Some Analysis on Mobile-Agent Based Network Routing*

Wenyu Qu and Hong Shen

Graduate School of Information Science
Japan Advanced Institute of Science and Technology
1-1 Asahidai, Tatsunokuchi, Ishikawa, 923-1292, Japan

qu_wenyu, shen@jaist.ac.jp

Abstract

Deployment of mobile agents in network-based applications has attracted lots of attentions in recent years. How to control the activities of agents is crucial for effective application of mobile agents. This paper focuses on the application of mobile agents in network routing. Two important activity properties of mobile agents are identified: the probability of success (the probability of finding the destination) and the distribution of mobile agents running in the network. To our knowledge, little work has been done on these two aspects. Our results show that the number of mobile agents can be controlled by adjusting the number of agents generated per request and the number of jumps each mobile agent can move. Thus, we can improve network performance by tuning relevant parameters.

Key words: Mobile agents, network routing, probability, population.

1 Introduction

This paper concentrates on the application of mobile agents in network routing. Mobile agent, a program that migrates autonomously from node to node on behalf of the user, has attracted considerable attentions. It can roam the network either on a predetermined path or one that the agent itself determines based on dynamically gathered information. Some merits of mobile-agent paradigm are described in [6, 9], including network load and latency reduction, protocol encapsulation, adaption, heterogeneity, robustness and fault-tolerance. There have been an increasing number of research activities to exploit mobile agents in various areas, ranging from e-commerce, distributed computation, to real time control.

Routing is a key for network performance, especially in large networks. In order to save network resources, it is desirable to dispatch a small number of mobile agents to get a high probability of success. In a network routing algorithm for large-scale networks, mobile agents will be generated frequently. If there are too many agents running in the network, they will consume too much network resources, which will affect network performance and ultimately block the entire network. On the other hand, if there are too few agents running in the network, it can not be guaranteed that the destination can be found quickly. Therefore, it is necessary to analyze both the probability of success and the population growth of mobile agents for network management. Although the efficiency of applying mobile agent techniques has been demonstrated and reported in the literature [2, 4, 8, 12, 14], little attention has been paid to the analysis of the population growth of mobile agents. In [10, 11], the authors presented some analysis on the population distribution of mobile agents running in the network, but they did not consider the probability of success. In this paper, we present some analysis on both the probability of success and the population growth of mobile agents. Our results show that we can control the probability of success and the number of mobile agents by tuning the number of agents generated per request and the number of jumps each mobile agent can move.

The rest of this paper is organized as follows. Section 2 introduces some preliminary knowledge for our analysis. Section 3 presents some analytical results on mobile agents, including the probability of success and the population of agents in network routing. Section 4 concludes the paper.

2 Mathematical Model

2.1 Mobile Agents

Mobile agent, a relatively new paradigm for network software development, has become an important technology

*This work was supported by Japan Society for the Promotion of Science (JSPS) Grant-in-Aid for Scientific Research under its grant for Special Research Domains.

in recent years. The word *agent*'s origin is deeply rooted in [8]: drawing from the Greek *agein*, which means to drive or lead; from medieval Latin *agere*, with a similar meaning; and from Old Norse, *aka*, which implies travel in a vehicle. A mobile agent, as defined in [5], is a small program that represents a user in a computer network and can migrate autonomously from node to node to perform some computation on behalf of the user. Its tasks, which are determined by the agent application, can range from on-line shopping to real-time device control to distributed scientific computing. The potential benefits of this technology, including the reduction of network bandwidth consumption and latency, have drawn a great deal of attention in both academia and industry [1, 2, 6, 7, 13, 14, 15].

When a mobile agent is encapsulated with a task, it is dispatched to a remote host. After executing and accomplishing its tasks, the summary report for its trip is sent back to the server. Since there are very few communications between the agent and the server during the process of searching; thus, the network traffic generated by mobile agents is very light. So mobile agent is an effective way for improving network performance.

2.2 Model

In our model, once a user keys in his request, the server will generate a number of mobile agents to search for the destination. Then, it will compare the cost of those pathes and decide the optimal path. Our model makes the following assumptions:

1. There are n nodes in the network, and each node has the same probability of $\frac{1}{n}$ to be the destination host.

2. A host can obtain its neighborhood information by periodically sending an update message to neighboring hosts or using the piggyback technique, that is, when a host needs to send a packet to its neighboring host, it attaches its information along with the packet.

3. At any time t, the expected number of requests keyed in one host is m. Once a request arrives, k agents are created and sent out into the network.

4. If an agent can not find its destination in the current host, it will jump to any of its neighboring hosts or stay in the current host with the same probability. If an agent stays in the current host, it will die at the next step.

5. To eliminate unnecessary searching in the network, we further assume that an agent will die if it can not find its destination host in d jumps.

The model works as follows: Once a request is keyed in, k mobile agents are generated. The agents traverse the network from the source host to search for the destination host. At each node, the agents check information of both the node itself and its neighboring hosts. If an agent finds its destination, it dies instantly. A new message agent is generated, and sends the message back to the source host. Otherwise, it will randomly select a neighboring host to move on. If an agent has not found its destination in d jumps, it dies immediately.

2.3 An Example

For example (see Figure 1), h_i and s_i are hosts in the network, and each s_i is also a destination. Once the server receives a request, k agents are generated and dispatched to search for the destinations (there may be multiple destinations for one request), before moving out, they check whether the server and the two neighboring host h_1, h_2 are the destinations. If they can not find a destination, they will jump into either h_1 or h_2. If an agent selects to jump into h_2, it will check information of the current host and its neighboring hosts, then it will find one of the destinations s_1, and jump into it to complete its tasks. We need not worry about that the destination s_3 will never be visited in our model because in real network, nodes are always hyper-linked with other nodes. At the same time, different requests visit different servers. In this paper, we consider the case in which there is only one destination. Our analysis can be easily extended to the cases in which there are numerous destinations.

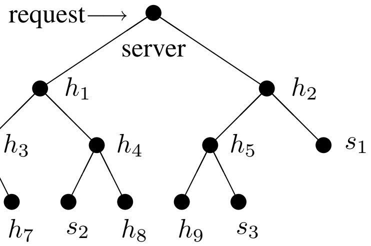

Figure 1: Model of a mobile-agent execution.

3 Analysis

In this section, we first introduce some notations and definitions used in this paper, then present our analysis of the probability of success and the population of mobile agents running in the network.

3.1 Preliminaries

As we have known, the topology of a network can be uniquely decided by its connectivity matrix. Therefore, the connectivity matrix plays an important role in network management. In this paper, we apply the connectivity matrix for our analysis. The network topology we used in this paper is a connected graph; thus, there is at least one path between any two hosts. Matrix $C = (c_{ij})_{n\times n}$ is the connectivity matrix which describes the connectivity of the graph, i.e., if there is a direct link between host i and host j, then

$c_{ij} = c_{ji} = 1$; otherwise, $c_{ij} = c_{ji} = 0$. Let c_j be the jth column vector of matrix C, then C can be expressed as $C = (c_1, c_2, \cdots, c_n)$. Let $d_j = \|c_j\|_1 = \sum_{i=1}^{n} |c_{ij}|$, $\sigma_1 = \max\limits_{1 \le j \le n} d_j$, $\sigma_n = \min\limits_{1 \le j \le n} d_j$; thus, $D = \text{diag}(d_1, d_2, \cdots, d_n)$ is a diagonal matrix. It is easy to see that d_j is the number of neighboring hosts of the jth host including itself and $\|C\|_1 = \max\limits_{1 \le j \le n} \|c_j\|_1 = \sigma_1$. $A = (C - I)D^{-1}(I - \frac{1}{n}D)$ is a matrix decided by the network. For example, suppose that the graphical structure of a network is shown in Figure 2, then we have the following results.

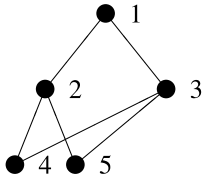

Figure 2: An Example

$n = 5$, $\sigma_1 = 4$, and $\sigma_5 = 3$. Matrix C and A are as follows:

$$C = \begin{pmatrix} 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 & 1 \end{pmatrix},$$

$$A = \begin{pmatrix} 0 & \frac{3}{20} & \frac{3}{20} & 0 & 0 \\ \frac{4}{15} & 0 & 0 & \frac{4}{15} & \frac{4}{15} \\ \frac{4}{15} & 0 & 0 & \frac{4}{15} & \frac{4}{15} \\ 0 & \frac{3}{20} & \frac{3}{20} & 0 & 0 \\ 0 & \frac{3}{20} & \frac{3}{20} & 0 & 0 \end{pmatrix}$$

3.2 Probability of Success

The probability of success directly affects the searching process, and affects the network performance as a result. However, the existing results have considered little on this probability. In this section, the probability of success is analyzed. Our results show that the probability of success is affected by the connectivity matrix of the network, the number of mobile agents generated per request, and the searching time.We assume that the distance between any two connecting nodes are the same; thus, we can consider the searching time by the jumping hops. The following lemma gives an estimation on both the upper bound and the lower bound of the probability of success that at agent can find its destination at dth jump:

Lemma 1 *The probability of success, $p(d)$, that an agent can find its destination at dth jump, satisfies the following inequality:*

$$\frac{\sigma_n}{n}\left(\frac{1}{\sigma_1} - \frac{1}{n}\right)^d \le p(d) \le \frac{\sigma_1}{n}\left(\frac{1}{\sigma_n} - \frac{1}{n}\right)^d \qquad (1)$$

Proof Denote the sequence number of the host that an agent enters at dth jump by J_d. After being generated by the source host, J_0, the agents begin searching for the destination. Then, $p(0)$, the probability that an agent can find its destination at birth, is equal to $\frac{d_{J_0}}{n}$, and the probability that it can not find the destination before the first jump, is equal to $1 - \frac{d_{J_0}}{n}$. If the agent can not find its destination, it will jump out and search on. The probability that it can find its destination at the first jump is $p(1) = \frac{d_{J_1}}{n}(1 - \frac{d_{J_0}}{n})\frac{1}{d_{J_0}} = \frac{d_{J_1}}{n}(1 - \frac{d_{J_0}}{n})\frac{1}{d_{J_0}}$, and the probability that it can not find its destination at the first jump is $(1 - \frac{d_{J_1}}{n})(\frac{1}{d_{J_0}} - \frac{1}{n})$. Similarly, $p(2) = \frac{d_{J_2}}{n}(\frac{1}{d_{J_0}} - \frac{1}{n})(\frac{1}{d_{J_1}} - \frac{1}{n})$, $p(3) = \frac{d_{J_3}}{n}(\frac{1}{d_{J_0}} - \frac{1}{n})(\frac{1}{d_{J_1}} - \frac{1}{n})(\frac{1}{d_{J_2}} - \frac{1}{n})$, and so on. By recursion, the probability that an agent can find its destination at the dth jump satisfies:

$$\begin{aligned} p(d) &= \frac{d_{J_d}}{n}\prod_{i=0}^{d-1}\left(\frac{1}{d_{J_i}} - \frac{1}{n}\right) \\ &\le \frac{\sigma_1}{n}\prod_{i=0}^{d-1}\left(\frac{1}{\sigma_n} - \frac{1}{n}\right) \\ &= \frac{\sigma_1}{n}\left(\frac{1}{\sigma_n} - \frac{1}{n}\right)^d \end{aligned}$$

and similarly

$$p(d) \ge \frac{\sigma_n}{n}\prod_{i=0}^{d-1}\left(\frac{1}{\sigma_1} - \frac{1}{n}\right) = \frac{\sigma_n}{n}\left(\frac{1}{\sigma_1} - \frac{1}{n}\right)^d$$

Hence, the lemma is proven.

It is easy to see that the probability $p(d)$ is relevant to the topology of the network and the number of jumps. With the number of jumps increasing, the probability of success decreases rapidly.

Theorem 1 *The probability of success, $P(d)$, that an agent can find its destination in d jumps satisfies:*

$$P(d) < \frac{\sigma_1(n - \sigma_n)}{n(n\sigma_n + \sigma_n - n)} \qquad (2)$$

Proof By Lemma 1, we have

$$\begin{aligned} P(d) &\le \sum_{t=1}^{d}\frac{\sigma_1}{n}\left(\frac{1}{\sigma_n} - \frac{1}{n}\right)^t \\ &= \frac{\sigma_1}{n} \cdot \left(\frac{1}{\sigma_n} - \frac{1}{n}\right) \cdot \frac{1 - (\frac{1}{\sigma_n} - \frac{1}{n})^d}{1 - (\frac{1}{\sigma_n} - \frac{1}{n})} \\ &= \frac{\sigma_1(n - \sigma_n)}{n(n\sigma_n + \sigma_n - n)}\left[1 - \left(\frac{1}{\sigma_n} - \frac{1}{n}\right)^d\right] \end{aligned}$$

Hence, the lemma is proven.

From Theorem 1, it is easy to see that $P(d)$ increases slower and slower with the increase of d. When d is large enough, the increase of $P(d)$ is little, therefore, it is not necessary to go on searching if the increase of $P(d)$ is small enough.

3.3 The Population of Agents

Mobile agents are generated frequently and dispatched to the network. If the number of mobile agents is small, it can not ensure that the destination can be found quickly. But if there are too many agents in the network, they will introduce too much computational overhead to host machines, which will eventually become very busy and indirectly block the network traffic. Therefore, it is necessary to analyze the number of mobile agents running in the network or at each host. As far as we know, little attention has been paid on the analysis of the population growth of mobile agents.

In this section, a general matrix-vector format distribution-formula of mobile agents running in the network is given, and the number of agents running both in the network and at each host is estimated.

First, we analyze the distribution of mobile agents running in the network without considering the bound of jumping hops for each mobile agent. It is easy to see that the distribution of mobile agents is a stochastic process. Assume at time $t-1$, there are $p_i(t-1)$ agents running in the ith host, then at time t, those agents that can not find the destination jump to the neighboring hosts of the ith host or die. As described in the model, the mean number of agents jumping into each neighboring host from the ith host at time t is $(1-\frac{d_i}{n})\frac{p_i(t-1)}{d_i}$. Therefore, at time t, the number of agents running in the jth host consists of two parts: km agents are newly generated, and $\sum_{i\in NB(j)}(1-\frac{d_i}{n})\frac{p_i(t-1)}{d_i}$ agents come from the neighboring hosts of the jth host. This dynamic process can be described as follows:

$$p_j(t) = km + \sum_{i\in NB(j)} \left(1-\frac{d_i}{n}\right)\frac{p_i(t-1)}{d_i} \tag{3}$$

which is obviously a Markov Process . Let$\overrightarrow{p}(t) = (p_1(t), p_2(t), \cdots, p_n(t))^T$, and $\overrightarrow{e} = (1, 1, \cdots, 1)^T$, then we can express the distribution-formula of mobile agents running in the network in matrix-vector format and have the following theorem.

Theorem 2 *The population distribution of mobile agents running in the network can be expressed in matrix-vector form as follows:*

$$\overrightarrow{p}(t) = km\overrightarrow{e} + A\overrightarrow{p}(t-1) \tag{4}$$

Proof Since

$$\begin{aligned} p_j(t) &= km + \sum_{i\in NB(j)} \left(1-\frac{d_i}{n}\right)\frac{p_i(t-1)}{d_i} \\ &= km + \sum_{i\in NB(j)} \left(\frac{1}{d_i}-\frac{1}{n}\right)p_i(t-1) \end{aligned}$$

$\overrightarrow{p}(t)$ can be expressed as follows:

$$\overrightarrow{p}(t) = km\overrightarrow{e} + (C-I)D^{-1}\left(I-\frac{1}{n}D\right)\overrightarrow{p}(t-1)$$

Since $A = (C-I)D^{-1}(I-\frac{1}{n}D)$, the proof is completed.

Theorem 2 shows that the distribution of mobile agents running in the network is decided by the connectivity matrix of the network, the time mobile agents alive, the initial distribution of the mobile agents, and the generating rate of mobile agents per request. From Theorem 2, we can easily get the following corollary.

Corollary 1 *If there are $\overrightarrow{q}(t-d)$ agents generated at time $t-d$, then the distribution of these agents at time t is $A^d km\overrightarrow{q}(t-d)$.*

Proof As shown in Theorem 2, the distribution formula of mobile agents should be

$$\overrightarrow{p}(t) = A\overrightarrow{p}(t-1)$$

Thus, by recursion, we can easily get the results.

Corollary 1 indicates that the number of mobile agents decreases with time t and the decreasing rate is decided by the connectivity matrix of network.

Based on the analysis above, we further analyze the distribution of mobile agents running in the network under the assumption that each agent can jump at most d hops. The distribution of mobile agents after d jumps is shown in the following theorem.

Theorem 3 *The distribution of agents can be described as:*

$$\overrightarrow{p}(t) = \begin{cases} 0 & t = 0 \\ (I + A + \cdots + A^{t-1})km\overrightarrow{e} & 0 < t \le d \\ (I + A + \cdots + A^{d-1})km\overrightarrow{e} & t > d \end{cases} \tag{5}$$

Proof From Theorem 2 and the assumption $\overrightarrow{p}(0) = 0$, we can get the following conclusion by recursion:

$$\begin{aligned} \overrightarrow{p}(t) &= A\overrightarrow{p}(t-1) + km\overrightarrow{e} \\ &= (I + A + \cdots + A^{t-1})km\overrightarrow{e} \end{aligned}$$

when $t \leq d$. As a result, when $t \geq d$,

$$\begin{aligned}\overrightarrow{p}(t) &= A\overrightarrow{p}(t-1) + km\overrightarrow{e} - A^d km\overrightarrow{e} \\ &= A^{t-d}\overrightarrow{p}(d) + (I + A + \cdots + A^{t-d-1})km\overrightarrow{e} \\ &\quad -A^d(I + A + \cdots + A^{t-d-1})km\overrightarrow{e} \\ &= (I + A + A^2 + \cdots + A^{d-1})km\overrightarrow{e}\end{aligned}$$

So the proof is completed.

Theorem 3 indicates that the numbers of agents running in the network and in each host are decided by the size of the network, the topology of the network, and m, k, d. From this result, we can further estimate the total number of agents running in the network or at each host:

Theorem 4 *The total number of agents running in the network is not greater than* $(\sqrt{n}-1)^2(d-1)km$.

Proof By equation (5) and the definition of matrix norm, we can get

$$\begin{aligned}\sum_{j=1}^{n} p_j(t) &= \|\overrightarrow{p}(t)\|_1 \leq \sum_{s=1}^{d-1} \|A\|_1^s \cdot nkm \\ &\leq (d-1)\|A\|_1 nkm\end{aligned}$$

Since

$$\begin{aligned}\|A\|_1 &= \max_{1 \leq i \leq n}\left[\frac{d_i - 1}{d_i}\left(1 - \frac{d_i}{n}\right)\right] \\ &\leq \frac{\sqrt{n}-1}{\sqrt{n}}\left(1 - \frac{1}{\sqrt{n}}\right) = \left(1 - \frac{1}{\sqrt{n}}\right)^2\end{aligned}$$

The proof is completed.

Now, we focus on the number of agents running in each host. We can get an upper bound of $p_j(t)$ as follows:

Theorem 5 *The number of agents running is the jth host is less than or equal to:*

$$p_j(t) \leq km + \frac{n - \sigma_n}{\sigma_n^2 + n - \sigma_n}(d_j - 1)km \qquad (6)$$

Proof First, we prove the following inequality by incomplete induction:

$$p_j(t) \leq d_j km - \frac{\sigma_n^2}{\sigma_n^2 + n - \sigma_n}(d_j - 1)km$$

it is easy to see that at time $t = 0$, this inequality is satisfied. Assume that at time $t - 1$, the inequality is also hold, then at time t, since

$$\begin{aligned}&\left(1 - \frac{d_j}{n}\right)\frac{p_j(t-1)}{d_j} \\ &\quad \leq \left(1 - \frac{\sigma_n}{n}\right)km \\ &\qquad -\frac{\sigma_n^2}{\sigma_n^2 + n - \sigma_n}\left(1 - \frac{\sigma_n}{n}\right)\frac{\sigma_n - 1}{\sigma_n}km \\ &\quad = km - \frac{\sigma_n^2}{\sigma_n^2 + n - \sigma_n}km\end{aligned}$$

we have

$$\begin{aligned}p_j(t) &= km + \sum_{i \in NB(j)}\left(1 - \frac{d_i}{n}\right)\frac{p_i(t-1)}{d_i} \\ &\leq km + \left[km - \frac{\sigma_n^2}{\sigma_n^2 + n - \sigma_n}km\right](d_j - 1) \\ &= d_j km - \frac{\sigma_n^2}{\sigma_n^2 + n - \sigma_n}(d_j - 1)km\end{aligned}$$

Thus, the inequality is proven. Thus, $p_j(t)$ can be rewritten as:

$$\begin{aligned}p_j(t) &\leq d_j km - \frac{\sigma_n^2}{\sigma_n^2 + n - \sigma_n}(d_j - 1)km \\ &= km + \frac{n - \sigma_n}{\sigma_n^2 + n - \sigma_n}(d_j - 1)km\end{aligned}$$

Hence, the theorem is proven.

Corollary 2 *The upper bound of $p_j(t)$ in Theorem 5 can be simplified as $p_j(t) \leq d_j km$.*

Proof Due to

$$\frac{\sigma_n^2}{\sigma_n^2 + n - \sigma_n} < 1$$

and

$$p_j(t) \leq d_j km - \frac{\sigma_n^2}{\sigma_n^2 + n - \sigma_n}(d_j - 1)km$$

The corollary is proven.

Let $f_j(t)$ denotes the number of agents moving out from the jth host to each neighboring host at time t, we have:

Corollary 3 *The number of agents moving out from the jth host to each neighboring host at time t, $f_j(t)$, satisfies:*

$$f_j(t) \leq \frac{n - \sigma_n}{n - \sigma_n + \sigma_n^2}km \qquad (7)$$

Proof From Theorem 5, we have the following inequality:

$$\left(1 - \frac{d_j}{n}\right)\frac{p_j(t-1)}{d_j} \leq km - \frac{\sigma_n^2}{\sigma_n^2 + n - \sigma_n}km$$

Thus the definition of $f_j(t)$, the corollary is proved.

4 Concluding Remarks

In this paper, we analyzed the application of mobile agents in network routing. We first proposed a model for the application of mobile agents in network routing, and then presented some analysis on both the probability of success and the population distribution of mobile agents. Our results showed that the probability of success, the total number of mobile agents running in the network, the number of mobile agents running in each host, and the number of mobile agents moving through each link are decided by the number of mobile agents generated per request, the time that each mobile agent has to search for the destination, and the connectivity matrix of network. It is possible to dispatch a small number of mobile agents to get a high probability of success by tuning the relevant parameters.

References

[1] F. M. T. Brazier, B. J. Overeinder, M. Steen, and N. J. E. Wijngaards. *Agent Factory: Generative migration of Mobile Agents in Heterogeneous Environments*. In Proc. of the ACM Symposium on Applied Computing (SAC02), pp. 101-106, 2002.

[2] J. Claessens, B. Preneel, and J. Vandewalle. *(How) Can Mobile Agents Do Secure Electronic Transactions on Untrusted Hosts? A Survey of the Security Issues and the Current Solutions*. ACM Transactions on Internet Technology, vol. 3, No. 1, Pages 28-48, February 2003.

[3] Kevin Curran, Derek Woods, Nadene McDermot and Colleen Bradley. *The Effects of Badly Behaved Routers on Internet Congestion*. International Journal of Network Nanagement, 2003, 13: 83-94.

[4] Stuart Feldman. *Electronic Marketplaces*. IEEE Internet Computing, July-August 2000, pp. 93-95.

[5] Neeran M. Karnik and Anand R. Tripathi. *Design Issues in Mobile-Agent Programming Systems*. IEEE Concurrency, July-September 1998, pp. 52-61.

[6] D. Lange and M. Osima. *Seven Good Reasons for Mobile Agents*. Communications of the ACM, Vol. 42, pp. 88-89, 1999.

[7] T. Li and K. Lam. *An Optimal Location Update and Searching Algorithm for Tracking Mobile Agent*. AAMAS'02, pp. 15-19 Bologna, Italy, July, 2002.

[8] Dejan Milojicic. *Guest Editor's Intorduction: Agent Systems and Applications*. IEEE Concurrency, April-June 2000 (Vol. 8, No. 2). pp. 22-23.

[9] Nelson Minar, Matthew Gray, Oliver Roup, Raffi Krikorian, and Pattie Maes Mit Media Lab. *Hive: Distributed Agents for Networking Things*. IEEE Concurrency, April-June 2000 (Vol. 8, No. 2), pp. 24-33.

[10] J. Sum, H. Shen, C. Leung, and G. Young. *Analysis on a Mobile Agent-BasedAnt Algorithm for Network Routing and Management*. IEEE Transaction on Paralell and Distributed Systems, Vol. 14, No. 3, pp. 193-2002, 2003.

[11] J. Sum, H. Shen, G. Young, J. Wu, and C. Leung. *Analysis on Extended Ant Routing Algorithms for Network Routing and Management*. The Journal of Supercomputing, 24, pp. 327-340, 2003.

[12] Tammo Spalink, John H. Hartman and Garth A. Gibson. *A Mobile Agent's Effects on File Service*. IEEE Concurrency, April-June 2000 (Vol. 8, No. 2), pp. 62-69.

[13] L. Tang and B. Pagurek. *A Comparative Evaluation of Mobile Agent Performance for Network Management*. Proceedings of the Ninth Annual IEEE International Conference and Workshop on the Engineering of Computer-Based Systems(ECBS'02), 2002.

[14] Y. Wang *Dispatching Multiple Mobile Agents in Parallel for Visiting E-Shops*. Proceedings of the Third International Conference on Mobile Data Management(MDM'02), 2002.

[15] D. Wong, N. Paciorek, and D. Moore. *Java-Based Mobile Agents*. Communications of the ACM, Vol. 42, pp.92-102, 1999.