An Improved Adaptive Routing Algorithm Based on Link Analysis

Jian Wang  
Sichuan University / College of Computer Science, Chengdu, china  
Email: wj_98@163.com

Xingshu Chen and Dengqi Yang  
Sichuan University / College of Computer Science, Chengdu, china  
Email: { chenxsh@scu.edu.cn, dengqiyang@163.com}

Abstract—DHT (Distributed Hash Tables) has been applied to the structured P2P system to achieve information retrieval and positioning efficiently. KAD is a large-scale network protocol based on the XOR metric in practice, which uses DHT technology to improve network salability without central server. However, the increasing malicious pollutes routing tables to reduce seriously the query performance. Thus, an improved adaptive algorithm based on social network is proposed in order to improve routing table updating algorithm. Firstly, the data structure of routing table is adjusted to store value of centrality and prestige. Secondly, the request nodes can adaptive select nodes to send messages. Then when a find process is terminated, the node will calculate the two values for all participating nodes using the corresponding centrality and prestige algorithms based on XOR metric. Finally, the node updates the routing table depend on the above result. The above algorithm was implemented in an open source project named LibTorrent to test effectiveness. This experiment last a month to verify the change of the search success ratio in a KAD network with about 30% malicious nodes. The results show that the optimized adaptive routing algorithm can effectively resist the attack for routing table and improve the search success ratio of the node. Moreover, this lightweight algorithm is conducive to the deployment in practice without extra network burden.

Index Terms—DHT, KAD, routing algorithm, Link Analysis

I. INTRODUCTION

P2P system can be subdivided into centralized P2P system, unstructured P2P and structured P2P system according to the overlay networks of organizing [1]. The first generation centralized P2P system disappeared from view because of inherent weaknesses. As a result of DHT technology, structured P2P system such as CAN, Kademlia, Tapestry and Chord, makes information retrieval and location have been greatly improved efficiency compared with unstructured P2P system. Hence, structured P2P is a main direction development of P2P system. KAD is a kind of DHT technology based on XOR metric, which has been widely adopted in practice file shared software, such as eMule/aMule[2]. BitTorrent DHT, Azureus[3], MainLine and so on.

KADEMLIA is widely used routing protocol based on XOR metric, which has some obvious advantages compared with others [4]. The distance between two nodes in the network is obtained using the XOR calculation. Before a node joins the DHT network, it requires a random 160 bit (or 128 bit) number as their identity using hash algorithm (MD5 or SHA1). When a node wants to share a file, it also uses the same hash algorithm for the file with any length to generate a 160 bit digest a value as the key word. A pair including key word with the publisher information (IP and Port) will be released to store in k nodes whose id values closest to the key word. To speed up the retrieval efficiency, each node in the network will create and maintain a routing table that stores a part of the information for all nodes. The routing table for KAD is organized by some k buckets [5]. When a node starts a request, it will pick up α nodes whose ID is closest to the object file hash value from the routing table to send the request parallel. Responders pick up β nodes whose values are closest to key value in their routing tables and back to the requester with messages including β nodes. Then, the requester selected some closest targets from α+β nodes depending on metric distance. The search process will be repeated iteratively until the target is found or failure. This failure means not more closer nodes can be found. This prefix matching mechanism ensure each search can at least halve the XOR distance, and the maximum number of search times not exceed log₂N, N denotes the number of nodes in the network. This algorithm made a better trade-off in performance and overhead.

II. RELATED WORK

Some previous papers focus on how to improve the lookup performance [6, 7]. Sergio Marti et al propose SPROUT algorithm to route using social links, which can increase the probability of successful routing [8]. Liu et al present a approach to build a interest-based communities, which can short the deliver path and improve the routing efficiency [9]. George Danezis and Prateek Mittal present the Sybilinfer algorithm for labeling the nodes in social...
network to identify the Sybil[10]. Yu et al analysis reliability among nodes depend on social link to determine which are Sybil in the network. They suggest using the SybilGuard algorithm to limit the influence of the Sybil attacks. Liu et al begin to study the nodes in the routing table based on KAD network, they present a social link based on IP distance Metric to improve routing performance in KAD network [11]. But, this algorithm is not suitable for the KAD network based on XOR metric. Moreover, a node with internal IP address can’t be properly ranked.

III. ROUTING ALGORITHM DESCRIPTION

The main difference between DHT and traditional network is that each node in DHT was been seen as a client and server. Each node not only can download/upload files, but also provide route services. Therefore, the performance of routing table will play a vital role for node search. The routing table in each node has provided the redundancy to response to frequent churn that nodes free join or leave the network. In order to maintain the availability, each node needs to interact with other nodes and update the corresponding entry in the routing table. Each routing table in a node saves part of the nodes’ information whose distance from himself between 2i and 2i+1, say d= [2i, 2i+1). The i denotes the changes range from 0 to 160. say i= [0,160]. An entry in routing table called a K-bucket can store no more than k nodes information. According to the found time, those nodes are arranged. The node with first seen was placed on the head, and latest seen was placed on the tail in each bucket.

A. State model for a node

Each node will create and update routing table, along with node joins network, sends request, provides service and leaves network. It is necessary to model a node in order to understand behavior. It shows the state transition for a node in figure 1.

The main operations for interaction among nodes are following:

- Starting state: It means a node issues a request to require a node in the DHT network in order to join the network. The responder can provide 20 nodes information in back message, and the requester inserts 20 nodes into the routing table. Then the node continues to fill the routing table after it sends some messages to find its own to the 20 nodes and receive the back messages. Meanwhile, the 20 nodes update the routing table with requester information. How to ensure the reliability of the fixed nodes is the premise of the node can successfully join the network. Practically, some fixed nodes instead of selected nodes randomly always are used as guide nodes to prevent nodes from isolation attack. Although this method is simple and effective, it increases the dependence of these fixed nodes and borrows the bottleneck and single point of failure problem. Therefore, the really matter is how to reduce the dependence on the fixed nodes to improve the join efficiency of nodes initialization process.
- Idle state: The term indicates that the node is in the absence of requesting or providing services to other nodes. The node in this state is only to randomly select some nodes from the routing table to ping. If no response messages, the target nodes will be removed by requester from the routing table, which ensures that most nodes are online. Unfortunately, taking to account the efficiency, the node is only to select part of nodes to detect. In other words, not all nodes in the routing table at a time online.
- Running state: Nodes request or provide service for other nodes in network. The nodes issue request or response for the request message that want to find a value or node.
- Download/upload state: Access to the information for the target nodes who own the source files, request nodes will start to download and upload process. When nodes have finished the download, they will announce to K nodes whose id is closest to the file’s hash value with store operation. The pair with node IP and Port will be stored in the k different locations so that others can find the value with high probability.

B. Four Operations for a node

Node in the idle state, running state will call or receive RPC (Remote Process Call) operation to achieve the maintenance of its own routing table. Four operations including in RPC are Ping, Find Node, Find Value and Store. Their functions as follow [13]:

- Ping: The function is to detect whether a specified node is still online.
- Find Node: The function is to ask a node return k nodes closest to a required ID from the routing table.
• Find Value: The operation is similar to Find Node. The function is to ask nodes whether own the key value as same as require value.
• Store: The function is to inform K nodes to store <key, Value> information as to others search. The node who has received the one of above process calls may update the routing table. But the updating operation will be successfully depends on the following figure.

![Figure 2](image)

The process can ensure the node fill corresponding bucket in routing table as soon as possible. When the number of a bucket reach limit and new request come, the node will ping another node that is first to be inserted into the bucket early.

IV. THE IMPROVEMENT FOR ROUTING TABLE MAINTENANCE MECHANISM

A. Analysis of the current problems

From the above analysis, the routing table maintenance mechanism is relatively efficient. It fully takes into account the dynamic and overhead for interactive with nodes. However, there are two problems for the mechanism as following:
• Has received the RPC command, the node only to test the node that in a bucket head. Its means whether the node will update the routing table depends on the head node rather than others in a bucket. Besides, the node can randomly select nodes in some buckets that haven’t updated any node to ping on a regular basis. Obviously, it is not feasible that ping all nodes when the node need to update. Considered an extreme case, a new node can’t be inserted into a bucket even if the other k-1 nodes are not online. This is not unsafe for a node if some malicious nodes launch the routing attacks. Because, these malicious nodes with multiple IP can stick to ping a victim node so that make themselves added into the routing table. At the same time, when a malicious node has been moved to the head in a bucket, the attacker can prevent new nodes from inserting into the bucket.
• A requester sends the request for finding value or find node, and responder randomly select α nodes from routing table. A requester will update the corresponding node information without checking. This means that a responder can more easily make requester use some victim nodes update the routing table. The search process will be repeated until one (or more) node is found or no closer nodes are returned. It is easy to known that a node find a target node after 1/2log2N (N is the number of all nodes) on average. Thus, a node may has received 1/2log2N*α malicious nodes information for a search process.

The maintenance of routing table should be improved to keep more reliable nodes in the routing table. Thus, it is the point that how to assess the reliability of node. It is more likely for a reliably node to provide search service in routing process. As a result, the improvement strategy of routing table maintenance consists of two parts in this paper: node evaluation algorithm and the node adaptive selection algorithm.

B. routing improvement algorithm based on link analysis

DHT network and social relationship network is very similar. It is helpful to assess the node in the DHT with link analysis in the social network. If nodes are viewed as nodes, the activity or relationship between nodes can be represent as links, which the DHT network model will be become a direction grapy. The characteristics of nodes, such as behavior, authority, position and so on, will be easy to study based on the grapy. Actually, both centrality and prestige are measures of prominence of a node in the network [12]. The higher of degree of a node, it’s more reliable for the other nodes in the network. To explain the term of node reliability, the definitions of centrality and prestige are introduced in this paper.

Node prestige: A prestige node is object of extensive links as a responder. Generally, a requester sends message to the node with higher prestige value, which gets the correctly back nodes with higher probability.

Node centrality: A centrality node is one who has more links, which is more important compare to other with relatively fewer links. Generally, a requester sends message to the node with higher centrality value, which gets the back nodes with shorter the distance for destination node.

From above definition, the focus of routing table updating is converted to calculate the value of centrality and prestige. To achieve this goal, the following measures will be implemented:
• Changes the routing table structure
To compute and store the value of centrality and prestige, the value of centrality and prestige of a node will be added respectively in each k bucket. A five-tuple of <ID, IP, Port, Auth, Cent> can be used to represent the node.
• Changes the RPC operations
There is no difference for a request node sending Ping or store command. When the node sends the find value or finds node command, it is essential for the requester to evaluate the centrality and prestige of those nodes that are extracted from a return message. A requester may update the routing table depend on the result of evaluation. The details of improved process are as follows: Node A selected some nodes to send the request. Each responder provides $\beta$ nodes for requester. As a result, a responder has an in-link and $\beta$ out-links. If the number of return nodes can not meet the requirements, the requester will start retransmission mechanism.

- How to select the nodes for a requester and responder depending on the adaptive algorithm.
- The requester does not update some nodes immediately and pick up some closer nodes by comparing the return nodes to repeat the previous step.
- When the target node or value is satisfied, the search process is termination.
- The value of centrality and prestige for each return node is calculated depend on the number of in-link and out-link.
- According to the value of centrality and prestige, the requester starts to update corresponding buckets.

The node adaptive selection must be referred to above process in order to explain more clearly. That is how to select some nodes in the beginning and how to update some nodes at the end.

Firstly, a requester picks up some evaluated nodes that are composed of nodes with higher prestige and nodes with higher centrality value for first requesting. Then, an equation to describe the relationship, say $Pr = dp / dc$, $\alpha = dp + dc$.

The $Pr$ is the proportion of prestige and centrality value. The higher the value of $Pr$ means the node set owns more security. On the contrary, the set node owns more efficient.

For the first requesting, the requester selects $\alpha-1$ central nodes and a prestige node. If it failed, the node will automatically pick up $\alpha-2$ central nodes and 2 prestige nodes, and so on. To obtain a higher value of prestige and centrality, a responder was encouraged providing some better nodes. Actually, it is no obviously different for requester and responder to select the prestige and centrality nodes from the routing table. The adjustment process is automatic, so it is called adaptive. But, it is allowed for user to adjust parameters in order to get a reasonable node set with different security and efficiency in practices.

Secondly, after the requesting is finished, the requester begin to update some nodes depend on the two values. The $Pu$ is the ratio of the amount of being updated prestige nodes ($dp'$) to centrality nodes ($dc'$) in a bucket, say $Pu = dp' / dc'$.

C. Algorithm Implementation

The improved algorithm for ranking all nodes in a find process is based on the prestige and centrality algorithm [12]. The requester needs to calculate the values at the end of find instead of each step. After entered the KAD network, the node will initialize threshold values of prestige and centrality for each node to 1 separately. In this paper, the algorithm of degree centrality is adjusted to calculate the centrality value. The value of a node is defined as $Cc(i)$

$$ Cc(i) = \frac{dx(i,j) \cdot \| I_i \|}{(n - 1)} = \frac{dx(i,j)}{(n - 1) \cdot dx(s,t)} \quad (1) $$

$$ dx(i,j) = dx(i,t) - dx(t,j) \quad \text{if} \quad dx(i,j) > dx(i,t) \quad . \quad (2) $$

Where $n$ is the number of nodes in a find process. Let $t$ is the target node and $s$ is the requester, the $dx(i,j)$ denotes XOR distance from i to t. The direct out-link need to be calculated for i, which is the centrality of those nodes who have been provided by i can be got. The value of the formula ranges from 0 to 1 because of the normalization with $(n - 1) \cdot dx(s,t)$. The $Cc(i)$ depends on two factors of the XOR distance and the number of nodes. According to XOR metric characteristic, $dx(i,j) > dx(s,t)$ indicates that j has exceeded the target node. So, $dx(i,j)$ must be adjusted with $dx(i,t) - dx(t,j)$ to reflect the extent of j close to t. From the DHT characterizes, it is easy to understand that a node more likely to know the nearer node. Hence, a node can be ranked with higher value because it provides a successor closer to target.

Similarly, the proximity prestige value is calculated as $Pp(i)$

$$ Pp(i) = \frac{| II_i | / (n - 1)}{\sum_{j \in I_i} dx(j,i) / \| II_i \|} \quad (3) $$

$$ dx(j,i) = dx(j,t) - dx(t,i) \quad \text{if} \quad dx(j,i) > dx(j,t) \quad . \quad (4) $$

Where $| II_i |$ is the size of set that all nodes who can link directly or indirectly to i. The value of the above measure ranges form 0 to 1. The $Pp(i)$ depends on two factors of the XOR distance and the number of nodes related to i. Note the distance is inverse proportional to the prestige value. Reviewing the characters of k buckets, a node can know more with their neighboring nodes. For example, the two values can be calculated in the figure 3.

![Figure 3. The simplified routing process from A to T](image-url)

The 4 bits number in brackets instead of 16 is the node ID in order to simplify the routing process. After node A finished the find operation, those related nodes including A, B, C, F, G and T to node A can be selected out to
calculate, say n=6 and \( d_A(i, j) = 14 \). Two values for all nodes are given in the following table.

| TABLE I. \( n=6 \) TWO VALUES FOR ALL RELATED NODES |
|----------|---------|---------|---------|---------|---------|---------|
|          | A       | B       | C       | F       | G       | T       |
| \( P_A(i) \) | 0.129   | 0.157   | 0.029   | 0.014   | 0       |
| \( C_A(i) \) | 0.004   | 0.005   | 0.038   | 0.033   | 0.132   |

Note that the value of \( d_A(G, A, G) \) has been adjusted to 13 because of \( d_A(G, A, G) > d_A(T, A) \). As can be seen from the table, the value of \( C_B(B) \) is smaller than \( C_C(C) \), which indicates that node linked by B is closer to T. The value of \( P_B(B) \) is bigger than \( P_C(C) \), which indicates that C is more reliable than B. Through the above algorithms, the requester can get the value of prestige and centrality for the all participating nodes. As a result, the requester will decide which of nodes will be replaced.

V. EXPERIMENT AND RESULT

To test the validity of this way, an open source project named LibTorrent was selected for this work. LibTorrent is a feature complete BitTorrent implementation focusing on efficiency and scalability [13, 14]. There are two experiments are used to test the search success ratio and result for resisting the routing attack. Two different versions of the client program are created, in which a client program is created using original routing algorithm and another client program is created using above improvement routing algorithm. A total of 1,000 normal nodes involved in this testing in the DHT network. Half of the total number of the nodes uses the former client program, while the rest of the nodes use the second program where some parameters are set, for example \( \alpha = 3 \) in method improved-1, \( \beta = 1 \) and \( \delta = 2 \) in method of improved-2. There are about 300 nodes who insist in poisoning other node’s routing tables enter the network. Test process lasted for one month. Relevant data are collected after each nodes issue about 300 find value commands in the DHT network. It shows the difference for the search success rate in figure 4.

From the above result can be seen that the improvement algorithm obviously improved the search success ratio and resisted the routing table attacking by preventing those malicious nodes from inserting. If a malicious node can not contribute for other’s request, the node will be removed from routing table because of lower the prestige or centrality value. Moreover, the two values not change even if the malicious initiative to send RPC message to a specified requester. In addition, the algorithm is a lightweight algorithm without increasing the client burden significantly.

VI. CONCLUSION

In this paper, we analyzed some matters for using RPC command to update routing table and presented a lightweight routing algorithm based on link analysis to resist routing table attack, which can make more actively nodes remain in the routing table. Some parameters are automatically adjusted in this algorithm to adapt to different network environment without extra bandwidth. The result of the experiment showed that this algorithm can rank the nodes in the routing table. As a result, the requester can make a trade-off between efficiency and security. With IPv6 implementation, node can obtain a unique ID by Hashing IP address. This rank algorithm will play a more important role for create and update routing table in order to route with more efficiency and reliability.

REFERENCES


Jian Wang received his M.S. degree from College of Computer Science, Sichuan University, China, in 2009. He is currently working toward the Ph.D. degree at the same school. His research interests mainly focused on computer network, information security, and especially peer-to-peer technology.

Xingshu Chen received her Ph.D. degree from Institute of Information Security at the University of Sichuan of China, in 2004. She has been a professor at the Computer Science College of Sichuan University, China, since 2005. Her general research interests lie in peer-to-peer networks, information security and computer networks. She is currently the director of the Network and Trusted Computing Institute, Computer College of Sichuan University.

Dengqi Yang received his M.S. degree from College of Computer Science, Yunnan University, China, in 2006. He is currently working toward the Ph.D. degree in School of Computer Science, Sichuan University of China. His research interests mainly focused on computer network, information security include and peer-to-peer technology.