Chord6: IPv6 Based Topology-Aware Chord

Jiping Xiong*, Youwei Zhang*, Peilin Hong+, Jinsheng Li+
University of Science and Technology of China
**{xjping,ywz}@mail.ustc.edu.cn +{plhong,jinsheng}ustc.edu.cn*

Abstract

Chord, in its original design, does not take into account physical network topology, thus resulting in high routing latency and low efficiency. As the entire Internet is being currently upgraded from IPv4 to IPv6 worldwide, we devise a smart scheme to exploit the IPv6 address hierarchical feature, so as to construct an efficient version of Chord dubbed Chord6. Simulation results have shown that our method can significantly reduce inter-domain traffic that causes the long routing latency. Another advantage of our scheme is that it is very simple and barely modifies the original Chord.

1. Introduction

Due to their advantages of being scalable and resilient, Distribute Hash Table (DHT) systems like Chord [1], CAN [2], Pastry [3] and Tapestry [4] have been identified as promising substrates for future Internet-scale peer-to-peer applications. With the primary purpose of providing lookup service, DHTs associate files with corresponding keys, with all keys constituting a key space. Then each peer is assigned a segment of that space. Therefore, a file's location can be found when the peer responsible for the file's key is contacted. In general, most DHT systems can achieve path lengths of O (log n) hops, where n is the number of participating nodes. However, these are just application-level hops, not true end-to-end latencies that do matter in networks. Actually, since existing DHT systems like Chord are constructed with negligence of geographic proximity of participating nodes, the lookup process can be rather inefficient, with extra latency introduced by routing into and out of the same area/domain many times. Therefore, we argue that to reduce extra inter-domain hops will be a very effective way to improve DHT performance.

IPv6, the next generation standard of network protocol, defines a hierarchically structured address format to reflect the network clustering property [5, 6]. That means hosts with the same address prefix of specific length will reside in the same autonomous system and vice versa (e.g. a large ISP will be assigned an address space with the prefix length of 32). Fig.1 gives out one example of hierarchical IPv6 prefix address.

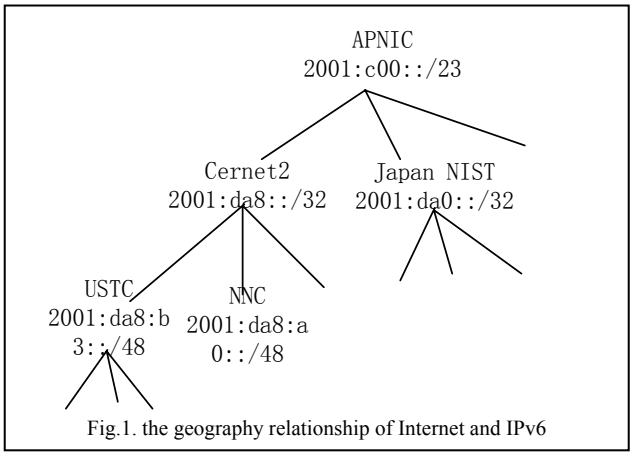

Fig.1. the geography relationship of Internet and IPv6

Our method does not need to assume the existence of landmark hosts to which each Chord nodes have to measure its round-trip-time. Besides, not like other solutions that require peers to maintain more states or introduce more message exchange, our solution is very simple and barely modifies the original Chord protocols. In fact, we only modify the way that node identifies are obtained, while the original procedures regarding routing and node's joining and leaving remain unaltered. In addition, our method can, with little modification, be applied to other DHT systems, such as CAN, Pastry and Tapestry. Throughout the paper, we will refer to a domain as a collection of IPv6 subnets with the same address prefix of specific length.

The rest of this paper is organized as follows. Section 2 describes Topology-aware Construction Mechanism; Section 3 gives out the performance benefits by simulation. We conclude in Section 4.

Proceedings of the Joint International Conference on Autonomic and Autonomous Systems
and International Conference on Networking and Services (ICAS/ICNS 2005)
0-7695-2450-8/05 $20.00 © 2005 IEEE

2. Topology aware construction mechanism

Basically, all keys in Chord are arranged in a one-dimensional circular key space, and each node obtains its node identifier by hashing its IP address. The node responsible for a key is the node whose identifier closely follows the key (numerically clockwise). For routing purpose, each node maintains a list of k neighbor nodes that immediately follow it in the key space. In order to achieve routing efficiency, each node must also keep another set of O (log n) nodes called fingers whose identifiers are spaced expo-nentially far from the node in the key space. With O (log n) hops, routing can be achieved by forwarding lookup request along the key space using finger and neighbor tables.

n bits	64-n bits	64 bits
global routing prefix	subnet ID	interface ID

Fig.2. the Standard Format of IPv6 Global Unicast Address

Since nodes in Chord get their node identifiers by hashing IPv4 addresses, nearby nodes with close node identifiers on the overlay can be geographically far away from one another. Thus, some application-level hops can involve long-distance links, resulting in high latency end-to-end paths. To improve efficiency of Chord, topology information should be incorporated while constructing the system. It is noticed that IPv6 used in next generation Internet boasts such advantages as a vast address space and an address hierarchically structured format. As showed in Figure 2, the routing prefix of an IPv6 Global Unicast Address is the value assigned to identify a domain/site (a cluster of subnets/links), the subnet ID is the identifier of a subnet within the domain, and the interface ID is a modified EUI-64 format as defined in RFC3513 [6]. It is obvious that the specified IPv6 address allocation scheme can be an ideal way to pro-vide useful topology information for the purpose of constructing our new Chord system.

In order to build our topology-away Chord, we slightly modify the process of producing node identifiers. In Chord6, a node identifier contains two parts: the higher bits are obtained by hashing the node's IPv6 address prefix of specific length, while the remaining lower bits are the hash value of the rest of that IPv6 address. With the new node identifiers production mechanism, all the nodes in a domain will be mapped onto a contiguous key space on the overlay. That is, nearby nodes in physical networks shall also be neighbors on the overlay. As a result, it is high unlikely that messages will be routed many times into and out of the same domains, thus significantly reducing routing latency.

Theoretically in a system with N nodes dispersed in M different domains, the end-to-end routing latency shall be O (log (N)) [1] in Chord. But as in Chord6, all the nodes in the same domain don't disperse in other different domains. Instead they will be resident in a continual logical space. So if routing latency within the same domain can be ignored, we can take the domain as a single node, thus the end-to-end latency will be reduced from O (log (N)) to O (log (M)). As we will see later, simulation results presented in Figure 4 and Figure 5 confirm our prediction.

3. Simulation results and discussions

In this section, we evaluate the efficiency of Chord6 by means of our simulation. In our experiments, we use a two-level hierarchical topology generated by BRITE [7], which closely correlates with real network topologies and abides by power law as well.

3.1. Average Overlay Message Hops

We generate a topology with 4096 nodes dispersed in 100 domains. With the identifier space size set as 2^{32}.

We first try to evaluate the PDF (probability density function) of application-level hops in our Chord6 and

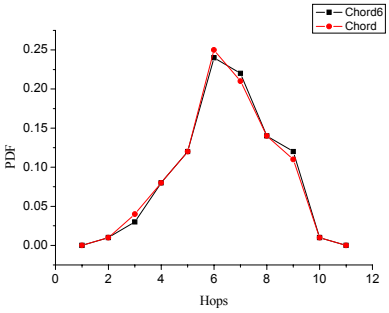

Fig.3. Distribution of application-level Hops

Chord. As shown in Figure 3, average hops per message in both systems are almost identical, which is reasonable. Since we barely modify the original Chord except the node identifier generation mechanism, there is hardly any difference between Chord6 and Chord from the perspective of the application level hops.

3.2. End-to-End Path Latency

Given that intra-domain latencies are quite small, it is reasonable for us to set the latency of one intra-domain hop as 10 milliseconds, and that of inter-domain hop as 100 milliseconds.

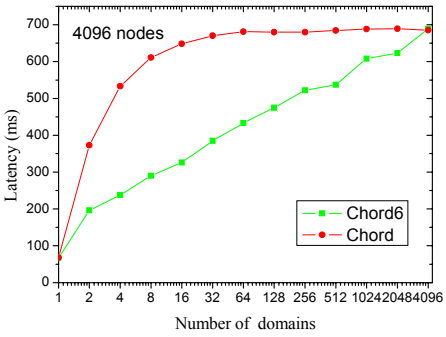

Fig.4. End-to-End Routing Latency with the number of nodes increased

We generate a topology with 4096 nodes and evaluate the true end-to-end path latency with the number of domains increased from 1 to 4096. Figure 3 shows that only in the worst case where no pare of nodes share the same IPv6 address prefix, the latency in both systems is the same. Though a lookup process in Chord only involves 7 application-level hops on average as shown in Figure 2, the true end-to-end latency can even reach 700 milliseconds in most cases. This means almost every application-level hop in the lookup path in Chord occurs between two different domains. In contrast, the more nodes share the same specific address prefix, the more extra inter-domain traffic can be saved in Chord6, which confirms our prediction. Consequently, the efficiency of Chord6 becomes much better than that of Chord.

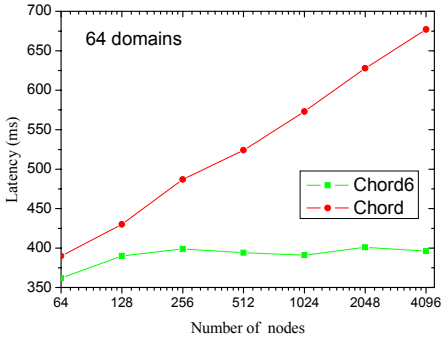

Fig.5. End-to-End Routing Latency with the number of nodes increased from 64 to 4096

In addition, we evaluate the end-to-end latency with the number of nodes increased from 1 to 4096, using a topology with 2 and 64 domains. As shown in Figure 4, the latency of Chord goes high as nodes increased, while the latency of Chord6 remains almost at the same low level. It is an important advantage of Chord6 that its performance keeps independent of the number of participating nodes if the number of domains rarely changes.

4. Conclusions and future work

To the best of our knowledge, we are one of the first group who ever attempts to couple IPv6 to P2P systems, so as to reduce the routing latency. The existing methods in context of IPv4 that try to address the routing latency problem will introduce more message exchange, and some of them even require the existence of super nodes, which incurs the undesired problem of single point failure. In this paper, we try to map physically nearby IPv6 hosts onto a contiguous space in the overlay using the hierarchical IPv6 address prefix, thus significantly reducing the end-to-end routing latency. Also, we give out Chord6 improved instances of Chord. We believe that to re-examine the P2P systems in the context of IPv6 will be an important research topic and deserve more work on it.

4. Acknowledgment

The authors would like to thank Jian Zhou, Lipeng Guo, Kaiping Xue, Zihua Xiu and other P2P group members for their helpful discussions and comments. Also w e would like to thank the anonymous reviewers for their useful comments.

6. References

[1] Stoica Ion, Morris R, Karger D, et al, “Chord: A Scalable Peer-to-peer Lookup Service for Internet Applications”, In ACM SIGCOMM’01, San Diego, CA, 2001, 31(4): pp.149-160.

[2] S. Rathasamy,P. Francis,M. Handley,R. Karp and S. Shenker, “A Scalable Content-Addressable Network”, in Proceedings of ACM SIGCOMM, San Diego, August 2001,pp.149-160.

[3] A. Rowston, and P. Druschel, "Pastry: Scalable, distributed object location and routing for large-scale peer-to-peer systems", in Proceedings of the 18th IFIP/ACM International Conference on Distributed System Platforms (Middleware 2001),Heidelberg, November 2001, pp. 329-350.

[4] Ben Y. Zhao, Ling Huang, Jeremy Stribling, Sean C. Rhea, Anthony D. Joseph, John D. Kubiatowicz, “Tapestry: A resilient global-scale overlay for service deployment”, IEEE Journal on Selected Areas in Communications, January 2004, Vol. 22, No. 1, pp. 41-53

[5] R. Hinden, S. Deering," Internet Protocol Version 6 (IPv6) Addressing Architecture", RFC3513, April 2003

[6] R. Hinden, S. Deering, E. Nordmark,. IPv6 Global Unicast Address Format. RFC3587, August 2003.

[7] Brite, a network topology generator. http://www.cs.bu.edu/brite/