iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 2 **Issue:** XII **Month of publication:** December 2014

DOI:

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)

Improved Chord Algorithm

Sonia[1], Mr. Jitender Arora[2]
[1]*Research Scholar,* [2]*Professor, RIMT, Chidana*

***Abstract**: A peer-to-peer networks (P2P) has increased popularity and is now more than a day. Peer-to-peer overlay network on top of the system Device or abstract physical network topology. The network consists of two types of structured and unstructured networks, peer-to peer fashion. Structured peer-to-peer network structure does not apply to cover. Any offer for any unstructured peer network organization or network connection optimization algorithms. According to specific rules or structural arrangement of these networks, Contract algorithms offer specific performance and coverage topology. Therefore, the link to the structure of the peer network coverage set. Usually such as rope system, hash table (DHT) and are distributed indexes. Chord algorithm is relatively simple and successful alternative program to promote the dissemination of structured P2P networks, but to further improve the efficiency of research should be better. Therefore, this paper cord organization P2P network to improve the search process. To improve the efficiency of research, Introduction to the routing table in counter-clockwise direction, and (in the search clockwise) from the routing table. Simulation results for the method to reduce the number of regulatory reform chord job search process efficiency P2P networks show Jumped.*
Keyword used: P2P, DHT,

I. INTRODUCTION

Peer Network increasingly structured. Popular because the good performance and scalability. Distributed hash table is used for data traffic and generate less exploration program planning knots. Agreement protocol approach based on the popular and well-known DHT. Protocol chord design, simple protocol-peer network architecture regulator. Protocol chord and distributes well, scalability and stability, and load balancing. There are many improvements and water research protocol. In this paper, we focus on structured peer networks and string algorithms Chord protocol performance and efficiency of the proposed changes to improve.

II. CHORD ALGORITHMS

Structured peer protocol chord, solve problems, and finds an effective target node using the new approaches. This Protocol with provable performance, simplicity and provable health. Protocol chord is set major node.

A. Properties of Chord

Properties of the chord peer-to peer networks are as follows:

1) *Decentralization:* Protocol series are not friendly, peer .The use of any central server or super node. With others in the system itself, the importance of each node. In this system, it has no single point of failure, and network very strong [1]
2) *Availability:* Agreement in a constant state of change for network, it works well: that the contract can always find path to the result, if there is a network error, you must the main task and a large number of nodes sure network without the addition of the ropes. [1]
3) *Scalability:* Protocol chord can be used for very large systems, the cost of the cables logarithmic growth protocol search. [1]
4) *Load balance:* Because the contract with the distribution of Wire keys the same hash function. Therefore, the key is unified divided into the node [1].
5) *Flexible naming:* Basic structure of the cord without any restrictions, and his name is great flexibility In the amount of data. [1]

B. Basic Chord Algorithm

Have some form of cord algorithm in this section is only a basic discussion of all form.Chord algorithm [2] is the hash function in the network construction, discover the tools and information source to participate and processing contract. You can change the function of the fixed retail output side length (node information and resources Information). Hash function H (X) has the following characteristics: (H x), and generate a fixed-length output, where H (x) Operational data of any length. (H x) can be easily calculated

for a given Q, in which the process is reversed device has gained almost impossible XH. So it is clear from attributes hash functions can ensure the uniqueness of the data.

1) *Topological structure of Chord:* Protocol chord Stoical service [3] proposed distribution algorithm. The texts are calculated, taking into account the information shown in the map network resources and data cables pear. In the string value topographical all collegiate> <IDK. IDK retail value is to find the resources. Access to resources than the actual value from the storage site. There should defragment series each network node resources. Dual-bit identifier M- miracle result indivisible function. Retail node is the IP address of the node ID, and International Domain Names (IDN) is the number He spoke the fragmentation of resources, the same name of the key, and in the IDK. IDnis group [0.2 M-1], and Internationalized Domain Names (IDN) Arranged in a circle in the form of small macro cycles. Figure 1.1 is an example of chord ring based on the basic chord algorithm with M=3.so maximum allowable node ID= (23=8).

The Construction process: First, the node may need of a small ring at the line domains International, an organization (IDN) names. In Figure 1.1, the network contract and keywords 0, 1, 3 1, 2, 4, IDK on IDN or equal for the first time, clockwise, is set for a node ID5.Keyword value. Called the contract Khalifa decade IDK (Khalifa (K)). As a result of the follow-up to the node (k) is equal to or less than IDK ring line, and set in a clockwise direction after the first node in the nodes. Therefore, in the true capacity is the successor of the node (1) , and information and information resources on the node 2 = 3 Khalifa (2) information sources 4 and 5, the successor node (4); Khalifa (5) = 0.

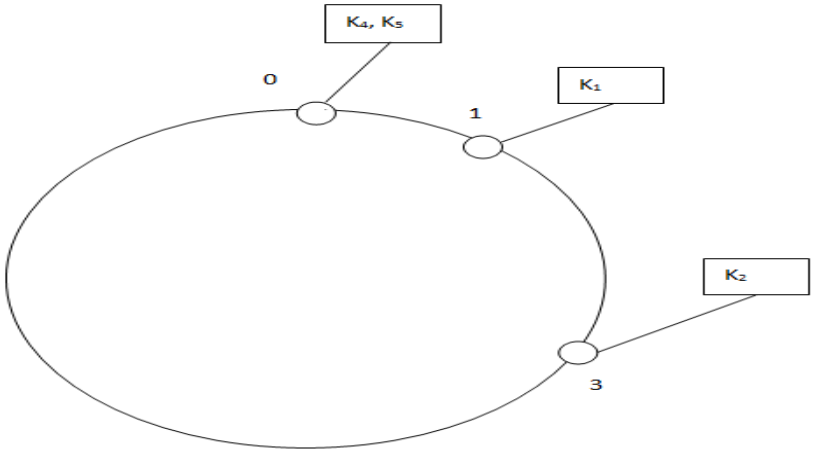


Figure 1.1: Chord Ring M=3

Figure 1.2 polyphonic ringtones to the needs of each node information for their successors to store information.

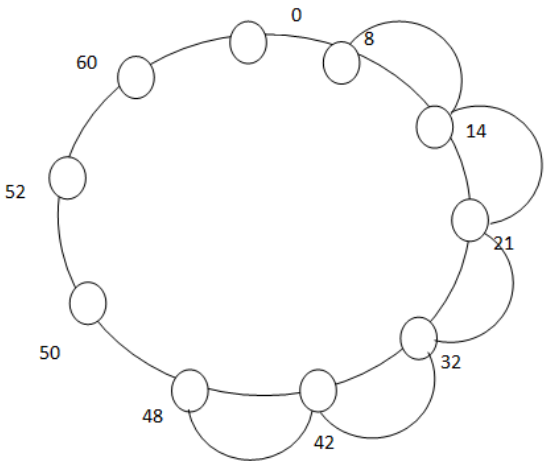


Figure1.2: Node 8 Searching Idk=45 in Basic Chord

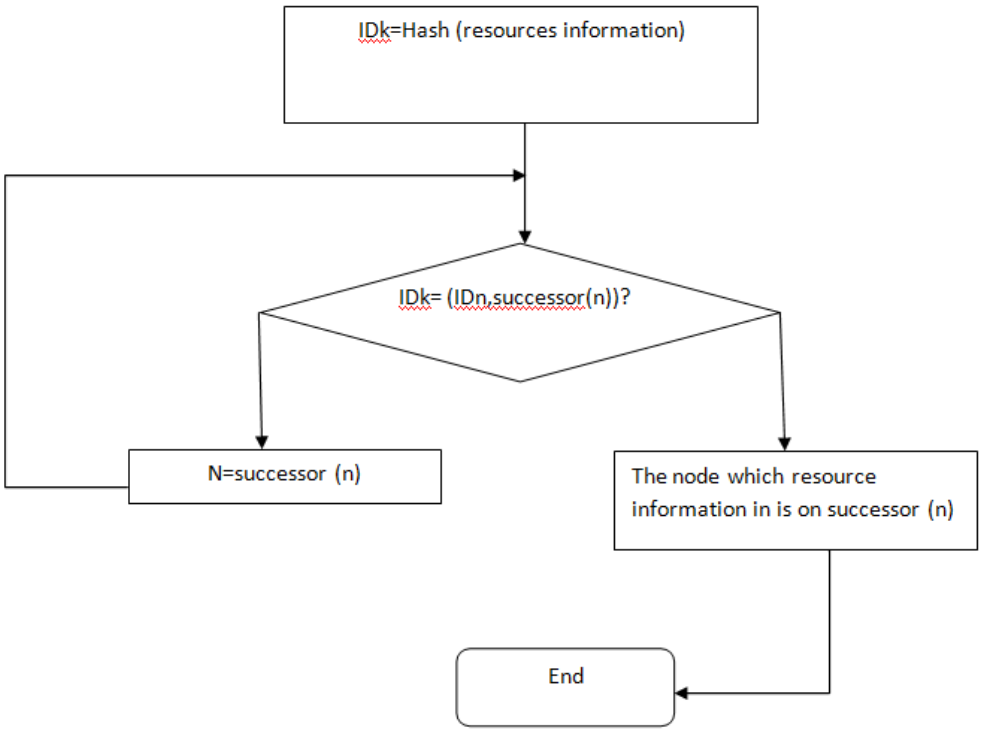


Figure1.3: Lookup Process in Basic Chord

Figure 1.3, M = 6, and the string network = 26 = 64, but in reality only 10 size the contract. This decade aware Khalifa glands. 8 is information stored 14S knots and 21 knots 14
Like. The search process continues for each polyphonic ringtones node and its successor Flow chart for above algorithm is shown in figure 1.3. According to flow chart in figure1.3: Get each node (s) in question (DP) first checks if IDK, IDN = or <Indonesia, Khalifa (n)> node, real or the result of Mandatory node or query message of the true successor (N) Repeat this process for each node. The algorithm is simple algorithm of the basic chords, and a layer of Efficiency, because the number of hops required to reach the worst possible destinations will node.tha n 1 times. and improved a copy of the chord algorithm, which is now being used on a large scale. The algorithm uses a digital form of fixed length, and the reduction of the hop count routing algorithm parameters.

2) *Improvement of the chord algorithm:* They are all ready to discuss your finger maintenance schedule existing agreements and the completion of a series of research for the clock. Thus, the search time is needed point problem is to search. To search for the key, which is located on the other side of the circular line clockwise to scan the network requirements? Therefore, the search process took a long time, the can reduce the effectiveness of the search process and protocol series. Therefore, it is necessary to improve the series search algorithm in order to improve the efficiency and reduce the search node time in the ring. We must consider chains protocol essential elements to quickly accept and effectiveness. The protocol chord keeps all colleagues and leadership information which makes it possible to use this guidance information improve efficiency searches chord.

Since time needed to search the key in any network can be as follows.

$$Time\ needed\ to\ find\ the\ keyword = average\ time\ per\ hop * \ hops\ look\ for\ \ key\ word$$

So reduction in the number of hops can reduce the search time. Chord protocol limits the use of paper in this project. A series of algorithms in order to solve the above problems. in this two project-oriented research, and to achieve a two-way table index. Find a way to use the process to shorten the search in

Table Polyphonic index. In other words, the concept of is for each node in the routing table of two tables: the first finger and second finger counter table. Due to the opposite direction, each node index table can jump closer to the goal of their colleagues. There are two routing tables are implemented on each node

a) Figure table stores the successors and their key mapping for the node.

b) Anti finger table stores the predecessors of the node.

So finger table links the peer of the network in clockwise direction where as anti finger table links them in counter clockwise. The ways data are stored on nodes remain same. Hence, no change in the network's topological structure. So in this improved chord algorithm there are three things maintain by each node.

a) Finger table

b) Anti finger table

c) Successor list

Finger table and successor list is same as the original chord algorithm. This thesis have introduced new antifinger table. There is no improvement if same half in ring have key .if key is in other half of the chord ring then has significant improvement in the lookup efficiency of the chord algorithm. Following example explains the process in figure 1.4 node 0 has predecessor node 6 and 7 and node 0 is in the finger table of the both table and hence both node 6 and 7 are in the anti finger table of the node.

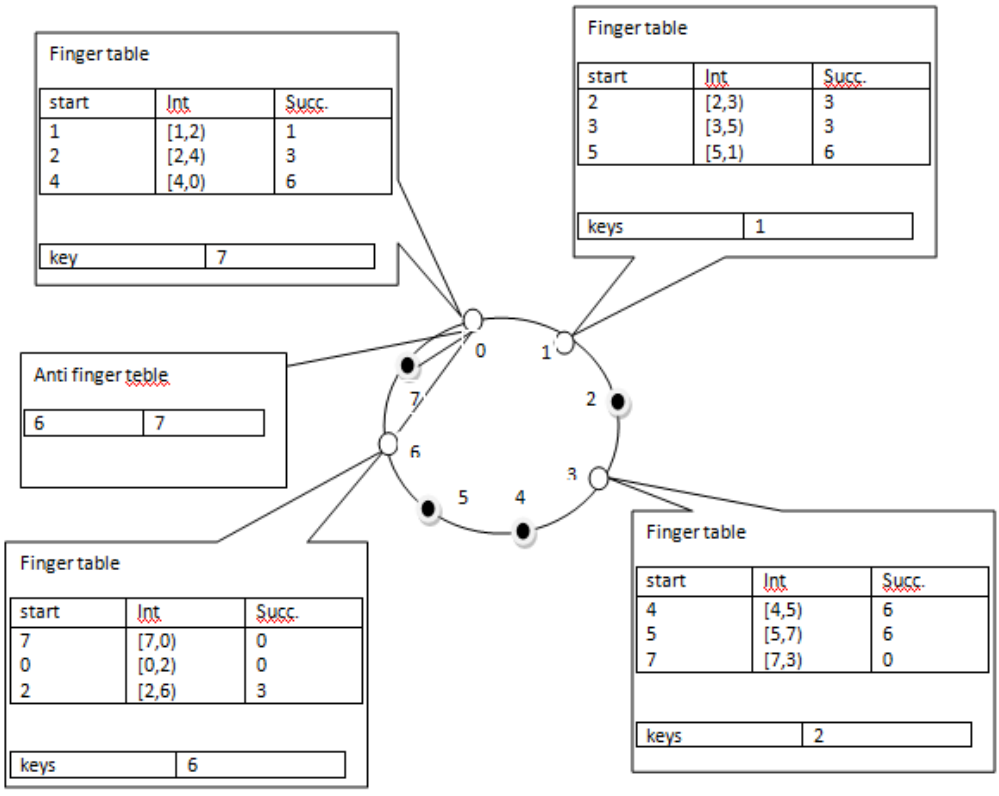


FIGURE 1.4: EXAMPLE

3) *Lookup algorithm:* In order not to change the original key and string needed to **c**hange the mapping of data search algorithm. To see the original algorithm, the main successor chord. In to improve the algorithm, we have to store the predecessor table node fingerprint resistant, so you cannot use previous research algorithm to find a unique key. The use of anti-traces key paper its predecessor is looking for his successor. The original drawing key cards to retain data. Sometimes it does not require the entire

research process: Using hostility fingers. Only the use of a sheet resistance in perspective of fingerprints, and if node in this table not exceed the key value. Node A used to find the underlying data. If the resulting data are resistant to fingerprints in the table and table with your fingers and search algorithms, in normal power cord. Find facilities clockwise and counterclockwise, the key can be found from the original algorithm.

C. Implementation

This section will examine the algorithm to change the original text discussion series Open Chord. Chord algorithm is Open chord and open source platform simulation. Finger table does not change, so do not require modification of the definition of the finger table. It is known that the table entries of the finger. Table entry similar finger on the table was too. Algorithm used to calculate the number of hops the algorithm performance. Since implementation of the simulator used for small scale (<50 nodes), it is also based on the JVM (Java Virtual Machine). So polyphonic ringtones lookup speed is high, so we avoid the timer to calculate costs.

Pseudo code of proposed algorithm is:

List1.1: Pseudo code of the modified chord algorithm

```
Node findPredecessor(key,n){
Node pred=n.getPredecessor();
if (pred==null)
return n; //n is the current node
else if (key.isInInterval(pred.ID, n.ID) //check if the key
is between the pred and current node
return n;
else {
Node n'=getClosestPrecedingNode(key) //if not, track
the closest preceding node and lookup again
return findPredecessor(key,n') // recursively
find the predecessor of node n'
}
}
Node findSuccessor(key,n){
Node succ=n.getSuccessor();
if (succ==null)
return n; //n is the current node
else if (key.isInInterval(n.ID, succ.ID) //check if the key
is between the current node and successor's node
return n;
else {
Node n'=getClosestPrecedingNode(key) //if not, track
the closest preceding node and lookup again
return findSuccessor(key,n') // recursively find
the predecessor of node n'
}
}
Set<Serializable> retrieve_R(key){
hops_R=0; //initialized the hops counter in
anti-finger table direction
while(!retrieved){
Node responsibleNode_R=null;
responsibleNode_R = findPredecessor(id);
```

```
hops_R+=1; //while not retrieve the desired
key, add the hop counter by 1
try{
result_R = responsibleNode_R.retrieveEntries(id); //
get the responsibleNode to fetch the entry
retrieved = true; //if successfully get the value,
set retrieved state to true
}catch(Exception e){}
continue;
}
}
if(result_R !=null)
values1.add(entry.getValue()); // add the lookup result
to the valueset
final_hopsR=hops_R; //get the hop counter for
the current lookup operation
return values1;
}
Set<Serializable> retrieve(key){
hops=0; //initialized the hops counter in finger
table direction
whild(!retrieved){
Node responsibleNode=null;
responsibleNode = findSuccessor(id);
hops+=1; //while not retrieve the desired key,
add the hop counter by 1
try{
result = responsibleNode.retrieveEntries(id); // get the
responsibleNode to fetch the entry
retrieved = true; //if successfully get the value,
set retrieved state to true
}catch(Exception e){}
continue;
}
}
if(result !=null) values.add(entry.getValue()); // add the
lookup result to the valueset
final_hops=hops; //get the hop counter for the
current lookup operation
return values;
}
```

Of course, the counter is set to jump pseudocode results of research, research and research clockwise
Put processes in the counterclockwise direction. Rear jump value leading ultimately to the research process.The total value will be helpful to view or modify water relative to the performance of the algorithm. Choose to continue automatically between clockwise or counterclockwise problem, we only get about after the jump using the lookup table and the table title ways antifinger fingers of original algorithm chord.

List1.2: Pseudo code of lookup method of chord Algorithm

```
// ask node n to find the successor of k
n.find_successor(k) {
      if (k∈(id(predecessor), id(n)])
          return n;
      else
          if (k∈(id(n), id(successor)])
              return successor;
          else {
              n'=closest_node(k);
              return n'.find_successor(k);
          }
}

// search the local routing table for the closest
// node to k, R denotes the routing table
n.closest_node(k) {
      x=R(1);
      for i=1 to ||R||
          if (|R(i)−k|<|x−k|)
              x=R(i);
      return x;
}
```

By using the above method using the X1 and X2 on Request to the next node in the direction of clockwise and determine the anticlockwise direction. After that account the distance between the X1 and X2, and we look forward less than the distance from the questions of destination Knot. Such that D1 = | X1-K | D2 = | X2-K |, if D1 is less than D2 Question of the X2 X1 opposite. Using this method to select the forward direction or in the other direction can be automated.

III. RESULTS

Open chords for the implementation and performance evaluation algorithms simulator modified chords. Will be used to improve the Chord algorithm and the original Chord algorithm to compare the performance of the two algorithms.

A. Test Results

The tests results will be show in term of two values. Which are as follows:

1) *Routing table size:* Changing the routing table algorithm has no input into the program. The size of a fingerprint-resistant, and the total number of the table **e**ntries. Because it is all ready knowledge, and not enter the finger table **l**ess than or equal to M. In the experiment, and the network **s**imulate different nodes and fingers tables account numbers. The footprint of the resistance of the entries in the table, in the jump in the direction of clockwise **a**nd direction counterclockwise. Size almost every peer routing table 2M.

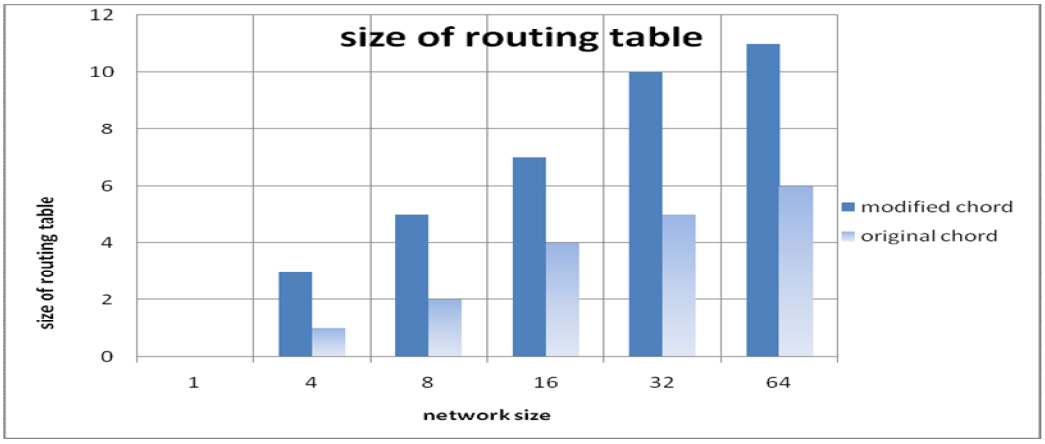


Figure1.5: comparison of routing table size for original and modified chord algorithm.

2) Look up Hops

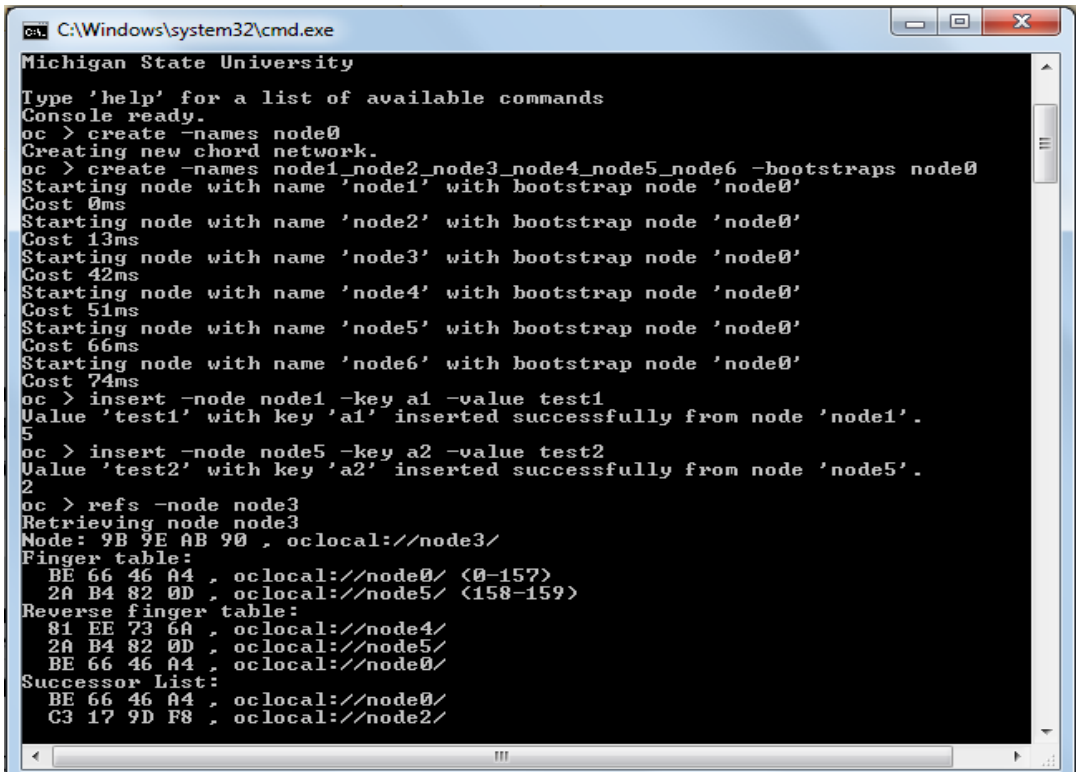


Figure1.6: Screenshot #2

Search process performance algorithm to calculate the modified string comparison, we chose the original chord find and modify the look and recorded on the same node and search the same key. Inspection process, for different network size.

No of nodes	Entries in finger table	Size of finger table	Hop count
1	0	0	1
4	2	8	1
8	3	24	1
16	4	64	1
32	5	160	2
48	6	288	2

TABLE1.1: Evaluation Table for Chord Algorithm.

In table 1.1 numbers of entries and size of the finger table are calculated and also calculated the hop count for different network size using original chord algorithm.

NO of nodes	Entries in finger table	Size of finger table	Entries in anti finger table	Size of anti finger table	Hop count(clockwise direction)	Hop count(anti clockwise direction)	Total entries in routing table
1	0	0	0	0	1	1	0
4	1	4	2	8	1	1	3
8	2	16	3	24	1	1	5
16	4	64	3	48	1	1	7
32	5	160	5	160	2	1	10
48	6	288	5	240	2	2	11

TABLE1.2: Evaluation Table For Modified Chord Algorithm.

Table 1.2 shows result for modified Chord algorithm. It is clear from table 3 that, use of anti finger table, make it possible to reduce hop count to search any key. To get the better result test this in an environment which is capable of generates nodes equal to the actual network.

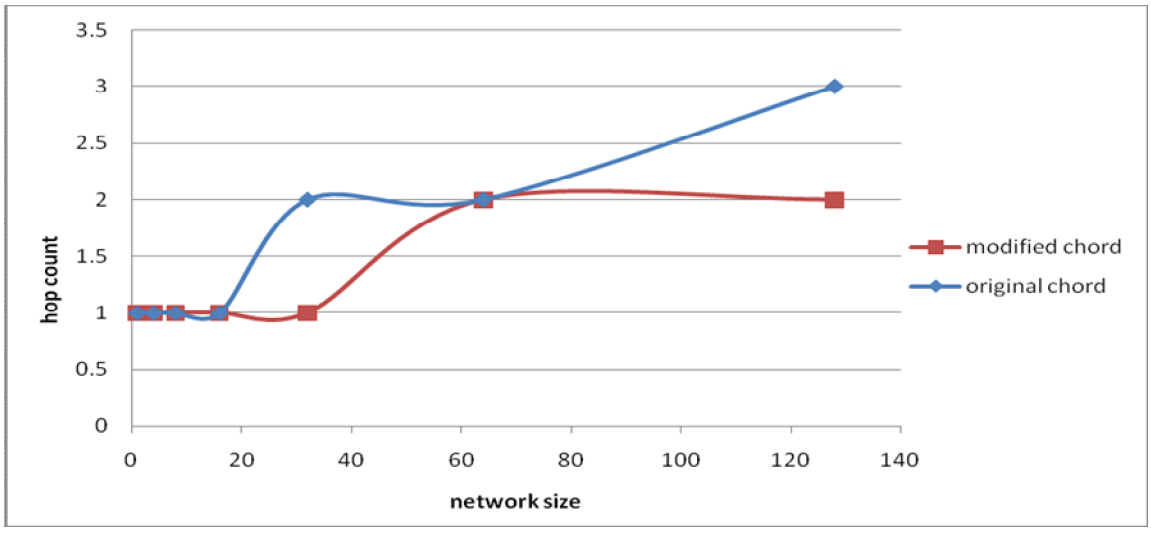


FIGURE1.7: Comparing Hop Count for Original And Modified Chord

In the graph of Figure 1.7 represents less rope algorithm jumps best original cord algorithm as the size of the network is to increase comparison. Thus, experience has shown that the update rope cord algorithm search algorithm has improved the original efficiency and performance is better.

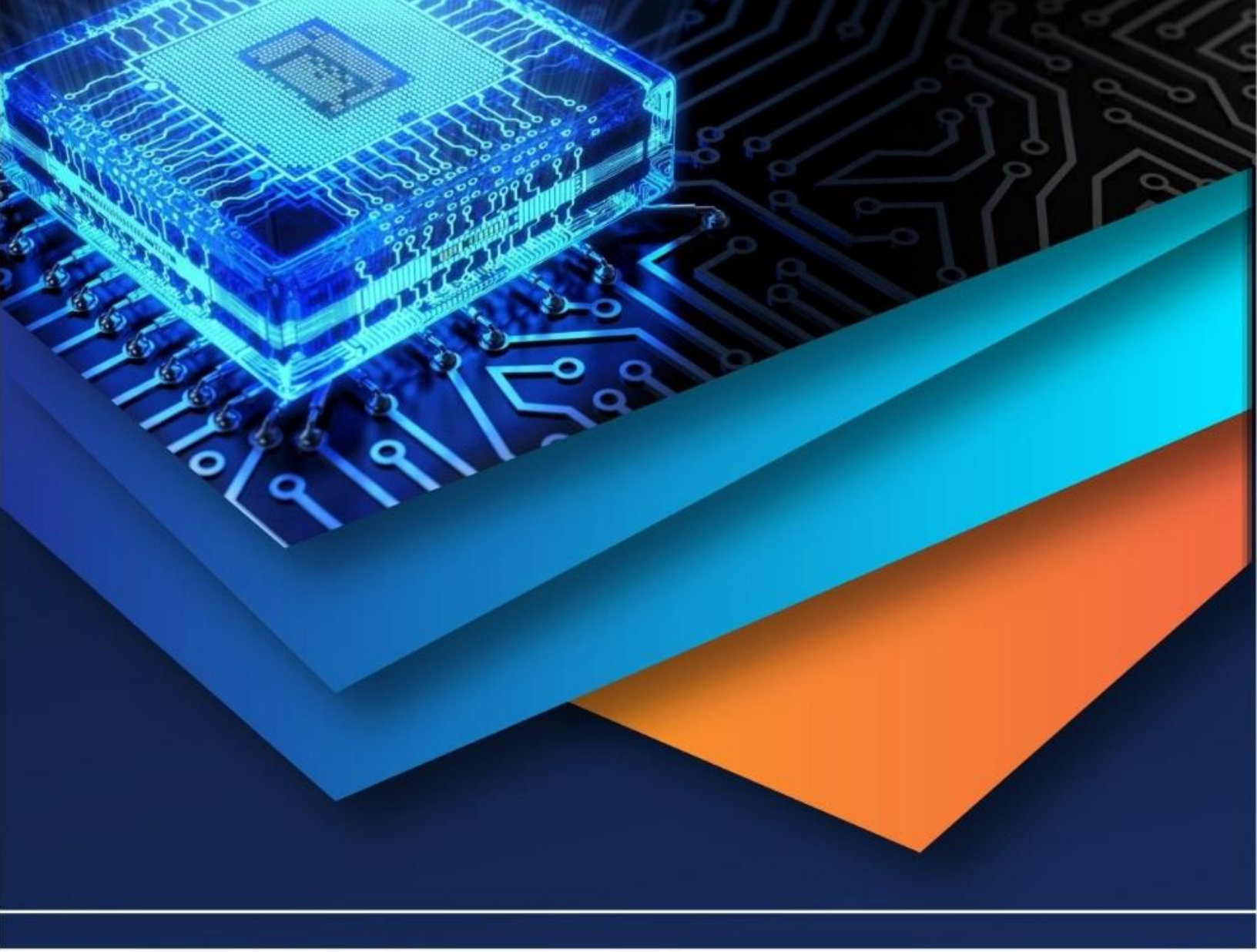
IV. CONCLUSION

Table key search algorithms use the original chord in finger preferably a direction in the direction of clockwise. It takes more time to find the hope to search for a specific on the other side of the ring line key .Thesis proposal routing table search algorithm in each

agreement clockwise and counter-clockwise directions. To this end, it Use tables finger print resistance and tables. Appearance the effectiveness of the water of the algorithm. Ropes open water simulation algorithm to simulate the changes. Through simulation result modified algorithm jumps rope, need to find a specific network to reduce the number of Keys. Chapter 5, it is shown that, with the increase of the size of the network, More hops to search for key rings lying on the other side of the
Table using your fingers only, but using antifinger table .you can search for a particular key, no fewer hops. Therefore, the proposed algorithm is effective and rope algorithm ropes reduce the search time.

REFERENCES

[1] I. Stoica R, Morris D, Karger M Kaashoek F, Balakrishnan H, Chord: a scalable peer-to-peer lookup service for Internet applications, In Proc ACM SIGCOMM01, San Diego, CA, Aug, 2001.
[2] De-gan Zhang, Yu-xia Hu,Dong Wang and Yan-pin Liang, A New Algorithm of Service Discovery Based on DHT for Mobile Application, JOURNAL OF NETWORKS,VOL. 6, NO. 10, OCTOBER 2011.
[3] Eng Keong Lua, Jon Crowcroft, Marcelo Pias, Ravi Sharma and Steven Lim, A Survey and Comparison of Peer-to-Peer Overlay Network Schemes, IEEE COMMUNICATIONS SURVEY AND TUTORIAL, MARCH 2004
[4] R. Denneman, P2P searching methods, research issues, solutions and their comparison, 11th
Twente Student Conference on IT, Enschede, June 29th, 2009
[5] John Risson, Tim Moors,Survey of research towards robust peer-to-peer networks: Searc methods,ScienceDirect, Computer Networks 50 (2006) 3485–3521
[6] Chen Gang, Wu Guoxin, Yang Wang. “GChord: an improved routing algorithm for Chord”. Journal of southeast university (Natural Science Edition), Vol.37, 9-12. Jan.2007.
[7] Jiajing Li, Xu Dongyang. “An Optimized Chord Algorithm for Accelerating the Query of Hot Resources”. International Symposium on Computer Science and Computational Technology, Vol.2, 644-647. Dec. 2008
[8] Hong Feng, Li Ming-Lu. “SChord: Handling Churn in Chord”. Journal of Nanjing University (Natural Science Edition), Vol.41, 288-293. Oct, 2005.
[9] Sonia G, Habib Y. “Improving Chord Network Performance Using Geographic Coordinates”. Proc of the 3rd International Conference of GeoSensor Networks. Berlin: Springer, 2009.
[10] Wang Biqing, He Peng. “L-Chord: Routing Model for Chord Based on Layer-Dividing”. International Conference on Computational Intelligence and Security, Harbin, China, Dec. 2007: 262-265.
[11] Hongwei Chen, Zhiwei Ye. “BChord: Bidirectional routing DHT based on chord”. 12th International Conference on Computer Supported Cooperative Work in Design, Xi’an China, April.2008: 410-415.
[12] Ashok Kumar Ojha, Krishna Kant. An Efficient Lookup Algorithm for Dynamic Peerto-Peer Chord . IEEE2010
[13] http://open-chord.sourceforge.net/
[14] http://en.wikipedia.org
[15] (2001) Kazaa media desktop. [Online]. Available: http://www.kazaa.com/
[16] Napster. [Online]. Available: http://www.napster.com/
[17] (2001) Gnutella development forum, the gnutella v0.6 protocol.[Online]. Available: http://groups.yahoo.com/group/the gdf/files/
[18] Gnutella website: http://www.gnutella.com



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY