Available online at www.sciencedirect.com
ScienceDirect
ELSEVIER
February 2009, 16(1): 111–115
www.sciencedirect.com/science/journal/10058885
The Journal of China Universities of Posts and Telecommunications
www.buptjournal.cn/xben

Topology-aware peer-to-peer overlay network for Ad-hoc

WANG Shi-guo[1,2] (✉), JI Hong[2], LI Ting[2], MEI Jing-qing[2]

1. Key Laboratory of Universal Wireless Communications, Ministry of Education, Beijing University of Posts and Telecommunications, Beijing 100876, China
2. School of Information Engineering, Xiangtan University, Xiangtan 411105, China

Abstract

The mismatch between the structured peer-to-peer (P2P) overlay network, which is based on Hashing, and the actual physical network, leads to query repeatedly passing through some nodes in the actual route when it is applied in Ad-hoc networks. An approach of getting an appropriate node identifier (ID) bearing its local physical information is proposed, in which the traditional theory of getting node ID through Hashing the node's Internet protocol (IP) address is abandoned, and a topology-aware overlay network suiting Ad-hoc networks is constructed. The simulation results show that the overlay network constructed in the proposed method can avoid the route being iteratively accessed. Meanwhile, it can effectively minimize the latency and improve the load balance.

Keywords P2P, Chord, topology-aware, overlay network

1 Introduction

With the fast development and wide application of computer and communication technology, more and more demands are emerging in the communication field, which exposes the shortcoming of the traditional client/server communication model, in which the single point failure will lead to the whole system breakdown. Fortunately, the P2P model emerges. P2P is a kind of distributed system, in which all nodes are at the same status. The nodes in the system share their resources, and manage the infrastructure together. The P2P system avoids the bottleneck problem, and it has the advantages of scalability and load balancing, all of which are not supported by the traditional network model. P2P provides a novel network model for future communication, and thus it has been the hot point of research for network application [1–2].

In P2P system, it is the fact that the structured P2P network is based on Hashing with scalability and reliability motivates researcher's interest. Some P2P structures with good performance are presented in Refs. [3–6], which provide self-organizing infrastructures for wide use. Among those structures, Chord is regarded as the best model with wide application prospect in the future [3]. However, there is a serious mismatching problem between the logical network and the physical network in the Chord used network, where the nodes' IDs are obtained by Hashing the nodes' IP addresses, and thus the IDs of the nodes connected directly are not correlated at all.

Ad-hoc network is a self-organizing network featured with scalability and failure-resistance. Since Ad-hoc network and P2P share several properties, if they cooperate effectively, improvement will be achieved in Ad-hoc, and their applications will become widespread.

In fact, if Chord is applied directly into Ad-hoc, it will induce accessing some nodes again and again just for a single resource query. This situation will be aggravated when Chord is used in Ad-hoc directly. As a result of the increasing burden, congestion may occur and the delay will be increased.

Considering the self-organizing and multi-hop characteristic in Ad-hoc, the traditional regulation of getting the node's ID through Hashing one's IP address is abandoned in this article. The node that wants to join the system can get an appropriate ID from its physical neighbor that has been in the system using the connection information. The other mechanism in the system is the same as the original Chord model. In this way, the maintenance overhead in the P2P system is unchanged, but it realizes the topology awareness between the overlay network and the physical network, and the proposal will minimize the overhead significantly when some resource is needed to be located. The simulation results show that the

Received date: 19-03-2008
Corresponding author: WANG Shi-guo, E-mail: sgwang7508@gmail.com
DOI: 10.1016/S1005-8885(08)60189-2

proposed P2P system can decrease latency and improve load capability.

The remaining sections of this article are organized as follows. In Sect. 2, related studies in P2P about topology awareness are introduced. The proposed topology-aware P2P system in Ad-hoc is described in Sect. 3. The simulation results and performance analysis are presented in Sect. 4. Finally, conclusions are given in Sect. 5.

2 Related work

Recently, topology-aware configurations of P2P network overlay topology have been extensively studied.

To improve the lookup efficiency in unstructured P2P system, some methods are presented. In Refs. [7–8], the researchers use the node's IP address and geographical information to form clusters in which super nodes assist other nodes in locating and routing between clusters when resource locating occurs. This method provides high performance for unstructured P2P system. Another method of constructing a topology-aware P2P model is presented in Ref. [9], in which the author utilizes latency to predefined landmarks to construct nodes and form clusters. The process is coarse and it requires the landmark nodes to keep steady.

In Ref. [10], the author tries to enhance locating efficiency by adding a secondary lookup overlay, which is composed of high-capacity nodes, and the closely located peers are grouped into the same autonomous system. It can reduce the underlying physical routing delay, but it needs some super nodes to manage the system. The method of getting an appropriate node ID by considering the nodes' physical position is first proposed in Ref. [11]. In the initialization period, some landmark keys are given, and any node that wants to join the system tries to find the closest one, and get a node ID from the given range. The landmark node and the nodes joining make up a cluster, and the landmark node administers all the other nodes in the cluster. Since the IDs in the same cluster are very close to each other, the construction realizes topology awareness to some extent. The method proposed in this article also benefits from applying the technique through which the joining node can get its appropriate ID value.

In Refs. [12–13], the researchers try to modify the original Chord structure through increasing the finger table that stores one's proximity information. In this way, improvement to solve the mismatch phenomena has been achieved, but it is far from achieving topology-aware motivation and it increases the maintenance overhead. Also, at the same time, it cannot guarantee its reliability and feasibility in arbitrary condition.

There are new methods proposed based on special future protocols, such as in Refs. [14–15]. For example, in Ref. [15], the researchers use IPv6 to realize their construction. If the proposed protocols have good performance and are easy to realize, higher efficiency can be achieved. However, they do not provide authentic methods, which can be used in P2P application currently, and they do not present specific law or systemic ways on how to construct a P2P network.

Although the previously suggested protocols can realize topology awareness to some extent, none of them are designed for the P2P system combined with Ad-hoc. Thus, we propose a new method to construct the topology-aware P2P model according to the property of multi-hop in Ad-hoc networks.

3 Topology-aware P2P model in Ad-hoc

To solve the mismatch problem in Chord P2P system, the node must get necessary information of its physical neighbors, and the way of getting the node's ID must be independent of its IP and should be related to its physical neighbors. Thus, a new method of getting the appropriate node ID, which relates to its surrounding nodes, is proposed, through which the overlay network and the physical network can be consistent.

3.1 Initialization of the P2P system

At the initial stage, according to the number of nodes and the connectivity between them, the whole ID space is divided uniformly. We take the scenario of four nodes in the initial stage as an example, and the connectivity of the nodes is shown in Fig. 1.

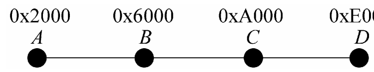

Fig. 1 The connection figure with four nodes

In this scenario, the nodes' IDs are 0x2000, 0x6000, 0xA000, 0xE000 respectively (to simplify the description, we describe the ID space in 16 binaries, which is actually 128 binaries).

3.2 Joining process of a new coming node

After discovering the initial network, when a node wants to join the P2P system, the following processes will be implemented:

1) The joining node broadcasts its joining request to all its physical neighbors (the value of time to live (TTL) is set to be 1), and waits for responses from its neighbors that have already been in the P2P system. The situation is shown in Fig. 2 (Node *F* sends its request to its physical neighbors *A* and *B*).

2) During the request processing, the node records all the information of its neighbors who reply to the request, and

arranges them in the order as the reply reaches (in Fig. 2, if the distance of F and B is shorter than that of F and A, node B replies F ahead of A usually).

3) The joining node integrates all its received information, and sends request for ID allocation to its neighbor node that replies to it at the first time in stage 1), and waits for the response containing the allocated node ID (the packet includes all the IDs of replying nodes).

4) The joining node gets the allocated node ID from the response message, and joins the system following the original regulation in Chord.

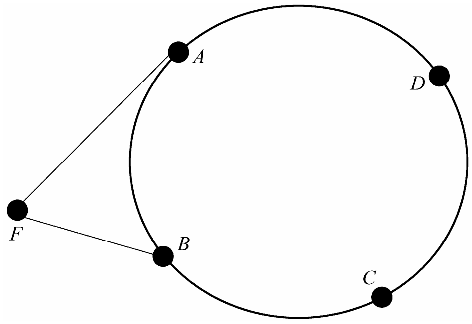

Fig. 2 The sketch map when a node wants to join the system

3.3 Messages processing for node in the P2P system

In this protocol, all packets are divided into five categories, which are named as resource download packet, resource upload packet, records updating packet, joining request packet, and ID applying packet. When a node that has been in the system receives a packet, it will take the corresponding steps to handle the packet. The process is shown in pseudocode as follows:

```
node n receives a packet p
if p is not an ID asking packet or a querying for joining system
enter the corresponding process
    end if
    if p is a query for joining system packet
    reply the query with its ID
    end if
    if p is a packet of ID applying
      if p includes no other nodes message or no node is node n's
      neighbour n allocates a proper ID and replies
      end if
      if p includes some n's neighbour
      selects the node with minimal delay, according to whose
      ID and n's ID, a proper ID is allocated, and reply
    end if
end if
```

3.4 Node ID allocating regulation in the system

How to allocate an appropriate ID is very important for maintaining load balance in this protocol. To describe the regulation clearly, we show the process with Fig. 2.

1) When a node in the system receives an ID allocating request, it will check the packet at first. If the packet is just an ID applying without information of the other node, the node will select itself as the predecessor node and allocate its successor node as the successor node for the applying node, or select itself as the successor node and its predecessor node as the predecessor node as well, which is determined by the ID space to its predecessor node and successor node. The condition is that node F connects with node B (or A) directly only, and therefore, it selects node B (or A) and A (or D) as the predecessor node and the successor node respectively, or selects node C (or B) and B (or A) as the predecessor node and the successor node respectively. Since the two ID spaces are the same with 0x4000 (from C to B and B to A), the two selection results are equal and are taken randomly. To optimize load balance, the allocated ID should take the central value between the predecessor and the successor. For example, in Fig. 2, if node F connects with B only, an ID value of 0x8000 or 0x4000 will be allocated for node F.

2) If the packet includes some information of other nodes, but none of which is the receiver's logical neighbor, the law for allocating ID is the same as the above condition.

3) If the packet includes some information of other nodes, and some of them are the receiver's logical neighbors, in this situation, the receiver sets itself as the successor node and selects the nearest node as the predecessor node. For example, if B receives a packet, which includes the information of node A, then A and B will be selected as the predecessor node and the successor node respectively.

When there is no free ID for the joining node (the situation occurs rarely, because the ID space is large enough, and it is distributed uniformly), the corresponding predecessor nodes and successor nodes will ask for their logical neighbors to shift their node IDs, until the ID space is large enough for the allocation.

4 Simulation and performance analysis

4.1 The simulation of average stretch and delay

The simulation is carried out on NS platform. To compare the amounts between logical hops and actual physical hops, dropping packet should be avoided, and therefore, TTL must be large enough (TTL is set as 10 000 in the simulation). At the same time, congestion should not happen in the simulation system. Consequently, a request packet is generated every four seconds. During the simulation process, one node is selected out randomly, and the node sends a resource location request every four seconds one after another, until all the resources are accessed (it is assumed that each node possesses only one resource whose ID is the same as its node ID).The comparison of the result with that of the original Chord is shown in Figs. 3 and 4. In Fig. 3, Chord-delay denotes the average delay of the original Chord and the proposed-delay denotes the average delay of our proposed method. In Fig. 4, Chord-stretch denotes the average stretch of the original Chord and proposed-stretch denotes the average stretch of our proposed method.

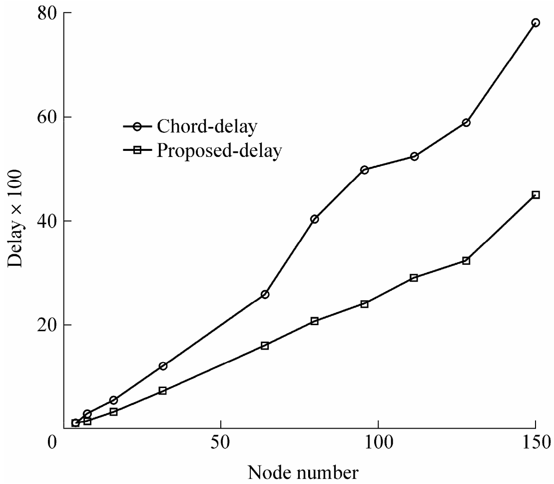

Fig. 3 Comparison of average delay

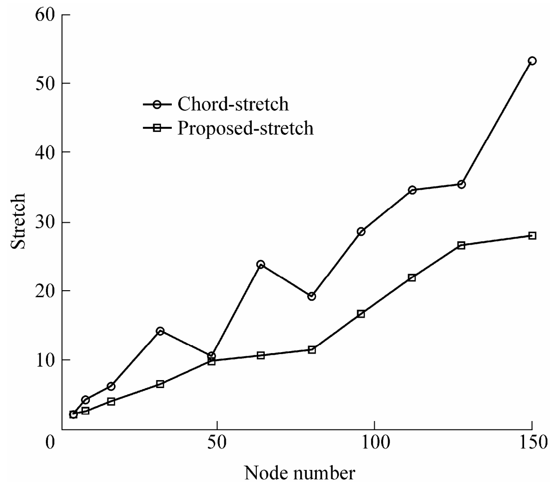

Fig. 4 Comparison of average stretch

From Figs. 3 and 4, it is clear that the average delay and average stretch are considerably smaller than that of the original Chord when our method is used in Ad-hoc P2P system. Specifically, the application can get the response more quickly in our proposed system.

4.2 Simulation of system burden

Aiming at comparing the burden of the system when the same task is to be implemented, the drop-packet ratio must be calculated. Therefore, the value of TTL is set to be 20 or 200 when the simulation is carried out respectively. The result is shown in Fig. 5, in which proposed-200, Chord-200, proposed-20, and Chord-20 denote the drop-packet ratios of our proposed system and the original Chord when the TTL is set to 200 and 20 respectively.

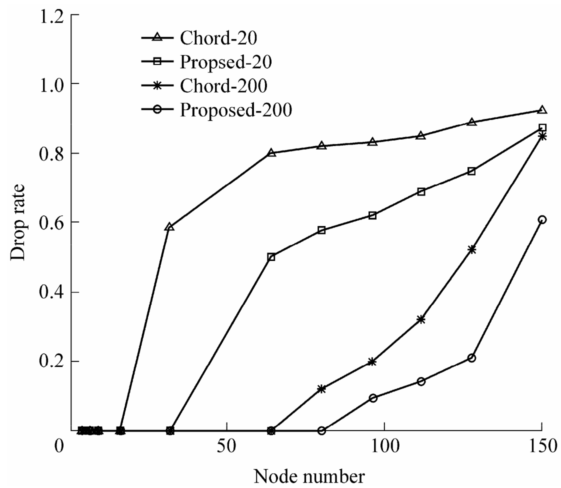

Fig. 5 Comparison of drop-packet rate

From the statistical result in Fig. 5, we can see that the packet drop ratio is considerably lower than that of the original Chord when they are used in Ad-hoc network, respectively. In fact, there are still some packets having been dropped in our proposed P2P system, because the value of TTL is limited, and the path between the query node and the destination is probably out of the TTL bound.

5 Conclusions

In this article, we studied how to build a topology-aware P2P system in Ad-hoc network. The physical information of a node's neighbors is considered carefully. In the proposed scheme, the node joining the network can get an appropriate ID, which is very close to the ID of its physical neighbors. In this way, a P2P system is constructed in self-organizing networks (Ad-hoc). The mismatch between overlay network and physical network is avoided effectively. The simulation results show that when our proposed method is used in Ad-hoc, improvement can be achieved as follows:

1) The network avoids access to a single node repeatedly just for a single resource location.

2) The reaction to application service will become faster.

3) The capacity of the system is improved greatly.

In this article, we solve the problem of mismatch when the P2P system is constructed over the Ad-hoc network. Our protocol is applicable to Ad-hoc in which the nodes are static.

However, there are several application scenarios where the nodes move randomly. A possible approach to keep matching between overlay network and physical network is switching nodes ID when the topology is changed, which shall be further studied in the future.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (0672124), the Hi-Tech Research and Development Program of China (2007AA01Z221).

References

1. Ahmed R, Boutaba R. Distributed pattern matching: a key to flexible and efficient P2P search. IEEE Journal on Selected Areas in Communications, 2007, 25(1): 73−83
2. Zhang R M, Hu C. Assisted peer-to-peer search with partial indexing. IEEE Transactions on Parallel and Distributed Systems, 2007, 18(8): 1146−1158
3. Stoica I, Morris R, David L, et al. Chord: a scalable peer-to-peer lookup protocol for internet applications. IEEE/ACM Transactions on Networking, 2003, 11(1): 17−32
4. Ratnasamy S, Francis P, Handley M, et al. A scalable content-addressable network. Proceedings of Conference on Applications, Technologies, Architectures and Protocols for Computer Communication (SIGCOMM'01), Aug 27−31, 2001, San Diego, CA, USA. New York, NY, USA: ACM, 2001, 161−172
5. Ben Y. Huang Z L. Stribling J, et al. Tapestry: a resilient global-scale overlay for service deployment. IEEE Journal on Selected Areas in Communications, 2004, 22 (1): 42−53
6. Rowstron A, Druschel P. Pastry: scalable, decentralized object location and routing for large-scale peer-to-peer systems. Proceedings of the 18th IFIP/ACM International Conference on Distributed Systems Platforms, Springer-Verlag, 2001: 329−350
7. Le H, Hong D, Simmonds A. A self-organizing model for topology-aware overlay formation. Proceedings of 2005 IEEE International Communications Conference, May 16−20, 2005, Seoul, Korea: Piscataway, NJ, USA: IEEE, 2005: 1566−1571
8. Zhao J, Lu J D. Solving overlay mismatching of unstructured P2P networks using physical locality information. Proceedings of the 6th IEEE International Conference on Peer-to-Peer Computing, Sep 6−9, 2006, Cambridge, UK: Piscataway, NJ, USA: IEEE, 2006, 75−76
9. Xiang Z, Zhang Q. Peer-to-peer based multimedia distribution service. IEEE Transactions on Multimedia. 2004, 6(2): 343−355
10. Kwon G; Dong K R. Bypass: topology-aware lookup overlay for DHT-based P2P file locating services. Proceedings of the 10th International Conference on Parallel and Distributed Systems, Jul 7−9, 2004, Santa Fe, NM: Piscataway, NJ, USA: IEEE, 2004: 297−304
11. Yu Y S, Miao Y B, Shieh C K. Improving the lookup performance of Chord network by hashing landmark clusters. Proceedings of 14th IEEE International Conference on Networks, Sep 13-15, Singapore. Piscataway, NJ, USA: IEEE, 2006: 1−4
12. Hong F, Li M, Yu J D. PChord: improvement on Chord to achieve better routing efficiency by exploiting proximity. Proceedings of the 25th IEEE International Conference on Distributed Computing Systems Workshops, Jun 6−10, 2006, Columbus, OH, USA. Piscataway, NJ, USA: IEEE, 2006: 806−811
13. Liu Y, Yang P, Chu Z, et al. TCS-Chord: an improved routing algorithm to Chord based on the topology-aware clustering in self-organizing mode. Proceedings of the First International Conference on Semantics, Knowledge, and Grid, Nov 28−29, 2005, Beijing, China. Piscataway, NJ, USA: IEEE, 2005: 25−25
14. Cui J Q, He Y X, Wu L B. More efficient mechanism of topology-aware overlay construction in application-layer multicast. Proceedings of International Conference on Networking, Architecture, and Storage, Jul 29−31, 2007, Guilin, China. Piscataway, NJ, USA: IEEE, 2007: 31−36
15. Dao L H, Kim J W. AChord: topology-aware Chord in anycast-enabled networks. Proceedings of 2006 International Conference on Hybrid Information Technology, Nov 9−11, 2006, Cheju Island, Korea. Piscataway, NJ, USA: IEEE, 2006: 334−341

(Editor: WANG Xu-ying)