposium on Theory of Computing, pp. 163–170. ACM Press, New York (2000)
10. Wang, S., Xuan, D., Zhao, W.: Analyzing and enhancing the resilience of structured peer-to-peer systems. J. Parallel Distrib. Comput. 65, 207–219 (2004)
11. Ratnasamy, S., Francis, P., Handley, M., Karp, R., Schenker, S.: A scalable content-addressable network. In: Proc. 2001 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications, pp. 161–172. ACM Press, New York (2001)