FURTHER ANALYSIS ON THE APPLICATION OF MOBILE AGENTS IN 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

John Sum
Department of Computing
Hong Kong Polytechnic University
Hung Hom, KLN, Hong Kong

Keywords: Mobile agents, routing, probability of success, population distribution.

Abstract: Mobile agent-based routing is a newly proposed routing technique for using 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 finding the destination (probability of success). Therefore, it is not only necessary but also important to analyze the searching activity and the population growth of mobile agents for improving the performance in agent-driven networks. Yet currently there is a lack of such analysis. In this paper, we present a new mobile agent-based routing model for describing the behavior of mobile agents for network routing. Then we analyze both the probability of success and the population growth of mobile agents running in the network. The theoretical results show that the probability of success and the number of mobile agents can be controlled by adjusting relevant parameters according various network characteristics. Our results reveal new theoretical insights into the statistical behaviors of mobile agents and provide useful tools for effectively managing mobile agents in large networks.

1 INTRODUCTION

With dramatic advances in the Internet and in the computer industry, computers are no longer isolated number factories. Many new applications, from e-business to e-government and e-education, have been created, thanks to the exponential growth of the Internet user base and the widespread popularity of the World Wide Web. Mobile agents, programs that can migrate from host to host in a network, at times and to places of their own choosing (Kotz and Gray, 1999), are changing the face of e-business and reshaping current business models (Wagner and Turban, 2002). In (Lange and Osima, 1999), Lange et al. concluded that mobile agents can reduce network load, overcome network latency, encapsulate protocols, execute autonomously and asynchronously, and adapt dynamically. They are naturally heterogeneous, robust and fault-tolerant to changing environments. In short, mobile agents are software entities that "bring the computation to the data rather than the data to the computation" (Schoder and Eymann, 2000).

Routing is at the core of e-business(D.M. Piscitello and Chapin, 1993), especially in large-scale networks (K. Curran and Bradley, 2003). By the definition in (Caro and Dorigo, 1998a), routing is the distributed activity of building and using routing tables, one for each node in the network, which tell incoming data packets which outgoing link to use to continue their travel towards the destination node. The main task of a routing algorithm is to direct data flow from source to destination nodes by maximizing network performance and minimizing user's costs.

Mobile agent-based routing is a newly proposed technique which adapts to the tremendous growth of the size of the Internet and the latest development of mobile computing. In a mobile agent-based routing algorithm (White, 1997), once a request for sending a message is received from the server, the server will generate a number of mobile agents. Those agents will then move out from the server and search for the destination. Once an agent has reached the destination, it turns back to the server along the searched path, and reports the path to the server. The server picks up the desired path from all the path collected, sends the message along the selected path, and updates its routing table at the same time. The report of an agent is given to the server only when the agent finishes its trip, but not in the middle of the trip; thus, there are very few communications between the agent

Qu W., Shen H. and Sum J. (2004).
FURTHER ANALYSIS ON THE APPLICATION OF MOBILE AGENTS IN NETWORK ROUTING.
In *Proceedings of the First International Conference on E-Business and Telecommunication Networks*, pages 204-212
DOI: 10.5220/0001405902040212

and the server during the searching process. Therefore, the network traffic generated by mobile agents is very light.

Different mobile agent-based routing schemes result in different network performance in terms of both quality and quantity of delivered service (Caro and Dorigo, 1998b). Two parameters are important in estimating a mobile agent-based routing model: the probability of finding the destination and the number of mobile agents being employed. It is easy to see that mobile agents will be generated and dispatched into the network frequently. Thus, they will certainly consume a certain amount of network resources. To save network resources, it is desirable to dispatch a small number of mobile agents and achieve a good probability of success. Therefore, performance analysis of the searching activity and population growth of agents is not just important, but necessary for improving performance of agent-driven networks. Unfortunately, such analysis of mobile agent behavior is in its infancy (Kim and Robertazzi, 2000), and little attention has been paid to the probability of success.

In this paper, we propose a new mobile agent-based routing model which tallies with the non-stationary stochastic nature of the Internet. Then we analyze it on both the probability of success and the population growth of mobile agents. The communication network we focused on is a connecting network with irregular topology. Our results show that both the probability of success and the number of mobile agents can be controlled 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 presents our model, section 3 introduces the notation used in this paper and analyzes both the probability of success and the population of agents in network routing, and section 4 concludes our paper.

2 MATHEMATICAL MODEL

In a mobile agent-based network routing model, a mobile agent will visit a sequence of hosts. The sequence of hosts between the server and the destination is called the itinerary of the mobile agent. Whereas a static itinerary is entirely defined at the server and does not change during the agent travelling. A dynamic itinerary is subject to modifications by the agent itself. In this paper, we propose a dynamic routing model that is well suited for routing in a faulty network or mobile network. Our model can be seen as an extended ant routing.

2.1 An Ant Routing Algorithm

As searching for the optimal path between two hosts in a stationary network is already a difficult problem, searching for the optimal path in a faulty network or mobile network will be much more difficult (Garey and Johnson, 1979). The ant routing algorithm is a recently proposed routing algorithm for use in this environment. The idea is inspired by the observation of real ant colonies. Individual ants are behaviorally simple insects with limited memory and exhibiting activity that has a stochastic component. However, ant colonies can accomplish complex tasks due to highly structured social organizations (M. Dorigo, 2000). Ant routing algorithm is designed taking inspiration from studies of the behavior of ant colonies (J. Sum and Young, 2003). The basic idea can be described as follows: Once a connection request has been received from a server, the server will generate a number of ants (the explorer agents). Those ants will then leave the source and explore the network. On each intermediary host, they choose a path with a probability proportional to the heuristic value (function of the cost and the favorite level) associated with the link. The ants cannot visit a host twice (they keep a tabu list of their visited hosts) and cannot use a link if there is insufficient bandwidth available. Once the destination is reached, the ants return from whence they came by popping their tabu list. On their way back, they lay down a pheromone-like trail. The server decides the desirable path from those collected, and sends a special kind of ant, the allocator, to allocate the bandwidth on all links used between the source and the destination. When the path is no longer required, a de-allocator agent is sent out to deallocate the network resources used on the hosts and links.

2.2 Our Model

In our model, mobile agents possess of some capabilities which real ants have not but are well suited to the network routing applications. For example, mobile agents are sighted (they can check information of both the host it stays and the neighbor hosts) which can improve the work efficiency of agents. They are restricted with a life-span limit (an agent will die if it can not find its destination in given steps) which can eliminate unnecessary searching in the network. Our model makes the following assumptions:

1. There are n hosts in the network, and each host has the same probability of $1/n$ to be the destination host.
2. 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.

3. When an agent reaches a host, it will check whether that current host is its destination. If the agent cannot find its destination in the current host, it will jump to one of the neighboring hosts.
4. Once an agent reaches its destination, it submits its goods list to the host and dies. After the host fulfils the relevant requirements, a new agent is generated and dispatched to the server with the resulting information.
5. To prevent the user from waiting too long, and to reduce unnecessary searching in the network, we further assume that if an agent cannot find its destination in d jumps, it will die.

Our model works as follows: At any time, there are lots of requests keyed in the network. Once a request is received from a server, a number of agents are created and sent out into the network. Those agents traverse the network from the server to search for the destination. At each host, the agents check information of both the host itself and its neighbor hosts. The probability that an agent can find its destination at its jth jump is $p(j)$. If an agent has reached its destination, it sends the collected information back to the server along the path searched immediately. Otherwise, the agent will select neighbor host and move on. An agent will die if it has not found its destination before the life-span expires.

3 ANALYSIS

Although mobile agents are emerging in diverse application fields, and their effectiveness and efficiency have been demonstrated and reported in the literature, the theoretical analysis of mobile agent behavior is in its infancy. For filling this need, we develop some stochastic analysis on both the probability of success and the population distribution of mobile agents running in the network in this section. First we introduce the notations and definitions used in our analysis.

3.1 Notions and Definitions

The topology of a network can be uniquely decided by its connectivity matrix. In this paper, we make use of the connectivity matrix in our analysis. The network topology we consider in this context is a connected graph, thus there is at least one link between any two hosts. Let matrix $C = (c_{ij})_{n\times n}$ be 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: $C = (c_1, c_2, \cdots, c_n)$. $d_j = \|c_j\|_1 = \sum_{i=1}^{n} |c_{ij}|$, $\sigma_1 = \max_{1\le j\le n} d_j$,

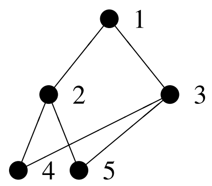

Figure 1: An Example of a Small Network

$\sigma_n = \min_{1\le j\le n} d_j$. $D = \text{diag}(d_1, d_2, \cdots, d_n)$ 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 $\|C\|_1 = \max_{1\le j\le n} \|c_j\|_1 = \sigma_1$. For example, Figure 1 shows the graphical structure of a small network. Accordingly, $n = 5$, $\sigma_1 = 4$, and $\sigma_5 = 3$. Matrix C is as follows:

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

3.2 Probability of Success

The probability of success directly affects the searching process, and affects the network performance as a result. However, it has not been sufficiently taken into account in existing works. In this section, the probability of success for both single agent and multiple agents are analyzed. Our results show that the probability of success is affected by the connectivity matrix of the network, the number of mobile agents, and the life-span limit. The following lemmas give both upper bound and lower bound on the probability of success that an agent can find its destination at dth jump:

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

$$\frac{\sigma_n}{n}\left(1-\frac{\sigma_1}{n}\right)^t < p(t) < \frac{\sigma_1}{n}\left(1-\frac{\sigma_n}{n}\right)^t$$

Proof Denote the sequence of the host in the itinerary of an agent by $J_0, J_1, \cdots$, and denote the set of neighbor hosts of the jth host by $NB(j)$. After being generated by the server, J_0, the agents begin searching for the destination. Then, $p(0)$, the probability that an agent can find its destination at birth, equals to d_{J_0}/n, and the probability that it can not find the destination before the first jump, equals to $1 - d_{J_0}/n$. 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 $p(1) = \sum_{i\in NB(J_0)} \frac{1}{d_{J_0}} [1 - p(0)] \frac{d_i}{n}$.

Then, $p(2)$, the probability that an agent can find its destination at the second jump is $p(2) = \sum_{j \in NB(J_1)} \frac{1}{d_{J_1}} [1-p(0)] [1-p(1)] \frac{d_j}{n}$. By recursion, it is easy to prove that the probability, $p(t)$, that an agent can find its destination at the tth jump satisfies:

$$p(t) = \sum_{l \in NB(t-1)} \left\{ \frac{1}{d_{J_{t-1}}} \cdot \frac{d_l}{n} \cdot \prod_{k=0}^{t-1} [1-p(k)] \right\}$$

Due to $\sigma_n \leq d_{J_i} \leq \sigma_1$ for any i, it is easy to prove that

$$\frac{\sigma_n}{n} \prod_{k=0}^{t-1} [1-p(k)] \leq p(t) \leq \frac{\sigma_1}{n} \prod_{k=0}^{t-1} [1-p(k)]$$

Hence, the lemma is proven. □

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

With the result of probability of success for each jump, we can estimate the probability of success for an agent during its life-span as follows:

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

$$\frac{\sigma_n (n-\sigma_1)^2}{n^2 \sigma_1} \left[1 - \left(1 - \frac{\sigma_1}{n}\right)^{d-1} \right] < P(d) < \frac{\sigma_1 (n-\sigma_n)^2}{n^2 \sigma_n} \left[1 - \left(1 - \frac{\sigma_n}{n}\right)^{d-1} \right]$$

Proof In our model, agents will not be generated and dispatched into the network if the destination is the server or a neighbor host of the server. Based on Lemma 1 and the fact that $P(d) = \sum_{t=2}^{d} p(t)$, the theorem is easily proven. □

The significance of Theorem 1 can be visualized from Figure 2.

We have conducted may experiments on various cased. Due to space limitation, we only list three cases for the purpose of explanation.

- Case 1: $n = 10000$, $\sigma_1 = \sigma_n = n$;
- Case 2: $n = 10000$, $\sigma_1 = 100$, $\sigma_n = 50$;
- Case 3: $n = 10000$, $\sigma_1 = 100$, $\sigma_n = 10$.

Case 1 illustrates a complete connected network, since any host in such a network knows the address of all the other hosts in the same network, $P(d)$ equals to a constant 1. In both case 2 and case 3, $P(d)$ is a monotonically increasing function on the jumping hops d. The connectivity of the network affects the increase of the probability of success. It also can be

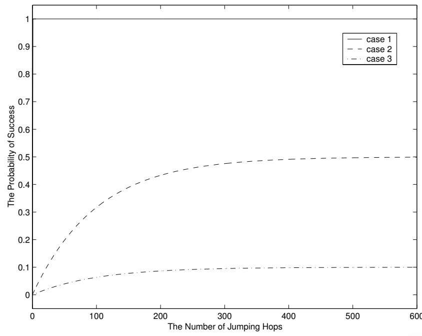

Figure 2: Variety of $P(d)$ with d.

seen 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. So, the life-span of an agent can be set according to a threshold value ϵ such that $P(d) - P(d-1) \leq \epsilon$.

In a multi-agent system, multiple agents may be dispatched to search for the same destination. For the case that the user doesn't really want all the dispatched agents, but part of them, find the destination, we estimate the probability of success that at least s agents from k agents can find its destination. Based on the conclusions above, we present the following theorem:

Theorem 2 *The probability of success, P_s, that at least s agents from k agents can find the destination in d jumps satisfies the following inequality:*

$$\frac{\triangle^s}{\sqrt{2\pi s}} < P_s < \frac{\triangle^s}{\sqrt{2\pi s}} \cdot \frac{1}{1-\lambda/(s+1)} \tag{1}$$

where $\triangle = \frac{\lambda}{s} e^{1-\lambda/s}$, $\lambda = kP(d)$.

Proof This probability satisfies binomial distribution, so we have

$$P_s = \sum_{i=s}^{k} C_k^i P(d)^i [1-P(d)]^{k-i}$$

It can be approximated by a Poison distribution as $P_s \approx \sum_{i=s}^{k} \frac{\lambda^i e^{-\lambda}}{i!}$. Besides, we have $\sum_{i=s}^{k} \frac{\lambda^i e^{-\lambda}}{i!} > e^{-\lambda} \frac{\lambda^s}{s!}$ and

$$\sum_{i=s}^{k} \frac{\lambda^i e^{-\lambda}}{i!} \leq e^{-\lambda} \frac{\lambda^s}{s!} \cdot \left[1 + \frac{\lambda}{s+1} + \frac{\lambda^2}{(s+1)^2} + \cdots + \frac{\lambda^{k-s}}{(s+1)^{k-s}} \right] \leq e^{-\lambda} \frac{\lambda^s}{s!} \cdot \frac{1}{1-\lambda/(s+1)}$$

Applying Stirling's formula: $n! \approx \sqrt{2\pi n} n^n e^{-n}$ for large n, we have

$$\frac{\triangle^s}{\sqrt{2\pi s}} < P_s < \frac{\triangle^s}{\sqrt{2\pi s}} \frac{1}{1-\lambda/(s+1)}$$

Hence, the theorem is proven. □

Especially, P_k, the probability of success that all the k agents find the destination, is no more than $P(d)^k$, and P_1, the probability of success that at least one agent from k agents can find the destination, is no more than $1-[1-P(d)]^k$. Furthermore, from Theorem 1 and Theorem 2, we have the following corollary:

Corollary 1 *The probability of success, P_s, that at least s agents from k agents can find the destination in d jumps satisfies the following inequality:*

$$P_s < \frac{1}{\sqrt{2\pi s}[1-k\alpha/(s+1)]} \quad (2)$$

where $\alpha = [\sigma_1(n-\sigma_n)^2]/[n^2\sigma_n]$.

Proof If we denote $x = \lambda/s$ in equation (1), then $\triangle = e \cdot xe^{-x}$. Let $f(x) = xe^{-x}$, it is easy to prove that $f(x)$ gets the maximum value when $x = 1$. That is $\triangle \leq 1$. Furthermore, $P(d) < \alpha$, hence the corollary is proven. □

3.3 Population of Agents

In this section, we estimate the number of agents running both in the network and on each host. According to the number of requests keyed in the network, mobile agents will be generated and dispatched to the network frequently. 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. In order to reduce the agents' population in the network, a life-span limit d is assumed. Generally, there are two approaches for designing the life-span limit d for mobile agents. One is to set d according to the expected number of hops an agent can jump, another is to set d by a threshold ϵ such that $P\{\text{steps} \geq d\} \leq \epsilon$. In this context, d is given before agents dispatched into the network.

In the first instance, we analyze the distribution of mobile agents running in the network without considering about 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 that 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 will either 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 $(1-\frac{d_i}{n})\frac{p_i(t-1)}{d_i-1}$. Therefore, at time t, the number of agents running in the jth host consists of two parts: km agents are newly generated, and $\sum_{i\in NB(j)}(1-\frac{d_i}{n})\frac{p_i(t-1)}{d_i-1}$ agents come from the neighboring hosts of the jth host, where $NB(j)$ is a set which consists by all the neighboring hosts of the jth host, and m is the average number of requests initiated at time t in a host. This dynamic process can be described as follows:

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

which is obviously a Markov Process . Let $\overrightarrow{p}(t) = (p_1(t), p_2(t), \cdots, p_n(t))^T$, $A = (C-I)(D-I)^{-1}[I-(1/n)D]$ is a matrix decided by the network (obviously we have $\|A\|_1 = 1-\sigma_n/n$) and $\overrightarrow{e} = (1,1,\cdots,1)^T$, then we can represent the population distribution of mobile agents running in the network in matrix-vector format as follows,

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

where the first term represents the newly generated agents and the second term is for those surviving agents generated previously. Equation (4) 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 Equation (4), we can obtain the following lemma:

Lemma 2 *Assume that there are $\overrightarrow{q}(t-d)$ agents generated at time $t-d$ for $t > d$. Then the distribution of these agents at time t is $A^d km\overrightarrow{q}(t-d)$.*

Proof As shown in Equation (4), the distribution formula of mobile agents should be

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

Hence, by recursion, the lemma is proven. □

Lemma 2 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 lemma.

From Equation 4 and Lemma 2, the distribution of mobile agents running in the network under the assumption that each agent can jump at most d hops can be expressed as follows:

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

$$\overrightarrow{p}(t) = \begin{cases} 0 & t = 0 \\ \sum_{i=0}^{t-1} A^i km\overrightarrow{e} & 0 < t \leq d \\ \sum_{i=0}^{d-1} A^i km\overrightarrow{e} & t > d \end{cases} \quad (5)$$

Proof By Lemma 2, if the distribution of newly generated agents is $km\overrightarrow{e}$, then the distribution of these agents after d hops is $A^d km\overrightarrow{e}$. In our model, all agents generated at time $t-d$ will die after $d$hops at time t. Therefore, these agents will be deduced from the total distribution. From Equation (4) 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 $t \geq d$,

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

Hence, the theorem is proven. □

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 connectivity of the network, and m, k, d. From this result, we can further estimate the total number of agents running in the network or on each host:

Theorem 4 *The total number of agents running in the network is less than $(n - \sigma_n)(d-1)km$.*

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

$$\begin{aligned} \sum_{j=1}^{n} p_j(t) &= \|\overrightarrow{p}(t)\|_1 \\ &\leq \begin{cases} 0 & t = 0 \\ \sum_{s=1}^{t-1} \|A\|_1^s \cdot km\|\overrightarrow{e}\|_1 & 0 < t \leq d \\ \sum_{s=1}^{d-1} \|A\|_1^s \cdot km\|\overrightarrow{e}\|_1 & t > d \end{cases} \\ &\leq \sum_{s=1}^{d-1} \|A\|_1^s \cdot nkm \end{aligned}$$

Since $\|A\|_1 < 1$, the theorem is proven. □

Theorem 4 indicates that there is an upper bound of the number of agents which is decided by the size of the network, the connectivity of the network, the number of requests received, the number of agents generated per request, and the life-span limit of the agents. Thus, the total number of agents running in the network will not increase infinitely with time t, we can control the total number of agents by tuning relevant parameters.

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 in the jth host satisfies the following inequality:*

$$p_j(t) \leq km + \frac{n - \sigma_n}{n(\sigma_n - 1)}(d-1)(d_j - 1)km \quad (6)$$

Proof See Appendix. □ The significance of Theorem 5 can be visualized from Figure 3.

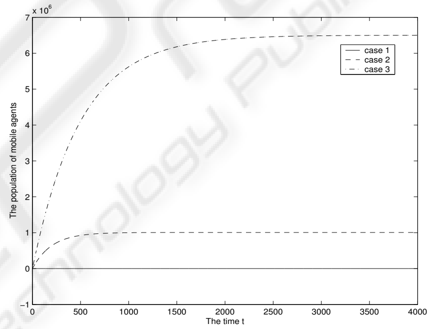

Figure 3: Variety of $p_j(t)$ with t.

Similarly, we still consider the following three cases in Figure 2 for explanation:

- Case 1: $n = 10000$, $\sigma_1 = \sigma_n = n$;
- Case 2: $n = 10000$, $\sigma_1 = 100$, $\sigma_n = 50$;
- Case 3: $n = 10000$, $\sigma_1 = 100$, $\sigma_n = 10$.

Once a request is received from a server in a complete connected network, the server knows the address of the corresponding destination and need not generate any mobile agents for routing. Therefore, the population of mobile agents in Case 1 equals to a constant 0. In both case 2 and case 3, $p_j(t)$ is a monotonically increasing function on time t. The connectivity of the network affects the increase of the population of mobile agents. It also can be seen that $p_j(t)$ increases slower and slower with time goes, and it will never exceed finite upper bound, as we have proved.

It is easy to understand that the term $(1 - d_j/n)p_j(t-1)/d_j$ in Equation (3) indicates the number of agents moving out from the jth host to each of its neighbor hosts at time t. Thus, from Theorem 5, we can further estimated it as follows:

Corollary 2 *The number of agents moving out from the jth host at time t, denoted by $f_j(t)$, satisfies:*

$$f_j(t) \leq \frac{n-\sigma_n}{n(\sigma_n-1)}(d-1)km \tag{7}$$

Proof As shown in Theorem 5, we have the following inequality:

$$\begin{aligned} f_j(t) &= \left(1-\frac{d_j}{n}\right)\frac{p_j(t-1)}{d_j-1} \\ &\leq km + \frac{1}{\sigma_n-1}km\sum_{i=1}^{t-2}\left(1-\frac{\sigma_n}{n}\right)^i \\ &\quad -\frac{\sigma_n}{n}km\cdot\sum_{i=0}^{t-2}\left(1-\frac{\sigma_n}{n}\right)^i \\ &\quad -\left(1-\frac{1}{\sigma_n-1}\right)km\left(1-\frac{\sigma_n}{n}\right)^{t-1} \\ &= \frac{n-\sigma_n}{n(\sigma_n-1)}km\sum_{i=0}^{t-2}\left(1-\frac{\sigma_n}{n}\right)^i \\ &\leq \frac{n-\sigma_n}{n(\sigma_n-1)}(d-1)km \end{aligned}$$

By 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 applying mobile agents in network routing, and then presented some analysis on both the probability of success and the population distribution of mobile agents. The parameters we analyzed include 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. Our results showed that these parameters 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. The main analytical results given in this paper are summarized in Table 1.

ACKNOWLEDGEMENT

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

PROOF OF THEOREM 5

Proof First, we prove that

$$\begin{aligned} p_j(t) &\leq d_j km + \frac{d_j-1}{\sigma_n-1}km\cdot\sum_{i=1}^{t-2}\left(1-\frac{\sigma_n}{n}\right)^i \\ &\quad -\frac{\sigma_n}{n}(d_j-1)km\cdot\sum_{i=0}^{t-2}\left(1-\frac{\sigma_n}{n}\right)^i \\ &\quad -\left(1-\frac{1}{\sigma_n-1}\right)(d_j-1)km\left(1-\frac{\sigma_n}{n}\right)^{t-1} \end{aligned}$$

Assume that at time $t=0$, there is no agent running in the network, that is, $\overrightarrow{p}(0)=(0,0,\cdots,0)^T$, and that each time the number of requests keyed in one host is m, k is the number of agents generated per request. Then, we have

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

We assume that

$$\begin{aligned} p_j(t-1) &\leq d_j km + \frac{d_j-1}{\sigma_n-1}km\cdot\sum_{i=1}^{t-3}\left(1-\frac{\sigma_n}{n}\right)^i \\ &\quad -\frac{\sigma_n}{n}(d_j-1)km\cdot\sum_{i=0}^{t-3}\left(1-\frac{\sigma_n}{n}\right)^i \\ &\quad -\left(1-\frac{1}{\sigma_n-1}\right)(d_j-1)km\left(1-\frac{\sigma_n}{n}\right)^{t-2} \end{aligned}$$

Table 1: Summary of the Main Results Given in This Paper.

No.	Result			Remark
1	$p(d)$	$\leq$	$\frac{\sigma_1}{n}\left(1-\frac{\sigma_1}{n}\right)^d$	Prob.
2	$P(d)$	$\leq$	$\frac{\sigma_1(n-\sigma_n)^2}{n^2\sigma_n}\left[1-\left(1-\frac{\sigma_n}{n}\right)^{d-1}\right]$	Prob.
3	P_s	$\leq$	$\frac{1}{\sqrt{2\pi s}[1-kP(d)/(s+1)]}$	Prob.
4	$\overrightarrow{p}(t)$	=	$km\overrightarrow{e}+A\overrightarrow{p}(t-1)$	Prob.
5	$\Vert\overrightarrow{p}(t)\Vert_1$	$\leq$	$(n-\sigma_n)(d-1)km$	Prob.
6	$p_j(t)$	$\leq$	$km+\frac{n-\sigma_n}{n(\sigma_n-1)}(d-1)(d_j-1)km$	Prob.
7	$f_j(t)$	$\leq$	$\frac{n-\sigma_n}{n(\sigma_n-1)}(d-1)km$	Prob.

then at time t, we have

$$
\begin{aligned}
p_j(t) &= km + \sum_{i\in NB(j)}\left(1-\frac{d_j}{n}\right)\frac{p_i(t-1)}{d_j} \\
&\leq d_j km + \left(1-\frac{\sigma_n}{n}\right)\frac{1}{\sigma_n-1}(d_{j-1})km \\
&\quad + \frac{1}{\sigma_n-1}(d_j-1)km\sum_{i=2}^{t-2}\left(1-\frac{\sigma_n}{n}\right)^i \\
&\quad - \frac{\sigma_n}{n}(d_j-1)km \\
&\quad - \frac{\sigma_n}{n}(d_j-1)km\cdot\sum_{i=1}^{t-2}\left(1-\frac{\sigma_n}{n}\right)^i \\
&\quad - \left(1-\frac{1}{\sigma_n-1}\right)(d_j-1)km\left(1-\frac{\sigma_n}{n}\right)^{t-1} \\
&\leq d_j km + \frac{d_j-1}{\sigma_n-1}km\cdot\sum_{i=1}^{t-2}\left(1-\frac{\sigma_n}{n}\right)^i \\
&\quad - \frac{\sigma_n}{n}(d_j-1)km\cdot\sum_{i=0}^{t-2}\left(1-\frac{\sigma_n}{n}\right)^i \\
&\quad - \left(1-\frac{1}{\sigma_n-1}\right)(d_j-1)km\left(1-\frac{\sigma_n}{n}\right)^{t-1}
\end{aligned}
$$

Thus, the first result is proven.

$$
\begin{aligned}
p_j(t) &\leq d_j km + \frac{d_j-1}{\sigma_n-1}km\cdot\sum_{i=1}^{t-2}\left(1-\frac{\sigma_n}{n}\right)^i \\
&\quad - \frac{\sigma_n}{n}(d_j-1)km\cdot\sum_{i=0}^{t-2}\left(1-\frac{\sigma_n}{n}\right)^i \\
&\quad - \left(1-\frac{1}{\sigma_n-1}\right)(d_j-1)km\left(1-\frac{\sigma_n}{n}\right)^{t-1} \\
&= km + \frac{n-\sigma_n}{n(\sigma_n-1)}(d_j-1)km\cdot\frac{1-\left(1-\frac{\sigma_n}{n}\right)^{t-1}}{1-\left(1-\frac{\sigma_n}{n}\right)} \\
&\leq km + \frac{n-\sigma_n}{n(\sigma_n-1)}(d_j-1)km\cdot\sum_{i=0}^{t-2}\left(1-\frac{\sigma_n}{n}\right)^i
\end{aligned}
$$

Since $p_j(t) \leq p_j(d)$ when $t \leq d$, and $p_j(t) = p_j(d)$ when $t \geq d$, we have

$$
p_j(t) \leq km + \frac{n-\sigma_n}{n(\sigma_n-1)}(d_j-1)km\cdot\sum_{i=0}^{d-2}\left(1-\frac{\sigma_n}{n}\right)^i
$$

Due to $1-\frac{\sigma_n}{n} \leq 1$, we have

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

Hence, the theorem is proven. □

REFERENCES

Caro, G. and Dorigo, M. (1998a). *AntNet: Distributed Stigmergetic Control for Communications Networks*. Journal of Artificial Intelligence Research, Vol. 9, pp. 317-365.

Caro, G. and Dorigo, M. (1998b). Mobile agents for adaptive routing. In *Thirty-First Hawaii International Conference on Systems*.

D.M. Piscitello, B. C. and Chapin, A. (1993). *Introduction to Routing*. http:// www.corecom.com /html /OSNconnexions.html.

Garey, M. and Johnson, D. (1979). *Ant Algorithms and Stigmergy*. Computers and Intractability: A Guide to the Theory of NP-Completeness, New York: Freeman.

J. Sum, H. Shen, C. L. and Young, G. (2003). *Analysis on a Mobile Agent-Based Ant routing Algorithm for Network Routing and Management*. IEEE Transaction on Paralell and Distributed Systems, Vol. 14, No. 3, pp. 193-202.

K. Curran, D. Woods, N. M. and Bradley, C. (2003). *The Effects of Badly Behaved Routers on Internet Congestion*. International Journal of Network Management, Vol. 13, pp. 83-94.

Kim, S. and Robertazzi, T. (2000). *Mobile Agent Modelling*. SUNY at Stony Brook Technical Report 786.

Kotz, D. and Gray, R. (1999). *Mobile Agents and the Future of the Internet*. ACM Operating Systems Review, Vol. 33(3), pp. 7-13.

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

M. Dorigo, E. Bonabeau, G. T. (2000). *Ant Algorithms and Stigmergy*. Future Generation Computer Systems, Vol. 16, No. 9, pp. 851-871.

Schoder, D. and Eymann, T. (2000). *The Real Challenges of Mobile Agents*. Communications of the ACM, Vol. 43, No. 6, pp. 111-112.

Wagner, C. and Turban, E. (2002). *Are Intelligent E-commerce Agents Partners or Predators?* Communications of the ACM, Vol. 45, No. 5, pp. 84-90.

White, T. (1997). *Routing with Swarm Intelligence*. Technical Report SCE-97-15, Systems and Computer Engineering, Carleton University.