IJCSE International Journal of Computer Sciences and Engineering Open Access
Review Paper Volume-3, Issue-9 E-ISSN: 2347-2693

A Carnal Map-Aware Chord Model Created on ANT Colony Optimization

N. Thamaraiselvan[1*] and A. Immaculate Mercy[2]

[1*,2] *PG and Research Department of Computer Science, Maruthupandiyar College, Thanjavur.*

www.ijcseonline.org

Received: Aug/22/2015 Revised: Aug/30/2015 Accepted: Sep/24/2015 Published: Sep/30/2015

***Abstract*—** On the basis of in-depth analysis of the mapping storage methods of the Identifier-Locator separation network, this paper proposed a physical topology awarded Chord model (Ant-Chord) which based on ant colony algorithm. The ideas of Ant-Chord is to regard the storage nodes in the whole Chord as a TSP problem and solve the TSP problem quickly by using the ant colony algorithm, then to build the Chord with the obtained TSP solution, and proposed a method which called "Luoyang shovel" to optimize the Ant-Chord's routing hops. The model is simple and easy to implement, which has small changes within the original Chord model and little extra overhead cost in the routing table storage. Simulation results show that, AntChord has obvious advantages in average routing hops and delay in comparison with other Chord model.

***Keywords*—** ACO, TSP, Physical Map Aware, Luoyang Shovel Method

I. INTRODUCTION

With the dramatic increase in the number of Internet users, traditional Internet has increasingly shown its initial design problems with its deficiencies in supporting network mobility and scalability. It has been widely accepted that the separation of identifier and routing locator is a feasible way to solve the ambiguity of IP address and some significant results have been achieved, but the complete identifier-locator separation mechanism has not yet formed. The Universal Network architecture[4-5] which proposed by Chinese experts solved the ambiguity problem of IP address by defining Access Identifier(AID), Routing Locator(RLOC) and its mapping theory. The identifier-locator separation network needs a new set of management mechanism to store and manage the Identifier-to-locator mapping, and the mapping query speed will directly affect the efficiency of communication.

Chord is a classic structured P2P network model, which is characterized by simple implementation and relatively efficient routing hops that its average hop count is O (logN), so the query efficiency of Chord is high. Therefore, we chose to introduce Chord to the identifier locator separation network, used for the storage and query of identifier-locator mapping. We conduct HASH operation to RLOC, and according to the obtained ID from HASH to store the corresponding mapping of the RLOC into the Chord.

However, as a logical topology model, Chord lacks the use of information of the physical network topology, so two logical adjacent nodes are often not adjacent in the physical layer, and then a logical short path will not be a physical short path. The performance difference of query and location between the logical layer and physical layer greatly reduced the practical efficiency of resource query in Chord. Therefore, if the mismatch problem between the Chord and underlying physical topology could be solved, it will further increase the application prospects of Chord, and then meet the mapping storage and query needs under the identifier-locator separation network.

Based on the research of related mapping storage methods, this paper proposed a Chord model (Ant-Chord) which based on ant colony optimization (ACO) algorithm, to store and query the identifier-locator mapping. The ideas of Ant-Chord is to regard the storage nodes in the whole Chord as a global TSP problem and obtain the approximate optimal solution of the TSP problem by using the ACO algorithm, then to quickly build a physical map-aware Chord, and proposed a method which called "Luoyang shovel" to optimize the Ant-Chord's routing hops. OPNET simulation results show that, AntChord has obvious optimizations in average routing hops and delay of resource discovery.

II. RELATED WORK

A. Typical Physical Topology Matching Chord Algorithms

Up to now the research on the physical topology matching has formed three main ideas within the P2P network: (1) the optimization of node ID assignment based on topology, whose basic idea is to map the ID space of the logical network to the physical network, so that two nodes adjacent on the ID space are physically close too. However, this method will lead to uneven node distribution in the ID space and intensify the load imbalance. (2) Optimization based on proximity routing: building the overlay network without regard to the underlying physical network, when it is routing, selects the nearest node to destination node from the routing table as the next transit node. The effect of this idea depends on the number of candidates provided by the routing table, the more candidates, the better, but also increased the burden on the routing table. (3) Optimization based on proximity neighbor selection: the node ID space is divided into several regions that nodes with similar or close node ID are in the same region, when a node N want to communicate with other neighboring regions, the routing table selects the nearest node to N in the destination region as the neighbor of N, and in accordance with this to build the overlay network. Focus on basic Chord mode which based on the structured P2P technology, it also has been improved based on the above three methods, respectively.

In the authors propose ChordPNS algorithm, each routing table maintaining a set of candidate nodes and selects the proximal node from the candidate nodes to route the query request. The algorithm reduces the query delay, but within the process of proximal node discovery and routing table maintenance, ChordPNS bring heavy network traffic overhead.

In the authors use the proximity list to obtain a better network approximation than the standard Chord model. Proximity list stores a set of nearby nodes of a node which discovered in a life period, next hop is decided by the proximity list and finger table jointly. Although this method gets a better routing efficiency than basic Chord, it has the problems of slow convergence and low efficiency when nodes frequently enter and exit the network, which will lead to the relative reduction of life period in nodes and rapid changes in dynamic network topology.

In the authors propose the Chord6 which making use of the addresses hierarchical nature of IPv6 to generate the node ID, so that each node in the same domain with more approximate ID, and the average path length is reduced from O (logN) to O (logM) by this method, where N is the number of network nodes, and M is the number of network domains. However, nodes between two approximate domains may be very far, which is determined by the nature of Hash function.

B. TSP and ACO Algorithms

Traveling Salesman Problem (TSP) [7] is well-known in the field of mathematics, which supposes there is a traveling salesman want to visit N cities, so he must choose a path to go, the restrictions of the path are that he have to visit each city only once, and at last go back to the original departure city. TSP is the problem asking the shortest path among all possible paths, which is a NP hard problem.

The reasons why we introduce the TSP problem to Chord, on the one hand because all the nodes in Chord can form a closed loop ordered by their IDs, on the other hand because the problem we want to solve in Chord is also a physical path length problem. According to the analysis of existing algorithms in 2.1, adding proximity routing table and dividing proximity physical region is the two main ways to achieve physical topology match. Adding proximity routing table has the minimum changes with the Chord structure, but increases the routing table overhead. Dividing proximity physical region, on the one hand needs to solve the balances between the number of regions and the number of nodes within a region, on the other hand it is opaque between each regions, so the efficient of inter-domain communication is not ideal.

Meanwhile, the above two methods are essentially used the idea of nearest neighbor that every time you try to find is the nearest node to one node but lack of considerations to the global topology, while a query is often a global routing process, and with increase in the number of nodes and complexity of network topology, this short-sighted approach is easy to fall into a poor local optimal solution. Therefore, we consider taking advantage of the global nature of TSP problem to produce better results.

As a widespread concerned academic issue, TSP problem has been researched for a long time and is also a standard test platform for lots of new algorithms. Metaheuristic algorithm is very successful in solving TSP problem and ant colony optimization (ACO) is one of the classic meta-heuristic algorithms. In fact, the first ACO algorithm -- ant system, and a number of the subsequent proposed ACO algorithms, were all first tested by the TSP problem. So far, ACO algorithm has become a very sophisticated algorithm in solving TSP problem, therefore, we choose ACO algorithm to solve the TSP problem within Chord.

In a real network, we cannot previously or immediately know its TSP optimal solution, thus unable to confirm the solution obtained by ACO algorithm is the optimal solution. At the same time, because of the constraints of network communication efficiency and computational cost, so to quickly get a TSP optimal solution with large-scale nodes becomes very unrealistic. As a meta-heuristic algorithm, ACO is in pursuit of a good solution or approximate optimal solution in a relatively low computational cost, but the algorithm does not guarantee to find the optimal solution. So our actual purpose of choosing the ACO algorithm is to quickly obtain the approximate optimal solution of TSP problem, not the optimal solution itself. The simulation in Section 4 will prove that the approximate TSP optimal solution which quickly obtained by ACO, its performance is satisfactory.

III. ANT-CHORD ALGORITHM

In order to quickly solve the TSP problem within AntChord, this paper used the ant colony system (ACS) algorithm with some simplifications and made corresponding adjustments and optimizations according to Chord features.

A. Ant-Chord Algorithm Description

Ant-Chord algorithm is described below, set the number of storage nodes in Chord is *n*; the number of ants is *m*.

1、Data initialization

Step1 Import network topology and get the communication delay(indicate the path length between nodes) between nodes by node broadcast, then save the information in the distance matrix *dist[n][n]*, where the value of array element *dist[i][j]*(*i*,*j*[0,*n*-1]) which represents the path length from node *i* to *j* is shown in (1).

$$dist[i][j] = \begin{cases} v_{delay} & i \neq j \\ 0 & i = j \end{cases} \quad (1)$$

Step2 Initialize the pheromone matrix *pheromone[n][n]*, where the value of array element *pheromone [i][j]* (*i*,*j*[0,*n*-1]) which represents the pheromone content of the path from node *i* to *j* is set a uniform initial value as shown in (2).

$$Pheromone[i][j] = v_{init\_pheromone} \quad (2)$$

Step3 Initialize path choice matrix *choice[n][n]*, where the value of array element *choice [i][j]*(*i*,*j*[0,*n*-1]) is shown in (3), it can be seen , the value of *choice [i][j]* is proportional to the pheromone content in its path, and inversely proportional to the distance.

$$choice[i][j] = \begin{cases} \dfrac{phreomone[i][j]}{dist[i][j]} & i \neq j \\ 0 & i = j \end{cases} \quad (3)$$

Step4 Initialize the parameters of ants and algorithm, such as the maximum iterations *epoch_max*, the length of global optimal path *dist_best,* the lower limit of pheromone content *pheromone_min* and so on.

2、Iteration

When the iteration number is equal to *epoch_max*, the iteration end; otherwise circulating the following acts:

Step1 Clear the way-finding memory of all ants on the previous iteration;

Step2 All ants were randomly assigned to an initial node;

Step3 Every ant builds a full path: in each step of the path construction, ant *k* (*k*[1,*m*]) in accordance with the roulette wheel selection method to decide the next node to go. The characteristic of this selection method is that the value of *choice[i][j]* corresponding to a path is greater, the probability of the path chosen by ant k is greater. So on the one hand ants tend to choose the paths which is considered better currently, on the other hand the other paths are also have the chance of being selected that can avoid algorithm falling into local convergence prematurely. The probability of an ant in node *i* will choose the node *j* as its next visit node is shown in (4), where N represents the set of nodes has not yet visited by the ant.

$$p_{ij} = \frac{choice[i][j]}{\sum_{l \in N} choice[i][l]}, \qquad j \in N \quad (4)$$

Step4 Ants return to the starting node, and calculate the length of the paths it constructed, then reduce the pheromone content of each path it visited (set the reduction as V_{sub}), so that the probability of other paths chosen by ants in next iteration is increased which could prevent the algorithm falling into a local optimum. If the pheromone content of a path is less than *pheromone_min*, then make it equal to *pheromone_min*, which can avoid the pheromone content of a path is too low so the path can hardly be selected;

Step5 Select the ant which constructed the shortest path in this iteration and increase the pheromone content (set the increment as V_{add}) of this path, which is to reward the best path in this iteration so the probability of this path chosen by ants in next iteration is increased, and V_{add} is significantly greater than V_{sub}. If the length of this path is less than *dist_best*, then use it to replace *dist_best*, and record the node order of this path; Step6 Go to the next iteration.

3、Chord construction

Use the path order recorded by *dist-best* to build the Chord, and implement the hops optimization strategy which named "Luoyang Shovel Method", as detailed in Section B.

Within the construction of the Chord, the ID of storage nodes consists of two parts, if set the length of ID is *L* bits, then the front L_1 bits of ID represent the TSP sequence information, and the back L_2 bits is the result of unity hash of the node ID, so that all nodes in accordance with its previous L1 bits of ID to build an Ant-Chord.

B. Luoyang Shovel Method

Simply matching the physical topology has little effect to the logical routing hops of Chord, so we designed a Chord routing hops optimization method called "Luoyang Shovel Method (LSM)" which records the storage space information of successor nodes when generating the routing table of Chord nodes to increase the Chord routing flexibility and achieve the optimal routing hops. The method draws on the idea of Chinese archaeological tool named "Luoyang Shovel" which is a semi-cylindrical shovel, and stamp down the ground straightly when using it then the semi-cylindrical shovel can bring out the underground soil. So that with the process of stamping we can not only learn the depth of the ground, but also understand the soil conditions corresponding to each depth, and thereby can have comprehensive analysis of whether there are ancient tombs underground. Therefore, we call the method which recording storage space of successor nodes to achieve optimization of routing hops as "Luoyang Shovel Method."

Fig.1 shows a simple query process in Chord, node N0 need to find the resource with ID 31, although from the figure we can immediately determine that the resource is stored in the node N32 which is also a successor node of N0, but in the actual query process in Chord it need to go through the routing process as N0-N16-N24-N32 shown in Fig.1, which caused by the relative lack of flexibility of Chord for the simplicity of Chord. We mentioned the routing hops of Chord are already quite efficient, but at this point there is still not a small room for improvement.

In order to optimize the routing hops of Chord, there already has a lot of programs [12], most of them focus on the expansion of the routing table, such as the classic two-way routing table program which adding a routing table can be reverse queried, so that extends the half-ring resource positioning of original Chord to the full-ring positioning, increased the query flexibility of Chord, but the half-ring positioning compared to full-ring positioning, the difference is actually just one routing hop and only when the ID of the query resource is bigger than the maximum node ID in one-way routing table, so the improved performance of two-way routing table program is limited, but the expansion rate of the routing table is great.

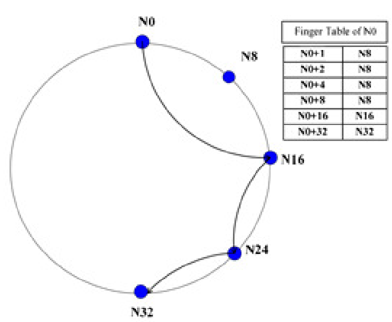

Figure 1. A Chord query process

Analyze the Fig.1 can find that if N0 knows the storage space of successor node N32, then N0 can send the resource query of R31 directly to N32. Therefore, in Luoyang shovel method, we record the storage space information of successor node when adding each successor node to the Chord routing table. As shown in Fig.2, entry "N32 (8)" in the routing table of N0 means the storage space of successor node N32 is 8, in other words the resources with ID from 17 to 32 are stored in the N32. Then N0 in the query of R31 can directly determine the resource belongs to its successor node N32, and thus do not need to experience the long process shown in Fig.1. Luoyang shovel method is still an amendment to the original contents of the routing table, while only need to record the storage information of the large successor nodes (as in Fig.2 only records the N32), so the storage overhead of the routing table is relatively small. From the following simulation results in Section 4 we can see that for a Chord with a capacity of 16384 nodes, only need to record the storage space information of 8 successor nodes in the routing table can make the routing hops in Chord having a very significant reduction.

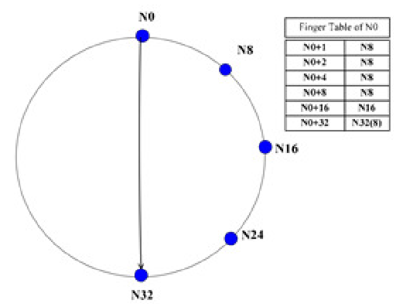

Figure 2. A "Luoyang shovel method" query process

C. Ant-Chord algorithm realization

In order to facilitate understanding of the specific realization of Ant-Chord algorithm, we list the data structure of ants and pseudo code of the algorithm procedure, details are as follows.

```
Data structure of ants:
Structure  single_ant
begin
    double  tour_lengh
  integer  visited[n]
    end
single_ant  ant[m]
Pseudo code of the Ant-Chord procedure:
procedure  AntChord
  InitializeData()
while (step < epoch_max)  do
for  k = 1 to  m  do
ConstructionPath(k)
        PheromoneSubtraction (k)
end-for
```

PheromoneAddition (*this_best*)
 if *this_best* < *dist_best* then
 dist_best ← *this_best*
 end-if
 end-while
 ConstructionChord()
 end-procedure

VI. SIMULATIONS AND PERFORMANCE ANALYSIS

Verification experiments using OPNET 10.0 as the simulation platform, and the experimental topology data is generated by the network topology generator NEM (Network Manipulator) with using the magoni_pansiot_ sampling method. We generated two Chords with the capacity of 4,096 nodes and 16,384 nodes respectively, and chose the original Chord (denoted as Random-Chord) and the dividing physical region Chord (denoted as Zone-Chord) for simultaneous simulations, as the experimental references of Ant-Chord. Simulation time was 3 hours, the packet interval is 1 second and the number of ants used in Ant-Chord algorithm is 10.

A. Simulation Model Design

In the OPNET simulation experiment, the nodes in Chord are divided into construction nodes and ordinary nodes. Each Chord has one construction node, the rest are ordinary nodes. The difference between the two nodes is that construction nodes will implement the Chord construction work after nodes broadcasting, and then it will begin the normal Chord routing simulation as the ordinary nodes. In the process of routing simulation, each node will generate a packet with random resource ID, if the resource ID of the generated packet belongs to its generation node, then this is a self-query and the routing hops and delay are recorded as 0. Otherwise, continue to forward the packet in accordance with the Chord routing rules until it reaches the destination and record the corresponding data of this packet forwarding.

Fig.3 shows the Process Model of “processing” in the construction node. OPNET uses the discrete event-driven mechanism to simulate the actual network, as the simulation begins, the “processing” module sets the related parameters of data acquisition of OPNET in “init” State (the state only executes once) and then transfers to “idle” State, this state will wait for event-driven to process accordingly and after processing will return to “idle” State waiting for the arrival of the next event.

In the entry of "(PKT_ARRVL) / chord_route()" as shown in Fig.3, "PKT_ARRVL" indicates that there is a packet receiving event, then the “processing” module will perform "chord_route()" operation. Here "chord_route ()" is divided into two phases, the first is the node broadcast phase that "chord_route()" will receive the packet generated by the generation process of its own node and then send the packet to other nodes, or receive the packets from other nodes and record the communication delay; second phase is after the node broadcast finished, at this time is the same as its name indicates, the implementation of regular Chord routing policy.

In the entry of "(SELF_INTRPT) / set_chord()"as shown in Fig.3, "SELF_INTRPT" is set to generate a self-interrupt event after the node broadcast finished, then the “processing” module will perform "set_chord ()" operation which used to construct a Chord (the specific construction method is decided by the type of Chord of each experimental scenario), then according to the Chord to generate its own routing table. When this event is processed over the simulations formally enter the regular Chord routing phase and begin the acquisition of the performance data. The difference between Ordinary nodes and construction node is embodied in this entry, the ordinary node uses the self-interrupt event to simply generate its own routing table, without having to construct a Chord.

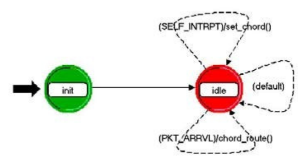

Figure 3. The processing module of the Ant-Chord construction node

B. Performance Analysis

Fig.4 to Fig.7 show the simulation results of the 4096Chord, blue, red and green lines represent the AntChord, basic Chord and the dividing physical region Chord, respectively. Meanwhile, in order to facilitate comparative analysis, we also added two additional experiments, which are the Ant-Chord without using

Luoyang shovel method (denoted as ANT_ CHORD_without_LSM) and the basic Chord with using Luoyang shovel method (denoted as RANDOM_CHORD_with_LSM), represented by blue and yellow lines respectively.

In the delay, we can see the Ant-Chord and the dividing physical region Chord are significantly better than the basic Chord and the Ant-Chord is better than the dividing physical region Chord. The basic Chord with using

Luoyang shovel method because of its routing hops optimization, so the delay is corresponding lower than the basic Chord, but the advantage of the delay is not very obvious compared with the Ant-Chord without using Luoyang shovel method, which can therefore explain the main reason for delay reduction in Ant-Chord is the TSP optimization, rather than Luoyang shovel method. In the routing hops, both the Ant-Chord and the basic Chord which used Luoyang shovel method have reflected a significant advantage.

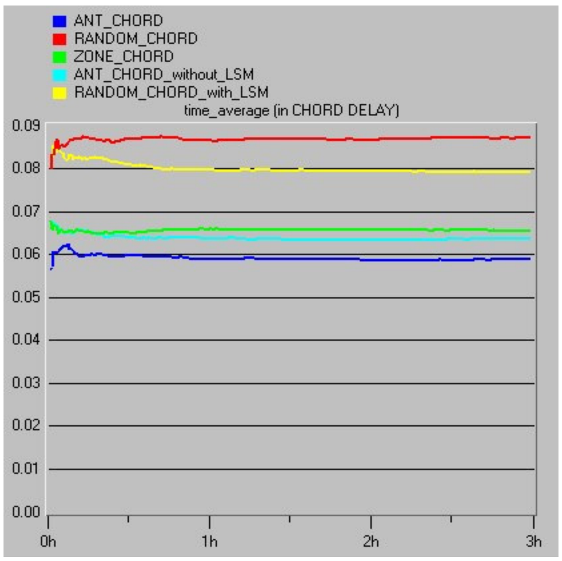

Figure 4. The average delay of 4096Chord

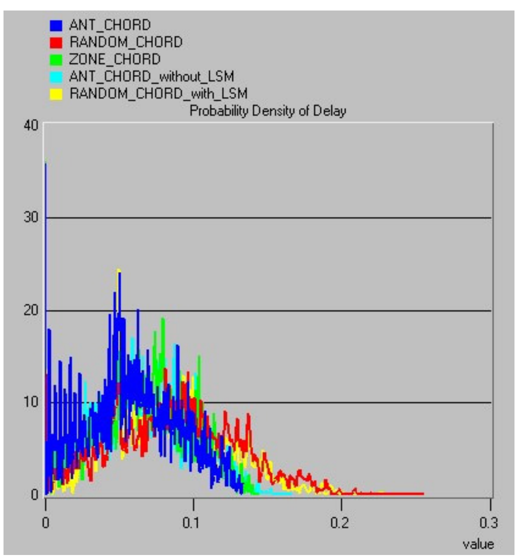

Figure 5. The probability distribution of delay of 4096Chord

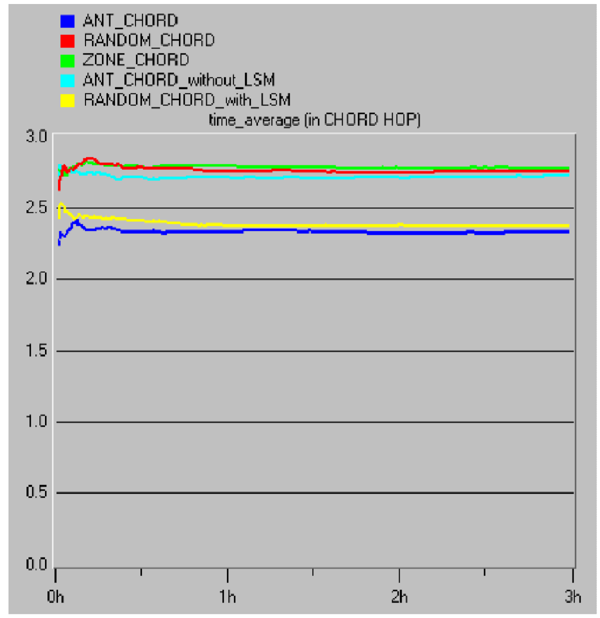

Figure 6. The average number of routing hops of 4096Chord

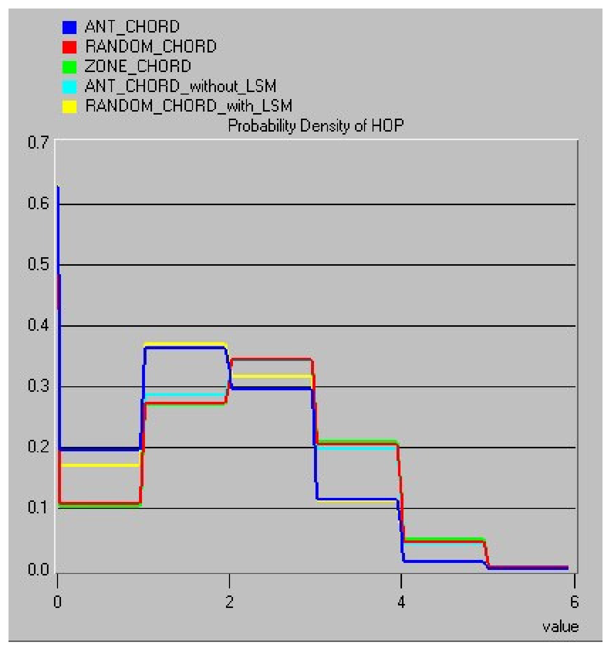

Figure 7. The probability distribution of routing hops of 4096Chord

Fig.8 to Fig.11 show the simulation results of the 16384Chord, in this simulation, we limited the number of iterations in Ant-Chord so that in each experiment AntChord only obtained an approximate optimal solution of TSP. Meanwhile, in order to facilitate comparative analysis, we also added two additional experiments, which are the

Ant-Chord without using Luoyang shovel method (denoted as ANT_CHORD_without_LSM) and the TSP_Chord used both the TSP optimal solution and Luoyang shovel method (BEST_TSP_CHORD), represented by blue and yellow lines respectively.

Apart from the similar conclusions can be drawn from 4096Chord in these figures, we can also see the performance of Ant-Chord and TSP_Chord is similar. This proved that the idea of to solve the physical topology matching problem through TSP is feasible, on the other hand proved that the approximate optimal solutions of TSP which quickly obtained by ACO algorithm, even if not the best, but because it is still using of the idea of global TSP optimization, so the effect is satisfactory. We compare Fig.4 and Fig.8 can find that with the increase of Chord, the advantages of Ant-Chord on delay become more pronounced than the dividing physical region Chord which using the idea of local optimization.

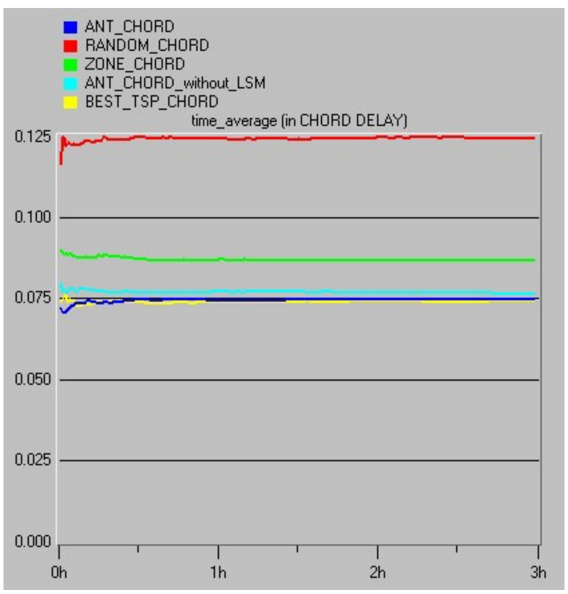

Figure 8. The average delay of 16384Chord

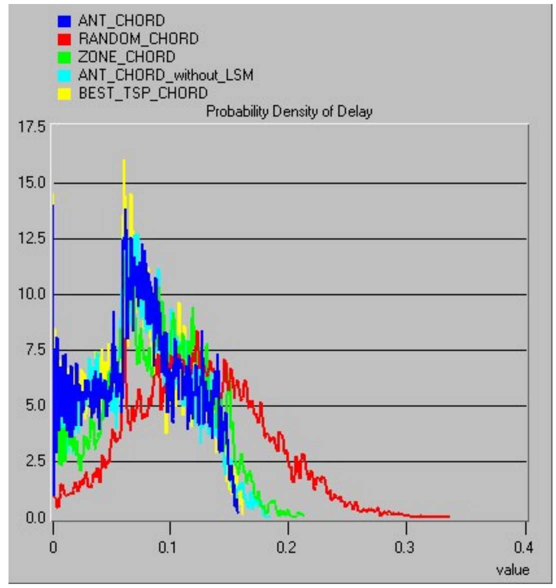

Figure 9. The probability distribution of delay of 16384Chord

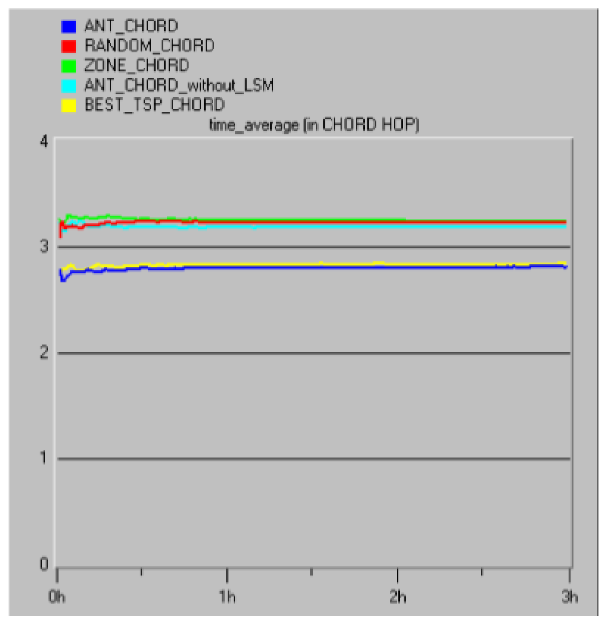

Figure.10 The average number of routing hops of 16384Chord

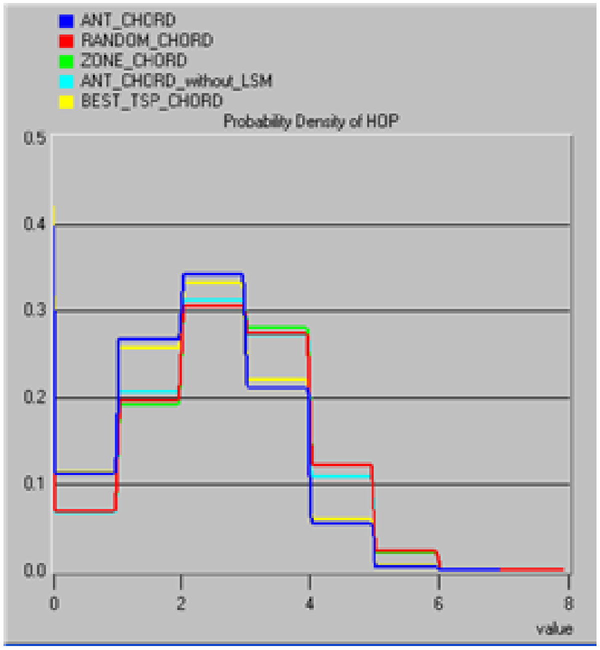

Figure 11. The probability distribution of routing hops of 16384Chord

V. CONCLUSION

This paper proposed to utilize the global TSP problem to achieve the physical topology matching in Chord, then use ACO algorithm to fast obtain the approximate optimal solution in order to meet the requirements of the network communication efficiency, and proposed the "Luoyang Shovel Method" which required small overhead but with prominent effect. Experimental results demonstrate the feasibility of the Ant-Chord model. For the construction program of new network or even the existing Chord network, and the research of the physical topology matching, Ant-Chord model may provide another idea for reference.

REFERENCES

[1] Hsien-Kai Hsin ; Grad. Inst. of Electron. Eng., Nat. Taiwan Univ., Taipei, Taiwan ; En-Jui Chang ; An-Yeu Wu, "Spatial-Temporal Enhancement of ACO-Based Selection Schemes for Adaptive Routing in Network-on-Chip Systems", Published in: Parallel and Distributed Systems, IEEE Transactions on (Volume:25 , Issue: 6) Page(s): 1626 – 1637.

[2] Caldeira, J.L. ; Tech. Univ. of Lisbon, Lisbon; Azevedo, R.C. ; Silva, C.A. ; Sousa, J.M.C., "Supply-Chain Management Using ACO and Beam-ACO Algorithms", Published in: Fuzzy Systems Conference, 2007. FUZZ-IEEE 2007. IEEE International Date of Conference: 23-26 July 2007 Page(s): 1 – 6.

[3] Huang Guangdong ; China Univ. of Geosciences, Beijing ; Ling Ping ; Wang Qun, "A Hybrid Metaheuristic ACO-GA with an Application in Sports Competition Scheduling", Published in: Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing, 2007. SNPD 2007. Eighth ACIS International Conference on (Volume:3) Date of Conference: July 30 2007-Aug. 1 2007 Page(s): 611 – 616.

[4] Jialiang Kou ; Sch. of Comput. Sci. & Technol., Wuhan Univ. of Technol., Wuhan, China ; Shengwu Xiong ; Zhixiang Fang ; Xinlu Zong, "Positive point charge potential field based ACO algorithm for multi-objective evacuation routing optimization problem", Published in: Evolutionary Computation (CEC), 2012 IEEE Congress on Date of Conference: 10-15 June 2012 Page(s): 1 – 8.

[5] Sun Xu ; Center for Earth Obs. & Digital Earth, Chinese Acad. of Sci., Beijing, China ; Zhang Bing ; Yang Lina ; Li Shanshan, "Hyperspectral image clustering using ant colony optimization(ACO) improved by K-means algorithm", Published in: Advanced Computer Theory and Engineering (ICACTE), 2010 3rd International Conference on (Volume:2) Date of Conference: 20-22 Aug. 2010 Page(s): V2-474 - V2-478.

[6] Yi Wang ; Dept. of Computer., Hong Kong Polytech. Univ., Kowloon, China ; Bathen, L.A.D. ; Zili Shao ; Dutt, N.D., "3D-FlashMap: A physical-location-aware block mapping strategy for 3D NAND flash memory", Published in: Design, Automation & Test in Europe Conference & Exhibition (DATE), 2012 Date of Conference: 12-16 March 2012 Page(s): 1307 – 1312.

[7] Chen Guojin ; Sch. of Mech. Eng., Hangzhou Dianzi Univ., Hangzhou, China ; Zhu Miaofen ; Su Shaohui ; Gong Youping, "Research on the Multi-domain Modeling and Optimizing Method for Loader Executing System", Published in: Intelligent System Design and Engineering Application (ISDEA), 2010 International Conference on (Volume:2) Date of Conference: 13-14 Oct. 2010 Page(s): 854 – 857.

[8] Noguchi, H. ; Port & Airport Res. Inst., Yokosuka, Japan ; Fujita, Y. ; Fujita, I. ; Noda, I., "Development of simple sand bypass system using a self-sinking suction pipe with holes", Published in: OCEANS '04. MTTS/IEEE TECHNO-OCEAN '04 (Volume:1) Date of Conference: 9-12 Nov. 2004 Page(s): 71 - 77 Vol.1.

[9] Saska, M. ; Julius-Maximilians Univ. of Wuerzburg, Wurzburg ; Hess, M. ; Schilling, K., "Route scheduling approach for airport snow shoveling using formations of autonomous ploughs", Published in: Control, Automation, Robotics and Vision, 2008.

ICARCV 2008. 10th International Conference on Date of Conference: 17-20 Dec. 2008 Page(s): 390 – 397.

[10] Canghui Zhu ; Minist. of Transp., Transp. Planning & Res. Inst., Beijing, China ; Jingdong Jin ; Gan Chai ; Chengwei Fang, “Coordinated ramp metering on expressway with congestion”, Published in: Transportation of China (AFTC 2010), 6th Advanced Forum on Date of Conference:16-16 Oct. 2010 Page(s): 118 – 122.

[11] Xuan Li ; State Key Lab. of Rail Traffic Control & Safety, Beijing Jiaotong Univ., Beijing, China ; Chao Shen ; Ai Bo ; Gang Zhu, “Finite-state Markov modeling of fading channels: A field measurement in high-speed railway”, Published in: Communications in China (ICCC), 2013 IEEE/CIC International Conference on Date of Conference: 12-14 Aug. 2013 Page(s): 577 – 582.

[12] Feng Dongqing ; Inst. of Inf. & Control, Zhengzhou Univ., Henan, China ; Yin Jinyan ; Chen Tiejun ; Lin Ming, “An intelligent method for high precision oil measurement control”, Published in: Intelligent Processing Systems, 1997. ICIPS '97. 1997 IEEE International Conference on (Volume:1) Date of Conference: 28-31 Oct 1997 Page(s): 666 - 669 vol.1.

[13] Zhuxin Zhang ; Jilin Univ., Changchun ; Dingxuan Zhao ; Tiehua Chen, “Design of the Master-Slave System and Research on Bilateral Servo Control System of Tele-Operated Manipulation Robots”, Published in: Automation and Logistics, 2007 IEEE International Conference on Date of Conference: 18-21 Aug. 2007 Page(s): 913 – 917.

