AntNet and Relative Pheromone Evaporation

Shigeo DOI and Masayuki YAMAMURA
Interdisciplinary Graduate School of Science and Engineering, Tokyo Institute of Technology
4259 Nagatsuta-cho, Midori-ku, Yokohama-shi, Kanagawa, 226-8502 JAPAN
doi@es.dis.titech.ac.jp, my@dis.titech.ac.jp

Abstract— **AntNet-FA showed high potential for best-effort routing through some experiments. However this algorithm was pointed out that it can't adapt to a situation of traffic change such as congestion. A pheromone evaporation algorithm, DCY-AntNet, was proposed but the algorithm evaporates pheromone simply with a constant rate. We propose an alternative algorithm using relative pheromone evaporation. Our algorithm takes into consideration of the traffic situation through local link queues at each node. That is, if there is no need to evaporate then no evaporation is occurred. This feature can make good use of agents' experiences. We ran some simulations and got better results than competitors for a sparse network.**

I. INTRODUCTION

As a routing algorithm for a large distributed network environment such as the Internet, OSPF [1], Q-Routing [2] and PQ-Routing [3] have been proposed. Q-Routing is based on Q-learning and PQ-Routing is an improvement algorithm on Q-Routing, respectively.

In 1997 Gianni Di Caro and Marco Dorigo have proposed AntNet algorithm [4], which is based on Ant System [5]. Ant System is based on the behavior of ants in nature. AntNet shows better performance than these algorithms through some experiments. Therefore, AntNet is a candidate for best-effort routing, but this algorithm is not robust after traffic condition is changed. Multi-agent based algorithms, like AntNet, are effective for widely distributed network routing algorithm since no one can view the whole network condition. For example, traffic condition and network topology changes dynamically. At the point, a network routing problem is like a dynamic TSP problem [6].

AntNet was reported an undesirable effect "Routing Lock" [7] , which means that once probability at each routing table entry selecting a neighbor node converges to nearby 1, the probability is not easily reduced if the link is congested. To avoid the side effect, the decay method (DCY-AntNet) was proposed. DCY-AntNet is an improvement algorithm of AntNet [7].

Although DCY-AntNet evaporates pheromone at regular interval without a consideration of current traffic status, undesirable effects are occurred by the algorithm. For example, it sometimes evaporates pheromone whenever there is no need to evaporate. Therefore, in this paper we propose an algorithm to evaporate pheromone with taking into consideration of current status of link queue at each node. Our proposal reduces the defect of DCY-AntNet. We also take into consideration of the constraint that an agent must not move to the nodes where it has already visited.

The organization in this paper is as follows: In section 2 we show the definition of network routing problem and the abstract of AntNet algorithm. In section 3 we show a relative pheromone evaporation algorithm. In section 4 we show some simulation results among six algorithm. In section 5 we conclude the experiment in this paper and describe the future work.

II. RELATED WORKS

AntNet algorithms are proposed in [8] [9] [4] . There are several variations of this algorithm, our algorithm is based on AntNet-FA[10], the latest AntNet algorithm. In this section, we explain AntNet-FA in summary.

A network is given as a undirected graph $G = (V, E)$. V is a set of nodes, which corresponds to routers. E is a set of links. Let us assume a datagram packet at a node s. Its destination node is d. Each datagram packet at a node s is routed with probability $P^s(n, d)$ by the following equation:

$$\sum_{n \in N(s)} P^s(n,d) = 1, ^\forall s \in N, ^\forall d \in N - \{s\} \tag{1}$$

where $N(s)$ is a set of neighbors of s .

A. AntNet

Hereby we summarize AntNet algorithm below:

1) Forward Ant:

1) In every node, a forward ant is launched at regular interval. The node decides the ant's destination in proportional to the destination of datagram packets.
2) The ant decides a link to go its destination. It uses the following equation to select the adjacent node of the current node with probability $P'^k(n, d)$.

$$\begin{cases} P'^k(n,d) & = \frac{P^k(n,d)+\alpha q_{k\to n}}{1+\alpha(|N(k)|-1)} \\ q_{k\to n} & = 1 - \frac{Q_{k\to n}}{\sum_{t \in N(k)} Q_{k\to t}} \end{cases} \tag{2}$$

where $Q_{k\to n}$ is the queue length(bit) from node k to node n and α is a heuristic parameter. In this paper, we used $\alpha = 0.5$ used in [4].

3) The ant computes the virtual trip time based on the property of the link such as bandwidth and propagation delay. Then, it pushes the virtual trip time into its stack.
4) The ant is sent to an adjacent node corresponding to the link queue. The ant is queued with high priority in a link queue of the node.

5) When the ant arrives at a node, repeat these 2 steps above until it reaches the destination. If a loop is detected during its trip, then the ant is killed in case that the loop length is greater than the half of total length, otherwise, the loop is removed and the trip continues.

2) Backward Ant:

1) If a forward ant reaches its destination, then the forward ant turns into "backward ant" and goes back to the forward ant's source node to reflect the traffic statistics. Therefore the forward ant's stack is inherited. The backward ant also has high priority in a link queue.
2) During on the backward ant's way, when the backward ant arrives at a node, then update the node's routing tables for all intermediate nodes from the current node to the forward ant's destination node using the ant's stack. The updates reinforces to use the ant's way to go the ant's destination. The reinforcement value is calculated by the ant's trip time. The time is shorter and shorter, The reinforcement value is higher and higher.

B. DCY-AntNet

DCY-AntNet is proposed in [7]. All the nodes in N calculates the following equation at regular interval $\Delta = \frac{\Delta t}{\tau}$ and updates routing tables. v is the decay parameter. In this paper, we set the parameter $v = 0.00015$ used in [7]. This algorithm runs independently.

$$P^s(n,d) \leftarrow (1-v)P^s(n,d) + \frac{v}{|N(s)|} \tag{3}$$

III. RELATIVE PHEROMONE EVAPORATION ALGORITHM

Whenever DCY-AntNet runs pheromone evaporation, it doesn't consider the current traffic situation. Hence ants' experiences come to nothing while traffic condition is stable. Moreover when traffic is congested, each node can't detect the congestion whether it is occurred over the network or not. Then an ant's travel time increases, the ant doesn't give reward to the intermediate nodes on its way because mainly the reward is based on the best of ants' travel time. However the evaporation algorithm runs concurrently, thus probability of routing tables in a node for each destination converges to uniform, $1/d$, where d is the degree of the node. This means DCY-AntNet is not robust for the increase of traffic density on a whole network.

So we propose improved algorithms, MDCY-AntNet and MDCY-AntNetL. The algorithms eliminate these defects of DCY-AntNet. Moreover, Also we add a feature to MDCY-AntNetL. The feature is that any ant can't visit its already visited node to avoid making a loop. We describe the algorithm in detail.

A. Algorithm Details

Let us assume that a node s has a set of links L and $N(s)$ is a set of the s's neighbors. q_k where k is a neighbor of s is the queue length of $s \rightarrow k$. The procedure of the relative pheromone evaporation is as follows:

1) The node runs a evaporation algorithm for each destination. The regular interval of the evaporation γ is same to that of DCY-AntNet:

$$P^s(n,d) \leftarrow P^s(n,d) + \gamma\frac{Q - q_n}{Q} \tag{4}$$

$$Q = \frac{1}{|N(s)|}\sum_{k\in N(s)} q_k \tag{5}$$

If $P^s(n,d) < 0$ is held, then substitute 0 for $P^s(n,d)$.

2) Normalize $P^s(n,d)$ to satisfy $\sum_{n\in N(s)} P^s(n,d) = 1$ for ${}^{\forall}d$.

We implement this procedure in MDCY-AntNet and MDCY-AntNetL. In addition, we implement a loop-free feature in MDCY-AntNetL. The loop-free feature is described below:

1) Let us assume an ant A. The source node of A is s and the destination node of A is d. Also A stays at a node k.
2) A set Av of available node of the ant A is defined as $N(k) \cap (N - V)$, where V is a set of the already visited nodes of A.
3) If Av is an empty set, do the following procedure:
    a) A is forced to go back its source s, as if A reached the destination d.
    b) Moreover, the node k launches a new forward ant. Its source is k and Its destination is d.
4) If Av isn't an empty set, do the following procedure:
    a) Calculate the equation:

$$sum = \sum_{j\in Av} P^k(n,d) \tag{6}$$

    b) If $sum = 0$, then A randomly selects a neighbor node from Av.
    c) If $sum > 0$, then calculate the following equation for $\forall n \in N(k)$:

$$P''^k(n,d) = \begin{cases} \frac{P^k(n,d)}{sum} & for\ n \in Av \\ 0 & otherwise \end{cases} \tag{7}$$

    d) Substitute $P''^k(n,d)$ for $P^k(n,d)$ and Av to $N(k)$ in equation (2), then route A using the equation (2).

B. Merits of our method

Now we describe the merits of our algorithms compared with DCY-AntNet.

DCY-AntNet evaporates pheromone to converge the entries of routing tables to $1/d$ where d is the number of neighbors. Thus, although probability of an entry of routing table is less than $1/d$ and the queue length is the longest, DCY-AntNet increases the probability unfortunately. This makes transfer time of ants and datagram packets longer and longer.

Figure 1 illustrates an example that DCY-AntNet misleads to increase probability of the entry where the queue length is the longest. This phenomenon violates the capability that a forward ant in AntNet can avoid the heavy link. If the length of a link queue is longer than the others, our algorithms decrease the probability so that ants and datagram packets aren't to

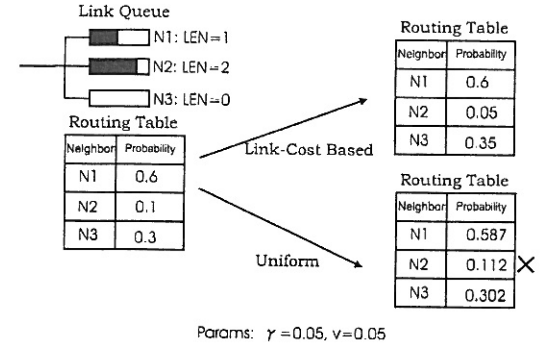

Fig. 1. Comparison between link-queue based evaporation and uniform evaporation

select the neighbor. In addition, our algorithms don't modify routing tables if any link queue is empty. Thus, our algorithms can exploit the route where forward ants experienced.

Next, the reinforcement value in AntNet is calculated by the forward ants' experiences. The value is smaller and smaller if the experienced time is longer and longer. the value is small while the experienced time by ants still remains, after network traffic density increases. On the other hand, DCY-AntNet runs concurrently and probabilities of routing tables converges gradually to $1/d$. This misleads nodes about delivering ants and datagram packets and nodes route the ants and the packets randomly. As a result, throughput of the whole network can decrease due to the emergence of loop packets.

The evaporation method of our algorithms is based on the relative load of link queues. Therefore, our algorithms preserve the route where forward ants found if the load of links are uniformly distributed. This means our algorithms are more robust than DCY-AntNet when traffic density of whole network increases. Figure 2 illustrates the behavior.

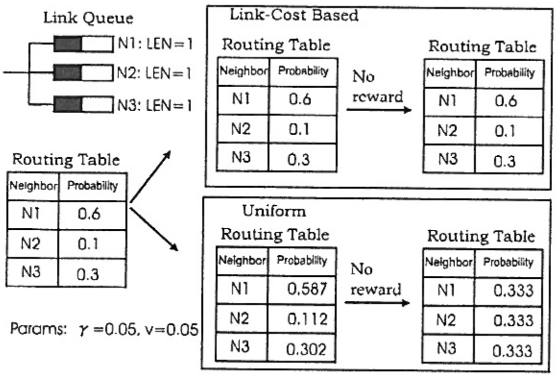

Fig. 2. Comparison of algorithm behaviors while no reward is given to nodes.

Last, this evaporation method is based on the local information such as link queues. Thus this method is easy to adapt the widely distributed systems such as the Internet.

By the way, we consider the demerits of our method. First, it is considered that the Loop-Free method above may generate many ants than AntNet-FA. However, we consider the routing algorithm can generate ants as possible, as far as ants doesn't disturb the traffic of datagram packets.

TABLE I
ALGORITHM FEATURES OF COMPETITORS

Algorithm	Relative Pheromone Evaporation	Uniform Pheromone Evaporation	Loop-Free
MDCY-AntNetL	○	×	○
MDCY-AntNet	○	×	×
AntNet-FA	×	×	×
AntNet-FAL	×	×	○
DCY-AntNet	×	○	×
DCY-AntNetL	×	○	○

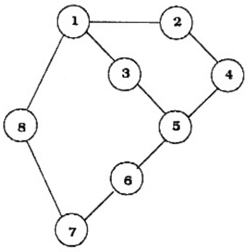

Fig. 3. SimpleNet Topology. propagation delay of all the links is 1msec and bandwidth of all the links is 10Mbps.

Second, the Loop-Free feature above can affect the routing table. That is, it may oscillate routing table, then there are several paths for a source node to a destination node when routing a datagram packet.

IV. EXPERIMENTS

A. Purposes

We select six competitors, MDCY-AntNetL, MDCY-AntNet, AntNet-FAL, AntNet-FA, DCY-AntNet and DCY-AntNet. The features of these algorithms are shown in Table I. The reason why we choose these algorithms is because these algorithms cover all the possible combination of features.

B. Environment

We ran some simulations on one network, SimpleNet. Figure 3 [4]. Each link was full-duplex and its bandwidth was 10M bit/s. This bandwidth is a little bit slow with contrast to the current Internet infrastructure, we consider that one of the requirement of such algorithms must be adaptive for a poor environment.

We used some simulation environments and parameters shown in Figure 3 and 4, and Table II. The packet length is subject to the negative exponential distribution, the arrival rate of a packet is subject to Poissonian distribution, the destination of packets is subject is subject to uniform distribution, respectively [7].

A node structure we used is shown in Figure 4 [7]. Packets generated at a node and transferred to its neighbor node was stored on its input buffer. In input buffer, a packet was processed with FIFO policy. If the destination of a packet dequeued from input buffer of the node was equal to the node, then the packet was processed locally. Otherwise, this packet was routed with its routing table and was determined

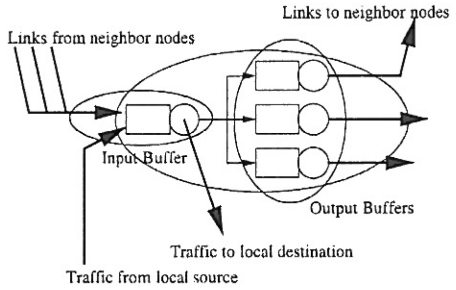

Fig. 4. Node structure.

TABLE II
COMMON SIMULATION PARAMETERS.

Topology	SimpleNet
Regular interval of launching one forward ant	300msec
Average length of each packet	512Byte
Bandwidth of each link	1.5Mbit/sec
Each input buffer size	∞
Each output buffer size	1Gbit
Time to live for each datagram packet	15sec
Maximum length of each packet	1500Bytes
Unit time of each simulation	0.1msec
Initial probability of each routing table	uniform
Total time for each simulation	400sec
Learning time before stating each simulation	500sec
Processing time for each node	3msec
Size of each forward ant	24 + 8 × (hops)
Size of each backward ant	24 + 8 × (hops)

which node is selected to send. After this determination, it was queued the output buffer corresponding to the selected node and was processed with FIFO policy except for backward ants, which have high priority than the others.

C. Criteria

We used a criteria for comparisons among algorithms as follows: 1) throughput of the whole network through one simulation [4] in bit per second, 2) distribution of transfer time of the whole packets through one simulation [4], 3) mean, variance and the maximum about packet transfer time through all simulations in msec, 4) loss, the percentage of packets not being sent within Time To Live(TTL) through all simulations.

In case congestion occurs on network, some packets will loop, which causes throughput degradation and degradation of distribution of packet transfer time. Therefore, as the point supporting uniform service to all network nodes, we measured these items about the whole network. As throughput stability is shown faster after congestion, the looped packets generation are restrained, so the algorithm is more adaptive. Also it is considered the maximum transfer time and the mean and the variance about packet transfer time will be larger if the throughput of whole network is lower. In the following experiment, each experiment for each algorithm was simulated in ten times.

We also calculate the entropy H . H indicates the randomness of routing table. Once traffic condition is changed locally, ants should increase the entropy to organize the routing tables.

$$H = \sum_{v \in N} \sum_{d \in N - \{v\}} \sum_{n \in N(v)} P^s(n,d) \log_2 \frac{1}{P^s(n,d)}$$

D. Congestion between adjacent two nodes

In this experiment we compared with the six algorithms in Table I on the situation occurring congestion between adjacent two nodes. The traffic pattern, HotSpot2, used in our simulations is as follows:

1) During each simulation, Node 1 launched a 512bytes datagram packet at 0.5msec interval to node 6, Node 6 launched a 512bytes datagram packet at 0.5msec interval to node 1 too.
2) From 50sec to 150sec, Node 8 launched a 512bytes datagram packet at 0.3msec interval to node 7, Node 7 launched a 512bytes datagram packet at 0.3msec interval to node 8 too.
3) From 150sec to 250sec, Node 3 launched a 512bytes datagram packet at 0.3msec interval to node 5, Node 5 launched a 512bytes datagram packet at 0.3msec interval to node 3 too.
4) From 250sec to 350sec, Node 2 launched a 512bytes datagram packet at 0.3msec interval to node 4, Node 4 launched a 512bytes datagram packet at 0.3msec interval to node 2 too.

We simulated the traffic pattern above in ten times. We obtained the following results: Figure 5 illustrates the transfer time distribution of datagram packets. Figure 6 illustrates the throughput dynamics of datagram packets. In the Figure 6, We indicated the unit "1Mbps" is equal to 10^6 bps. Figure 7 illustrates the entropy dynamics of routing tables.

TABLE III
THE STATISTICS WITH HOTSPOT2 TRAFFIC ON SIMPLENET.

Algorithm	Average	Variance	Maximum Transfer Time	Loss Rate
MDCY-AntNetL	2604.7	1.18×10^7	15000	0.4
MDCY-AntNet	4736.3	3.35×10^7	15000	4.7
DCY-AntNetL	3121.1	1.79×10^7	15000	0.1
DCY-AntNet	4174.4	3.22×10^7	15000	2.7
AntNet-FAL	3598.5	2.39×10^7	15000	0.4
AntNet-FA	6314.3	4.16×10^7	15000	15.1

Figure 5 illustrates that MDCY-AntNetL is the best algorithm. For example, the transfer time of 90% of all the launched datagram packets is about 7500msec for MDCY-AntNetL, about 8000msec for DCY-AntNetL, about 10000msec for AntNet-FAL.

The stability of throughput dynamics from Figure 6, MDCY-AntNetL is stable before 75sec but AntNet-FAL and DCY-AntNetL is stable after 75sec. This indicates MDCY-AntNetL is effective to distribute the load of links once traffic jam is occurred. In the traffic pattern, the throughput of generated datagram packets from node 8 to node 7 is 512Bytes/0.3msec = 13.6×10^6 bit/sec. This is more than the bandwidth 10Mbps from node 8 to node 7. Therefore, routing tables should be changed to send datagram packets faster in this situation.

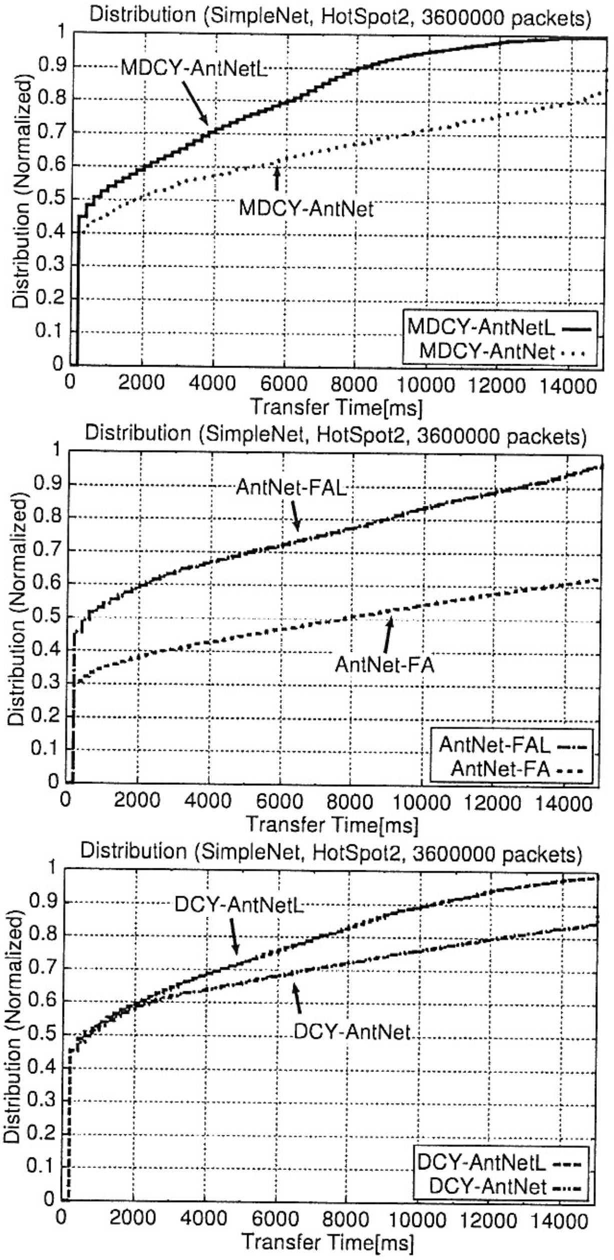

Fig. 5. Comparison of Transfer Time Distribution of Datagram Packets

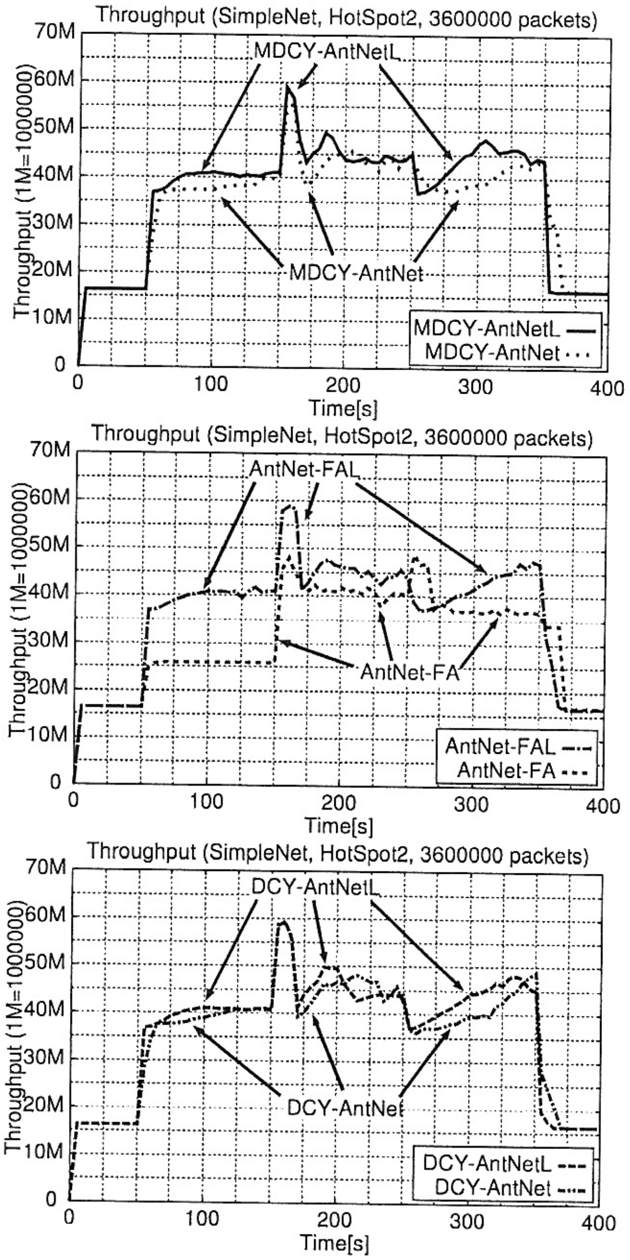

Fig. 6. Comparison of Throughput Dynamics of Datagram Packets

When a node uses only one link to route datagram packets for a destination, the entropy is equal to 0 for the destination. If a node can't afford to send datagram packets using only one link, a node must increase the entropy to distribute datagram packets.

In AntNet-FA once traffic condition is changed but nodes in SimpleNet didn't increase the entropy, especially after 50sec. This means the reactiveness of AntNet-FA is less than the others. This phenomenon is observed from the view of throughput dynamics.

SimpleNet is the network that all the node except for Node 1 and Node 5 has only two links, thus ants can easily make a loop by the algorithms without Loop-Free feature. Imagine an ant is launched from node 7 to node 8 while the link from node 7 to node 8 is congested (i.e. from 50sec to 150sec). Suppose the ant avoided using the link and reached node 6. However, the routing table for node 8 at node 6 can indicate to select node 7 because the distance is the smallest to select node 7 as a next node. Thus, the ant can go to node 7 and make a loop, then it dies. This means the transfer interval about traffic statistics and and the update interval of routing table by ants is less than Loop-Free algorithms. Consequently node 8 and node 7 can't detect that traffic condition is changed and use only one link, algorithms without Loop-Free feature can't increase throughput of datagram packets.

Algorithms without pheromone evaporation can send datagram packets before traffic condition is changed (i.e. 50sec). After traffic congestion is occurred, the distance of alternative

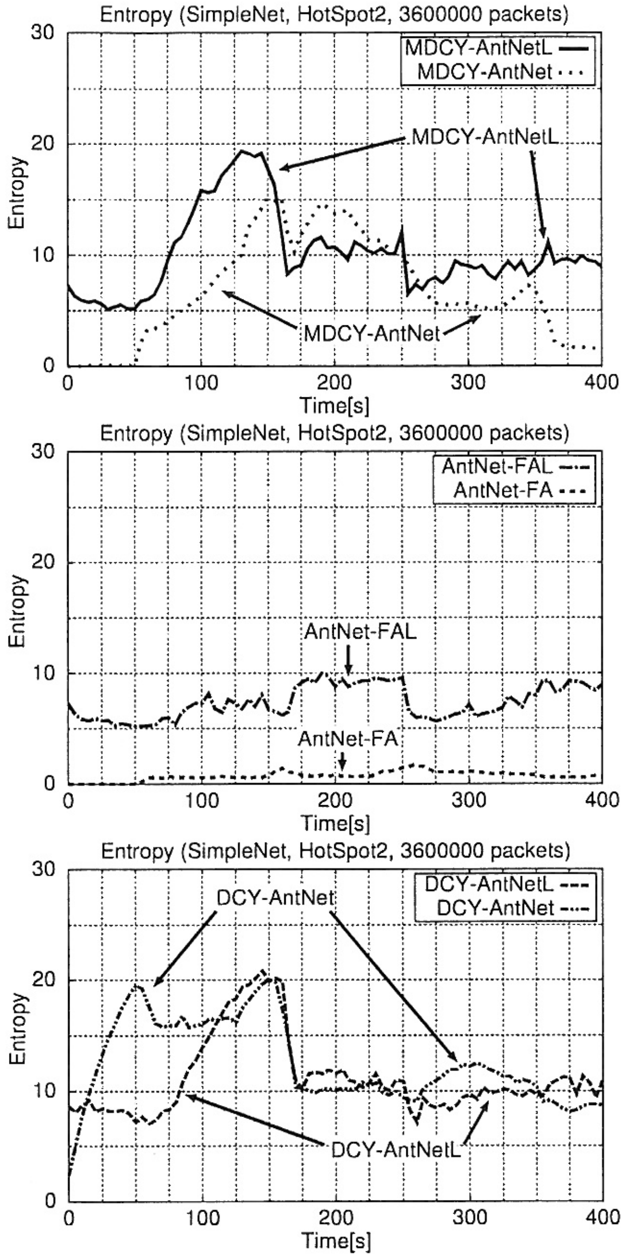

Fig. 7. Comparison of Entropy Dynamics

route is longer than that of direct route. So the reinforcement value r is small or little while the best time (i.e. the time using direct route) is still in the history in the node. Thus AntNet-FAL and AntNet-FA can't increase entropy because r is small.

Through the simulations there are some packets that can't be sent within TTL. From 50sec to 150sec the reasons are as follows: 1) Node 1 and node 6 can indirectly detect that the links from both node 8 to node 7 and the opposite link have been congested. 2) The route $Node1 \rightarrow Node8 \rightarrow Node7 \rightarrow Node6$ is the optimal path by distance. Thus, one link can afford the generated datagram packets before congestion, the path is fixed to send datagram packets from node 1 to node 6 and the opposite link. For the reasons above, Node 1 and node 6 use the route until congestion is detected. When the datagram packet launched from node 6 is about to use the link from node 8 to node 7, the datagram packet is queued. It is considered that the queue length is too long to send within TTL, 15sec.

At last, to do with traffic congestion for sparse networks such as SimpleNet, we consider the evaporation algorithm is effective.

V. CONCLUSION AND FUTURE WORK

We have proposed a relative pheromone evaporation algorithm, MDCY-AntNet, which showed better performance through an experiment than AntNet and DCY-AntNet.

Of course we have been inspecting effect of the Loop-Free feature from the view of network topology. Although an ant is dormant to make a loop in its history[11], the feature seems to be good. However there are many links in the network, this feature can affect the whole throughput and packet transfer distribution. Faltoutsos have discovered that the Internet has power-laws [12], scale-free properties in the Internet topology. Moreover the average degree of ASs and routers have been changing. In addition the average degree of routers is 2.53 and that of ASs is 4.2 [13]. We have to understand how an ant with the Loop-Free feature works on scale-free networks.

We have to consider how network topology and network traffic pattern affect the routing network ability.

REFERENCES

[1] John Moy. *RFC 2328, OSPF Specification Version 2*. Network Working Group, July 1998.

[2] Justin A. Boyan and Michael L. Littman. Packet routing in dynamically changing networks: A reinforcement learning approach. *In Advances in Neural Information Processing Systems*, pages 671–678, 1994.

[3] Samuel P. M. Choi and Dit-Yan Yeung. Predictive q-routing: A memory-based reinforcement learning approach to adaptive traffic control. *In Advances in Neural Information Processing Systems 8 (NIPS8)*, pages 945–951, 1996.

[4] Gianni Di Caro and Marco Dorigo. AntNet: Distributed stigmergetic control for communications networks. *Journal of Artificial Intelligence Research (JAIR)*, 9:317–365, 1998.

[5] Marco Dorigo, Vittorio Maniezzo, and Alberto Colorni. The ant system: Optimization by a colony of cooperating agents. *IEEE Transactions on Systems, Man, and Cybernetics*, 26:29–41, 1996.

[6] Casper Joost Eykelhof and Marko Snoek. Ant systems for a dynamic tsp - ants caught in a traffic jam. *Proceedings of Ants2002*, pages 88–99, 2002.

[7] Kazumasa Oida and Akira Kataoka. Lock-free AntNets and their adaptability evaluations. *Electronics and Communications in Japan (Part I: Communications)*, 84:51–61, 2000.

[8] Gianni Di Caro and Marco Dorigo. AntNet: A mobile agents approach to adaptive routing. *Technical Report IRIDIA 97-12*, 1997.

[9] Gianni Di Caro and Marco Dorigo. An adaptive multi-agent routing algorithm inspired by ants behaviour. *Proceedings of PART'98 - fifth Annual Australasian Conference on Parallel and Real-Time Systems*, 1998.

[10] Gianni Di Caro and Marco Dorigo. Two ant colony algorithms for best-effort routing in datagram networks. *Proceedings of PDCS'98 - 10th International Conference on Parallel and Distributed Computing and Systems*, 1998.

[11] Geoffrey Canright. Ants and loops. *Proceedings of Ants2002*, pages 235–242, 2002.

[12] Michalis Faloutsos, Petros Faloutsos, and Christos Faloutsos. On power-law relationships of the internet topology. *In Proceedings of ACM SIGCOMM*, pages 251–262, 1999.

[13] Hongsuda Tangmunarunkit, Ramesh Govindan, Sugih Jamin, Scott Shenker, and Walter Willinger. Network topology generators: Degree-based vs structural. *ACM SIGCOMM*, 2002.