TSINGHUA SCIENCE AND TECHNOLOGY
ISSN 1007-0214 15/20 pp213-218
Volume 9, Number 2, April 2004

Pheromone-Based Ant Routing System for IP Networks*

ZHANG Lin (张 林)**, REN Yong (任 勇), SHAN Xiuming (山秀明)

Department of Electronic Engineering, Tsinghua University, Beijing 100084, China

Abstract: The pheromone-based ant routing algorithm is a distributed routing algorithm with good scalability and robustness. A 2-D cellular automata (CA) model of the computer network was presented to analyze the algorithm. The results show that the procedure of establishing a stable route is self-organized towards the attractive peculiar state, and the duration of time for the routing establishment is power-law distributed. A practical ant routing protocol over an IP network was also presented, and two simulations were done to compare the performance dynamic and the load balancing performance between this protocol and the open shortest path first (OSPF) protocol. The results show that the ant routing protocol out-performs OSPF in these aspects.

Key words: routing; power-law distribution; self-organization

Introduction

Internet routing is very important in the network architecture as the scale of the network increased exponentially in the past ten years. However, classical routing algorithms, such as the routing information protocol (RIP)[1] and the open shortest path first (OSPF) algorithm[2], become inadequate in scalability and robustness. In 1997, Schoonderwoerd presented the ant-based algorithm[3]. Simulating the foraging behavior of the ants, the algorithm forwards data packets according to the pheromone that routing ant packets deposit on the nodes. The algorithm shows good adaptability and robustness. Based on this work, some revised schemes were presented with better performance[4-6]. Unlike classical routing algorithms, the ant-based routing establishes routes via the interaction of software agents, which cooperate to find shortest paths in a collective manner.

However, there is no apparent analytical work presented to show the basic principles of the algorithms. In this paper, a pheromone-based routing system similar to Ref. [3] is implemented on a 2-D cellular automata (CA) network model. The simulations show that the route searching is a self-organized procedure towards the achievable criticality point.

To implement the algorithm over the predominant IP-based network, a practical protocol is also introduced in this paper, which consists of two kinds of message packets. Comparison of this protocol to the widely-used OSPF shows that the algorithm performs better in dynamics and load balancing.

1 Rationale

1.1 Algorithm

To simplify the complex architecture of networks to a representative model with an acceptable load of computation is essential. The CA is a time-discrete, space-discrete, status-discrete, and status-finite dynamic system[7], which has been used for simulating many complex systems, such as hydrodynamics, sand pile models, and high-way

Received: 2002-10-29

* Supported by the Tsinghua "985" Innovation Foundation, the National Natural Science Foundation of China(Nos. 60202010 and 90204004), and the National High-Tech Research and Development (863) Program of China (No. 2001AA121062)

** To whom correspondence should be addressed.
E-mail: linzhang@tsinghua.edu.cn;
Tel: 86-10-62788878

traffic[8], because the CA provides both global information of the system and quantitative results in some cases.

To analyze the algorithm, we simplify the network as a 2-D CA model, which is an isotropic grid. Each node in the grid represents a router with four nodes above, below, to its left and right. Two nodes A and B are arbitrarily selected from the grid. Routing simulations are conducted between them. The algorithm in Ref. [3] is revised in the CA model as follows:

1) Each node i has two pheromone parameters: $H_i^{A\to B}$ and $H_i^{B\to A}$.

2) Node A releases ant routing packets to node B at each clock click. These ants are called forward ants. These packets are forwarded by the nodes to their neighbor nodes on each clock click with the forward transfer probability $p_{i\to j}^{\mathrm{f}}$ (i and j are neighboring):

$$p_{i\to j}^{\mathrm{f}} = \frac{H_j^{B\to A}}{\sum\limits_{k\in \mathrm{N}\ (i)} H_k^{B\to A}} \tag{1}$$

where $\mathrm{N}\ (i)$ is the set of nodes neighboring node i.

3) On reaching target B, the ants change the target to A and move back in the same manner. These ants are called backward ants. The backward transfer probability on each node is determined by $H_i^{A\to B}$, which is

$$p_{i\to j}^{\mathrm{b}} = \frac{H_j^{A\to B}}{\sum\limits_{k\in \mathrm{N}\ (i)} H_k^{A\to B}} \tag{2}$$

4) Each forward or backward ant has an age parameter, counting the hops that the ant has passed since it was last released by node A or B. When the age of an ant exceeds its maximum value $\mathrm{AGE}_{\max}$, the ant is destroyed and a new ant will be launched by node A or B depending on who generated it.

5) The nodes A and B guarantee that the maximum number of ants in the grid is N, i.e., if the number of ants in the network equals N, they will cease releasing new ants until an ant gets to its destination or is destroyed due to its expiration.

6) The pheromone parameters, $H_i^{A\to B}$ and $H_i^{B\to A}$, are updated by each forward or backward ant, respectively. The ants deposit pheromone on the nodes they pass through. The increment is

$$\Delta H = \lambda^{\mathrm{AGE}} \tag{3}$$

$\lambda \in (0,1)$ is a constant.

7) At each time click, the pheromone parameter of each node is attenuated by multiplying a proportion parameter D ($0 < D < 1$) which is called vaporizing rate.

The above CA model summarizes the basic principles for pheromone-based routing, which are all local without any need for global information of the network. If D, λ, and N are properly selected, a shortest route between nodes A and B will emerge after iteration (as shown in Fig. 1 and Fig. 2).

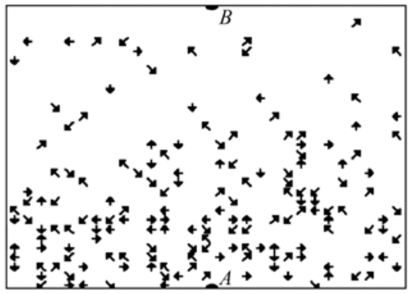

Fig. 1 Agents randomly move at the beginning of the simulation

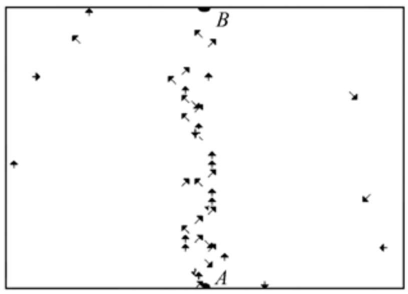

Fig. 2 Agents are attracted to the stable route established by the self-organization procedure

1.2 2-D CA simulation

The vaporizing-rate D reflects the system's remembrance of the history status. When D = 1, all the history status have same power; when D < 1, the system maintains an exponentially decayed memory; if D is too small, the cooperation among ant packets is so weak that it will take quite a long time to establish routes; If D = 0, there is no cooperation among packets and no route can be established, and thus all the packets will be randomly roaming. The closer to 1 D is, the stronger the influence of the history status. Hence, D is a parameter that describes the global characteristics of the system. D determines the evolution pace of the system. When D is too small, the system is chaotic; when D is too big, the system lacks agility to system state variations. When D is properly set between 0.5 and 0.8, the system shows satisfying robustness and adaptability.

λ reflects the system's preference to a shorter route.

The bigger λ is, the stronger the influence of the younger ants compared to the elder packets. Hence, the system is liable to establish a shorter route, and the final emerging route is the shortest route. With the same rule, different pheromone curves can be applied to gain different routes. If ΔH is determined on bandwidth or delay, the maximum bandwidth or minimum delay route will be established. This is helpful to implement quality of service (QoS) routing.

The number of ants, N is an important parameter of the system. When the number of software agents participating in the route searching is small, it will take quite a long time to establish a stable route. However, if it is too large, the payload will be heavy and the transition efficiency will be low.

The simulation is conducted in a C++ coded simulator with the size of grid being 30×30. Set AGE_{max} = 200, D=0.6, and λ = 0.2. After about 50 iterations, almost all the ants are attracted to the shortest path between A and B. Because of the probability the ants will look forward, the route tends to change initially, but becomes stable later. Since the performance of the predominant TCP protocol will be severely deteriorated due to frequent disorder of data packets, which is probably caused by alteration within routes, it is a requirement for route algorithms to present routes with longer lifetimes. Taking this criterion into account, we claim in the simulation that if a route remains unchanged for 15 clock clicks, it is recorded as a stable route.

1.3 Temporal-spatial correlation in the algorithm

To analyze the relationship between the duration time for establishing stable routes and the number of ants, we arrange the following simulation. For each integer $N \in [5,100]$, we conducted 100 simulations, establishing a stable route each time, the duration of which is recorded. The results show that the smaller the number is, the wider the duration of time is distributed, and the higher the statistical average is. In contrast, the higher the number is, the more convergent the duration of time is, and the lower the statistical average is (as shown in Fig. 3).

The average duration for establishing a stable route is shown in Fig. 4. Its slope shows good evidence of the power law distribution. Using fast Fourier analysis on the distribution, the power spectrum in Fig. 5 has the shape of $1/f^{\alpha}$ noise with a slope of about –1.44.

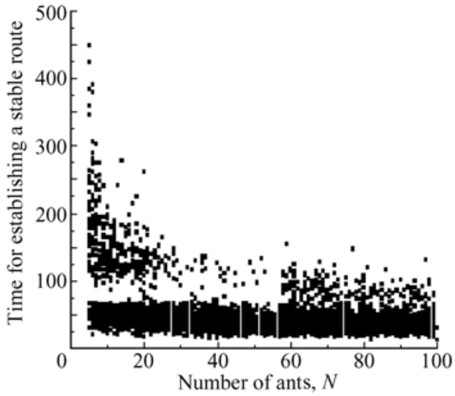

Fig. 3 Distribution of duration time for establishing a stable route

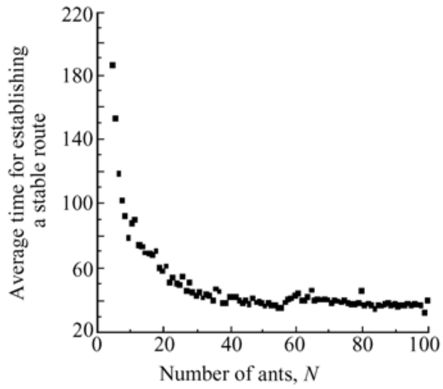

Fig. 4 Average time for establishing a stable route

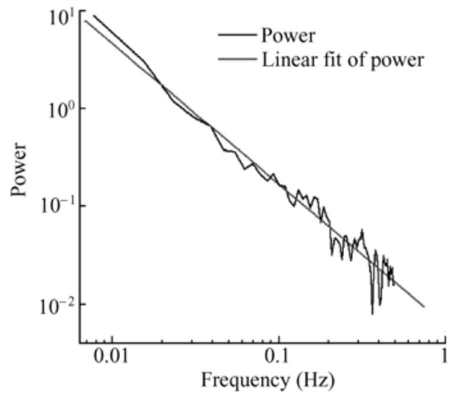

Fig. 5 Power spectrum of establishing time of stable routes

The algorithm discovers the shortest route in a self-organized procedure towards the achievable criticality point. At that point, the pheromone is distributed along the shortest route. The simulation results show that the power spectrum of the algorithm's practical convergence time is power-law distributed, which is good evidence for long-range temporal correlation. The intensity of the correlations is determined by parameter D. It is also highly reasonable to conclude that there are long-range spatial correlations between the agents from the simulation results. Although the rules for the software agents are spatial short-range, the global information of the system can be shared among distant agents. Therefore, a global solution can be obtained by a set of local rules.

2 Implementing Algorithm over IP Network

To implement the algorithm over IP networks, routers must cooperate by responding to the routing packets from other routers and by forwarding the data packets according to the local pheromone table. Routers should also be aware of the latency on the links connected to it so that they can calculate the age of the ant packets.

We introduce the following two kinds of routing messages into IP-based network to implement the pheromone-based routing.

2.1 Ant packets (APs)

Ant packets are responsible for searching the best routes in a cooperative manner, which is similar to the algorithm introduced in Section 1.1. The rules are described as follows:

1) Every router launchs AP with a random destination. The AP carries its source node ID, destination node ID and lifetime since launched, denoted as $R(s, d, t)$. The structure of AP is shown in Fig. 6.

2) There is a forward probability table $\{p_{ij}\}$ $(i < M, j < L)$ on each router. M is the total number of routers in the network and L is the number of links connected to the router. p_{ij} denotes the forwarding probability to destination node i from outgoing link j. AP is forwarded to the next hop according to p_{ij}.

3) At initialization, $p_{i1} = p_{i2} = \cdots = p_{iN} = 1/L$, i.e., the APs are forwarded without any preference to any special neighbor. As AP passes through a router, the forward probability table $\{p_{ij}\}$ will be updated. If $R(s, d, t)$ comes from link l, the s-th line of $\{p_{ij}\}$ is changed to:

$$p_{sj} = \frac{p_{sj} + \Delta p}{1 + \Delta p}, \quad j = i;$$

$$p_{sj} = \frac{p_{sj}}{1 + \Delta p}, \quad j \neq i;$$

$$\Delta p = \frac{C_1}{t} + C_2 .$$

C_1 and C_2 are system-defined constants.

4) Upon reaching its destination, the AP is destroyed. If the AP roams around the network without reaching the destination after a given period of time, it is also destroyed.

5) Unlike the APs, the real data traffic packets are forwarded by the maximum probability rule. The data packets always choose the outgoing link with highest probability to guarantee a stable route for data flow.

Different from the simulation case in the 2-D CA module, the time in the realistic networks is continuous. Consequently, the age of the ants are recorded in a 32-bit unsigned double-word in the unit of ms.

<table>
<tr><td>Bits 0-7</td><td>Bits 8-15</td><td>Bits 16-31</td></tr>
<tr><td>Operation code</td><td>Version</td><td>QoS reservation</td></tr>
<tr><td colspan="3">Source IP address</td></tr>
<tr><td colspan="3">Destination IP address</td></tr>
<tr><td colspan="3">Life time</td></tr>
</table>

Fig. 6 Format of AP

2.2 Latency measurement packets (LMPs)

The latency measurement packet is actually a derivative in the data-link layer, which is used to detect the latency on the link. Because there is no synchronous clock across the network, the lifetime field of the ant packets must be accumulated in a "hop-by-hop" manner. Every router in the network sends the LMP periodically to its neighbor routers who send back LMP along the same link right after they receive it. The structure of LMP is shown in Fig. 7. Both the forward LMP and backward LMP are put in the queue of the link so that the round trip time of an LMP over a link is composed of three parts: forward queuing time, backward queuing time, and router action time. The router action time is the time for the receiving router to examine an LMP and put it in the backward queue, which is insignificant compared to the first two parts, since there is no need for the router to query the routing table to respond to an LMP. On receiving an LMP, the sender will estimate the link latency by dividing the round trip time by two.

Bits 0-7	Bits 8-15	Bits 16-31
Operation code	Version	Sending time (Sender's local time)

Fig. 7 Format of LMP

3 Simulation

We established a continuous time network simulator with C++. The simulation was done over a topology as shown in Fig. 8, which has been used in Ref. [4].

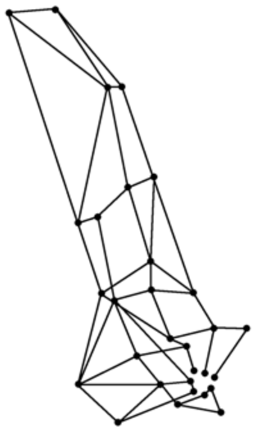

Fig. 8 Topology of the network

3.1 Dynamic performance

At the beginning of the simulation, 5 s are given to the routing algorithms to accumulate the topology information of the network. After 5 s, data packets of 800 Bytes long are injected into the network from a randomly chosen node. From the viewpoint of a special node, the arrival of data packet is Poisson distributed with an average interval of 0.5 ms. After 6 s, four links fail so that node *A* connects to the network by only one surviving link. We conducted this scenario with the two algorithms to compare their dynamic performance.

The result (shown in Fig. 9) shows that under the two algorithms, data packet loss appears after the link failure, while the ant-based routing recovers from failure more quickly than OSPF. The periodical and event-trigger refreshing mechanism of OSPF propagates the failure notifications throughout the network explicitly in a coarse-grained time scale, while ant-based routing propagates the failure notification in fine-grained time scale via co-operation among ant packets, which leads to a faster response to the change of the network topology. The results indicate that the fine-grained ant-based routing could present a softer fine-tuning method to the routing table and, as a result, a less oscillatory routing strategy than OSPF.

3.2 Load balancing performance

Load balancing is a desirable characteristic for a routing algorithm, especially in cases of heavy loads, in which a

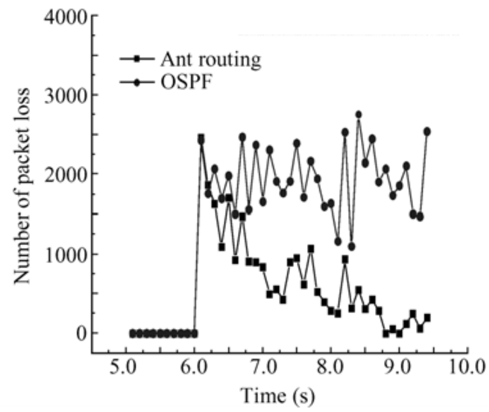

Fig. 9 Dyamic performance

routing algorithm with load-balancing property could distribute the load to an alternative link when a specific bottle-neck link appears, preventing deterioration of service and promoting the overall efficiency of the network resource.

We conducted the following scenario to compare the load-balancing performance of the two algorithms. Similar to the case in Section 3.1, the first 5 s of the simulation is the learning procedure for the two algorithms, after which the data packets are injected into the network with a length of 800 Bytes and average interval of 0.5 ms in Poisson arrival. From the 6th second, the injecting flow increased as much as 4 times, i.e., the packet size is increased to 3200 Bytes. We can see from the results shown in Fig. 10 that OSPF routing has severe data packet losses, while the ant routing loss is slight. Typical OSPF is unaware of the link loads, and it calculates the best routes in a static metric measurement. However, ant routing takes the dynamic load distribution into account, making corresponding adjustments to the change of load, which impart good load balance performance.

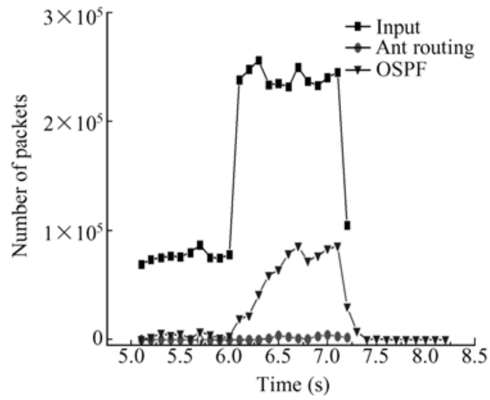

Fig. 10 Load balancing performance

4 Conclusions

In this paper, the cellular automata model is used to analyze the ant-based routing algorithm. According to the simulation results, the system can evolve towards a peculiar state with power-law scaling noise. The system is temporal correlated and probably has long-range spatial correlation. Therefore, the global optimal routes can emerge by local rules.

We also present an implementation protocol over an IP network, which uses the basic idea of pheromone-based ant routing. Simulation results show that this routing algorithm gains better dynamic and load balancing performance than the predominant OSPF routing protocol. The pheromone-based ant algorithms might indicate a promising method to implement flow and congestion control in a context of traffic engineering with a cross-layer optimization.

Acknowledgements

The authors would like to thank Adam Geisler for his enthusiastic help in revising the draft.

References

[1] Malkin G. RIP version 2. RFC 1388, http://www.ietf.org/rfc/rfc 1388.txt, 1993.

[2] Moy J. OSPF version 2. RFC 1583, http://www.ietf.org/rfc 1583.txt, 1994.

[3] Schoonderwoerd R. Ant-based load balancing in telecommunications networks. *Adaptive Behavior*, 1996, **5**(2): 169-207.

[4] Di Caro G, Dorigo M. AntNet: Distributed stigmergetic control for communications networks. *Journal of Artificial Intelligence Research,* 1998, **9**: 317-365.

[5] Di Caro G. Two ant colony algorithms for best-effort routing in datagram networks. In: Proceeding of the 10th IASTED International Conference on Parallel and Distributed Computing and Systems. Las Vegas, Nevada, 1998.

[6] Subramanian D. Ants and reinforcement learning: A case study in routing in dynamic networks. http://cs-tr.cs.rice.edu/Dienst/UI/2.0/Describe/ncstrl.rice_cs/TR96-259, 1998.

[7] Wolfram S. Theory and Applications of Cellular Automata. World Scientific, 1986.

[8] Nagel K, Schreckenberg M. A cellular automaton model for freeway traffic. *Journal de Physique I*, 1992, **2**(12): 2221-2229.