LDHT: Locality-aware Distributed Hash Tables*

Weiyu Wu[#1], Yang Chen[#], Xinyi Zhang[*], Xiaohui Shi[#], Lin Cong[#], Beixing Deng[#], Xing Li[#]

#Department of Electronic Engineering, Tsinghua University, China

**Department of Electrical Engineering, University of California, Los Angeles, USA*

[1]wuwy02@mails.tsinghua.edu.cn

***Abstract* – As the substrate of structured peer-to-peer systems, Distributed Hash Table (DHT) plays a key role in P2P routing infrastructures. Traditional DHT does not consider the location of the nodes for the assignment of identifiers, which will result in high end-to-end latency on DHT-based overlay networks. In this paper, we propose a design of locality-aware DHT called LDHT, which exploits network locality on DHT-based systems. Instead of assigning uniform random node identifiers in traditional DHT, nodes in LDHT are assigned locality-aware identifiers according to their Autonomous System Numbers (ASNs). As a result, each node will have more nearby neighbors than faraway neighbors in the overlay. We evaluate the performance of LDHT on different kinds of typical DHT protocols and on various topologies. The results show that LDHT improves the traditional DHT protocols a lot in terms of end-to-end latency, without introducing additional overhead. It is indicated that LDHT is fit for different kinds of DHT protocols and can work effectively on all structured P2P systems including Chord, Symphony and Kademlia.**

I. INTRODUCTION

Distributed Hash Table (DHT) is the substrate of structured P2P systems. It supports the scalable storage and retrieval of {key, value} pairs on the overlay network. DHT-based systems are an important class of P2P routing infrastructures.

In DHT-based systems, nodes are assigned uniform random identifiers from a large identifier space. Data object (or value) is placed at the node with identifier corresponding to its unique key, which is chosen from the same identifier space. Lookup queries are forwarded across the overlay paths to nodes in a progressive manner, with the identifiers closer to the key in the identifier space.

DHT-based systems can guarantee that any data object can be located in small *O(logN)* overlay hops on average, where *N* is the number of nodes in the system. However, overlay hop count is not enough to evaluate the performance of DHT-based systems. Another efficient metric is the end-to-end latency of the overlay path. Routing algorithms that ignore the latency of individual hops will result in high-latency paths.

Without considering network locality in DHT, the underlying network path between two nodes can be significantly different from the path on the overlay network. Therefore, the lookup latency in the overlay network could be quite high and adversely affect the performance of the applications running over the DHT.

In this paper, we propose a design of an ASN-based locality-aware DHT called LDHT, which exploits network locality in DHT-based systems. We assign the node identifiers in a geographic layout manner to ensure nodes close in the network topology to be close in the identifier space. We use a node's ASN to generate the prefix of the identifier in order to make nodes in a same AS have close identifiers. As a result, nodes in LDHT-based systems will have more close neighbors than faraway neighbors in the network topology. The end-to-end latency for the query on the overlay network will thus be reduced. We use three typical DHT-based systems, Chord [1], Symphony [2] and Kademlia [3] as the basic DHT protocols to evaluate our design. According to the simulation results on different topologies, it is indicated that LDHT can improve the performance of DHT-based systems on both path length and Relative Delay Penalty (RDP) significantly, without adding overlay hops.

The rest of this paper is organized as follows. First we review related work in Section II. Then we present the design of LDHT in detail in Section III and evaluate its performance in Section IV. We conclude the whole paper in Section V.

II. RELATED WORK

Three basic approaches have been suggested for exploiting network locality into typical DHT protocols [4].

A. Proximity Routing

Proximity routing is the approach that the routing choice is based not only on which neighboring node makes the "most" progress towards the key, but also on which neighboring node is "closest" in the sense of latency. At each hop, a nearby node is chosen among the ones in the routing table. This approach strikes a balance between making progress towards the destination in the identifier space and choosing the closest routing table entry according to the network locality.

Proximity routing has been used in a version of Chord [1]. A set of alternate nodes are maintained for each finger table entry rather than one, and then queries are routed by selecting the closest node among the alternate nodes according to some network proximity metric.

B. Proximity Neighbor Selection

This is a variant of the above idea, but the proximity criterion is applied when choosing neighbors, not just when choosing the next hop. Routing table entries are chosen to

* This work is supported by National Basic Research Program of China (No.2007CB310806) and National Science Foundation of China (No.60473087).

refer to nodes nearby in the network topology, among all live nodes with appropriate identifiers.

Proximity neighbor selection has been used in the routing table construction of Tapestry [5] and Pastry [6]. They choose the closest node in the network topology according to some network proximity metric among the nodes whose identifiers have the appropriate prefix.

C. Geographic Layout

Geographic layout is the way exploiting network locality into node identifiers. In this approach, nodes' identifiers are assigned in a manner which ensures nodes close to each other in the network topology are also close in the identifier space.

In [7], the authors propose a hierarchical location-based node ID assignment to encode physical topology. A location-based node ID is a concatenation of a hierarchical prefix assigned to a node's region and a suffix of randomly generated bits. The scheme is based on geography, that is, different prefixes are assigned to different geographical regions.

Chord6 [8] is an IPv6-based modified version of Chord with the approach of geographic layout. It exploits the hierarchical feature of IPv6 address. In Chord6, a node's 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 of the rest of that IPv6 address.

III. Design of Locality-aware DHT

A. Basic Idea

The basic idea of LDHT is to exploit network locality on DHT-based systems in a geographic layout manner. Different DHT-based systems have different routing strategies and neighbor selection schemes, but they could have the same way of node identifier assignment. Once the routing strategy and neighbor selection scheme is determined, nodes choose neighbors only according to the identifiers of each other. Our purpose is to make LDHT compatible for all DHT-based systems, no matter what routing strategies and neighbor selection schemes they use. While proximity routing and proximity neighbor selection are approaches altered for different DHTs. So we choose the approach of geographic layout to exploit network locality.

In traditional DHT, no information about a node's network location or its proximity to other nodes can be deduced through its random identifier. Randomness in node identifiers will probably lead to high end-to-end routing latency. In LDHT, we construct a structured identifier space. Each node is assigned a locality-aware identifier, thus its network topology information can be embedded into its identifier.

When nodes are choosing neighbors in DHT-based systems, they will choose more nodes with identifiers close to themselves. So if we assign close identifiers to nodes close to each other in the network topology, they will have more close neighbors than faraway neighbors in the network topology. In LDHT, neighborhood relations of regions along the identifier ring reflect their proximity relations in the network topology. When DHT routing makes progress in the identifier space, similar progress is made in the network topology and thus overlay path costs are bounded.

B. Identifier Assignment

We use a node's ASN to represent its network locality for the reasons below. First, a node's ASN can be easily obtained by itself using WHOIS, which is a TCP-based query/response protocol widely used for querying a database in order to determine the owner of a domain name, an IP address, or an ASN in Internet. With abundant WHOIS databases available in Internet, this approach will not result in a single point of failure problem. In the worst case, if a node can not access any WHOIS database, it can generate a random number as its ASN, which will not effect the normal operation of the whole system. While if using the geographical information like the scheme in [7], we will need to either deploy and maintain a dedicated centralized database to partition the regions and assign prefixes, or have each end host maintain this kind of up-to-date database by itself. A centralized database will lead to single point of failure, and, maintaining the database by each end host is too costly. Second, when using ASNs, LDHT can work on both IPv4 and IPv6 networks. While depending on the hierarchical feature of IPv6 address, Chord6 [8] can only work on IPv6 networks.

We divide each node's identifier into two parts, *Global Part* and *Local Part*. Assuming that the length of the identifier is n bits, *Global Part* covers the highest m bits of the identifier, and *Local Part* covers the remaining n-m bits. *Local Part* is the prefix of the hash of the node's IP address, which is the same as most traditional DHTs. *Global Part* is generated according to the node's ASN. We assign a same *Global Part* to nodes in a same AS, in order to make them close to each other in the identifier space.

The length of the *Global Part* m is a tradeoff between end-to-end performance and load balancing of nodes, which can be adjusted according to the scale of the application system. In our simulation described in Section IV, we construct the node identifier with m=7. An ASN is an integer between 0 and 65536. We use a node's ASN modulo 2^m, which is converted to binary code, as its *Global Part*.

We concatenate *Global Part* and *Local Part* together, and form the whole locality-aware identifier.

C. Workflow of LDHT

Fig. 1 shows the workflow of LDHT. When a node joins LDHT, it first obtains its ASN and generates *Global Part* by the ASN in the length of m bits. Then, it will generate its *Local Part*, which is the prefix of the hash of its IP address in the length of n-m bits. Then, the node joins the two parts together to form a whole identifier. With this locality-aware identifier, it joins the DHT-based system and works the same way as in the original DHT protocol, such as neighbor selecting, message routing, etc.

With the API interfaces provided by our locality-aware DHT, distributed structured P2P applications will have better end-to-end performance. We will evaluate the performance of LDHT in next section.

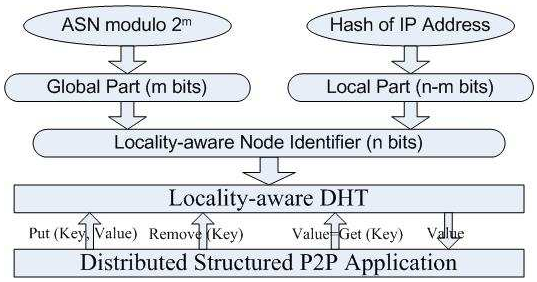

Fig. 1 Workflow of LDHT

IV. Performance Evaluation

We use Chord, Symphony and Kademlia as the basic DHT protocols, and add our approach on them to form LDHT-based systems. Performance of LDHT is evaluated and compared with that of the three original DHT protocols on two representative network topologies, one of which is generated by GT-ITM [9] and the other is collected from real-world Internet.

A. Simulation Setup

We use our own simulator to construct Chord, Symphony and Kademlia, and add our LDHT design to them respectively.

To accurately prove the effectiveness of our scheme, we implement two network topologies for the performance evaluation, Topology1 and Topology2.

Topology1 is generated by GT-ITM [9] with the scale of 4000 nodes. It's a two-level hierarchical topology. The top level of Topology1 consists of 200 ASes in 150 by 150 grids. The bottom level consists of a random number of nodes in the range of [13, 26] within each AS in 10 by 10 grids.

We use a real-world Internet distance dataset of 226 PlanetLab [10] nodes to construct Topology2 with the scale of 4520 nodes. The dataset contains latencies between nodes in PlanetLab with ping method in real Internet. These 226 PlanetLab nodes are distributed dispersedly in 80 different ASes. We use the location of the 226 PlanetLab nodes to generate a larger scale topology, which can still reflect the nodes' distribution of the real Internet. The 226 PlanetLab nodes serve as transit nodes, and 20 stub nodes are assigned to each transit node. We assign different distances to the edges in Topology2: the distance of intra-stub edges is 1; the distance of the edges between a transit node and a stub node is a random integer within [5, 15]; and the distance between transit nodes is from the distance dataset. Topology2 consists of all the stub nodes.

We use SHA-1 as the hash algorithm to generate the hash of IP address, with the length of 160 bits. For each system, we perform random queries for $4*10^4$ times to get the statistical and average simulation results. (In other words, we insert $4*10^4$ random keys into the overlay network.)

In the evaluation, we consider the following metrics:

- *Path length*: the latency in an end-to-end overlay path of each query. It is an efficient yet accurate metric to measure network structure and data delivery performance in different overlays.
- *Relative Delay Penalty (RDP)*: the ratio of end-to-end routing delay between a pair of nodes over that of a direct IP path per query. RDP represents the relative cost of routing on the overlay. The smaller it is, the better the path on the overlay network fits the path on the IP network.
- *Hop count*: the number of overlay hops in an end-to-end path of each query.

B. Simulation Results

We complete our simulations on Topology1 and Topology2 described in Section IV-A.

Fig. 2, Fig. 3 and Fig. 4 show the CDF of the path length of both original and LDHT-based Chord, Symphony and Kademlia. We also calculate the average path length of each protocol and topology and show the results in Table I. We use some short names due to the limited space. "TP1" and "TP2" means Topology1 and Topology2. "Orig" means original protocol and "LDHT" means LDHT-based protocol.

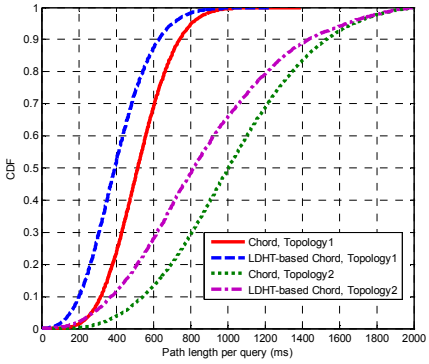

Fig. 2 Path length per query of Chord and LDHT-based Chord

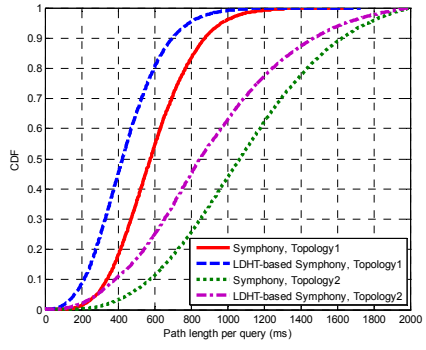

Fig. 3 Path length per query of Symphony and LDHT-based Symphony

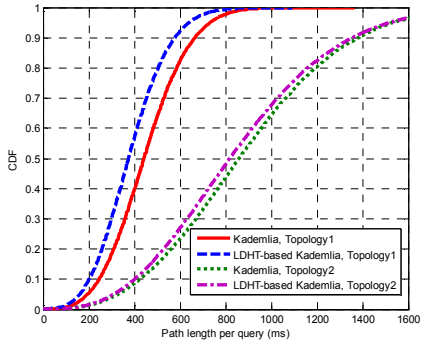

Fig. 4 Path length per query of Kademlia and LDHT-based Kademlia

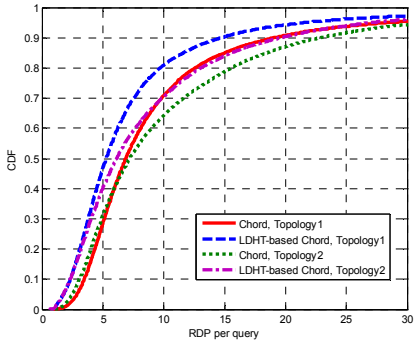

Fig. 5 RDP per query of Chord and LDHT-based Chord

TABLE I AVERAGE PATH LENGTH (MS)

	Chord		Symphony		Kademlia	
	Orig	LDHT	Orig	LDHT	Orig	LDHT
TP1	525	407	595	443	449	383
TP2	1024	869	1083	897	884	853

From the figures and the table, we can see that the LDHT-based systems have smaller path length than the original ones for all of the three DHT protocols on both topologies. It indicates that LDHT is much more efficient in terms of end-to-end latency. In fact, the query path on the LDHT overlay network has many intra-domain connections between neighbors, which are much shorter in terms of latency than inter-domain connections. As the original DHT overlay network doesn't take network locality into account, many neighbor connections are high-latency inter-domain links instead.

Fig. 5, Fig. 6 and Fig. 7 show the CDF of the Relative Delay Penalty (RDP) of both original and LDHT-based Chord, Symphony and Kademlia. Table II shows the average RDP of each protocol and topology. The meanings of the short names are the same as Table I.

We can see that on both topologies, RDP of the three LDHT-based systems are smaller than the three original ones. It indicates that the end-to-end path between two nodes on the LDHT overlay network is closer than that on the original DHT to the underlying IP network path. The relative routing cost of LDHT overlay network is smaller than the original overlay network.

Fig. 8, Fig. 9 and Fig. 10 show the comparison of hop count per query of both original and LDHT-based Chord, Symphony and Kadmlia on the two topologies. We present the results of the 10th, 50th and 90th percentiles of nodes. The results indicate that the hop count's distribution of our LDHT-based system is the same as the original DHT. The reason is that our design only changes the manner in which the identifiers are assigned, but doesn't change the original DHT's routing strategy and neighbor selection scheme at all.

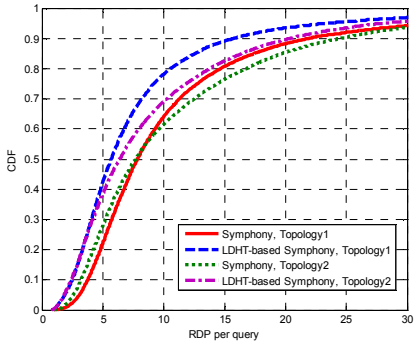

Fig. 6 RDP per query of Symphony and LDHT-based Symphony

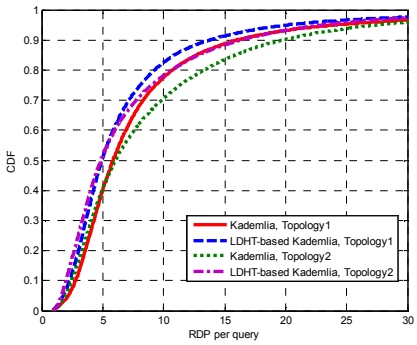

Fig. 7 RDP per query of Kademlia and LDHT-based Kademlia

TABLE II AVERAGE RDP

	Chord		Symphony		Kademlia	
	Orig	LDHT	Orig	LDHT	Orig	LDHT
TP1	10.71	8.22	12.19	8.64	9.11	7.50
TP2	14.24	13.16	15.48	12.80	13.54	10.82

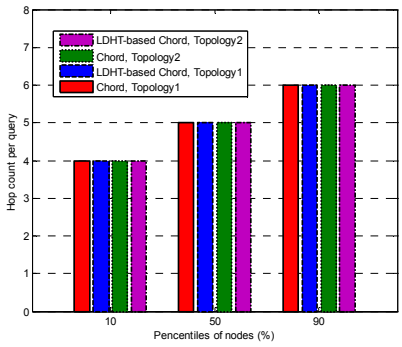

Fig. 8 Hop count per query of Chord and LDHT-based Chord

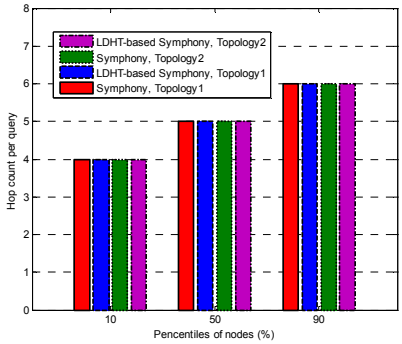

Fig. 9 Hop count per query of Symphony and LDHT-based Symphony

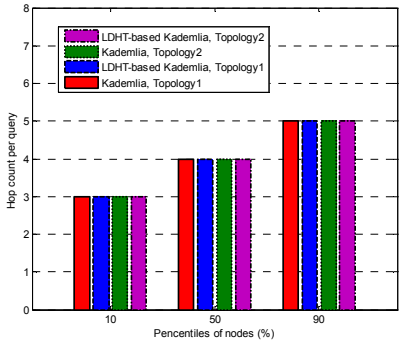

Fig. 10 Hop count per query of Kademlia and LDHT-based Kademlia

C. Simulation Conclusion

Results above clearly show that, LDHT is applicable for different DHT protocols and topologies. In comparison with original DHT, LDHT has better performance on end-to-end latency, without adding overlay hops.

V. Conclusion and Future Work

In this paper, we propose a design of an ASN-based locality-aware DHT called LDHT, which exploits network locality on DHT-based systems. We assign a node's identifier in a geographic layout manner that nodes with close identifiers in the identifier space are close in the network topology, so that they will have more close neighbors than faraway neighbors, and the end-to-end latency in a query can thus be reduced. We use a node's ASN to generate *Global Part* of the identifier that makes the nodes in a same AS have a same identifier prefix. As a result, there are more intra-domain neighbor connections in the path on LDHT-based overlay network.

We develop LDHT-based Chord, Symphony and Kademlia to evaluate the performance of our design in three metrics. Our simulations are done on both topologies generated by GT-ITM and real-world Internet. The simulation results prove the effectiveness of LDHT. The advantage of LDHT over traditional DHT lies in its better performance in terms of end-to-end latency like path length and RDP, without adding overlay hops. Meanwhile, LDHT is applicable for different kinds of basic DHT protocols and can work well on various topologies.

As for future work, first, we would like to deploy a publicly accessible DHT service, like OpenDHT [11]. People can easily issue put and get operations to any DHT node without running a LDHT client in order to use the LDHT service. Second, we will consider the proximity among ASes to improve the performance of LDHT. We have already done some works in [12] and hope to use this kind of scheme to make LDHT stronger.

References

[1] I. Stoica, R. Morris, D. Karger, M. F. Kaashoek, and H. Balakrishnan, "Chord: A scalable peer-to-peer lookup protocol for internet applications," IEEE/ACM Transactions on Networking, vol. 11, pp. 17–32, 2003.

[2] Gurmeet Singh Manku, Mayank Bawa and Prabhakar Raghavan, "Symphony: Distributed hashing in a small world," in Proc. UCITS'03, 2003.

[3] P. Maymounkov and D. Mazi`eres, "Kademlia: A peer-to-peer information system based on the xor metric," in Proc. IPTPS'02, 2002.

[4] Miguel Castro, Peter Druschel and Y. Charlie Hu, "Exploiting network proximity in Distributed Hash Tables," in Proc. IPTPS'02, 2002.

[5] B. Y. Zhao, L. Huang, J. Stribling, S. C. Rhea, A. D. Joseph, and J. D. Kubiatowicz, "Tapestry: A resilient global-scale overlay for service deployment," IEEE Journal on Selected Areas in Communications, vol. 22, pp. 41–53, January 2004.

[6] Antony Rowstron and Peter Druschel, "Pastry: Scalable, decentralized object location and routing for large-scale peer-to-peer systems," in Proc. IFIP/ACM International Conference on Distributed Systems Platforms (Middleware'01), 2001.

[7] Shuheng Zhou, Gregory R. Ganger and Peter Steenkiste, "Location-based node IDs: enabling explicit locality in DHTs," Carnegie Mellon University, Tech. Rep. CMU-CS-03-171, 2003.

[8] Jiping Xiong, Youwei Zhang, Peilin Hong and Jinsheng Li, "Chord6: IPv6 based topology-aware Chord," in Proc. ICNS'05, 2005.

[9] (2007) The GT-ITM homepage. [Online]. Available: http://www.cc.gatech.edu/projects/gtitm/.

[10] (2007) The PlanetLab homepage. [Online]. Available: http://www.planet-lab.org.

[11] Sean Rhea, Brighten Godfrey, Brad Karp, John Kubiatowicz, Sylvia Ratnasamy, Scott Shenker, Ion Stoica, and Harlan Yu, "OpenDHT: A public DHT service and its uses," in Proc. ACM SIGCOMM'05, August 2005.

[12] Lin CONG, Bo YANG, Yang CHEN, Guohan LU, Beixing DENG, Xing LI, Ye WANG, "NTS6: IPv6 based network topology service system of CERNET2," in Proc. MUE'07, Apr 2007.