Available online at www.sciencedirect.com
SCIENCE DIRECT
Microprocessors and Microsystems 29 (2005) 289–297
MICROPROCESSORS AND MICROSYSTEMS
www.elsevier.com/locate/micpro

A novel defense model for dynamic topology network based on mobile agent

Y.C. Jiang[a,*], Z.Y. Xia[b], S.Y. Zhang[a]

[a]Department of Computing and Information Technology, Fudan University. Shanghai 200433, People's Republic of China
[b]Department of Computer, Nanjing University of Aeronautics and Astronautics, Nanjing 210043, People's Republic of China

Received 19 January 2004; revised 17 August 2004; accepted 31 October 2004
Available online 23 November 2004

Abstract

It is common for the network topology to change during its operation, which demands that the network defense system adapt itself for the current topology. Aiming at such need, this paper provides a novel defense model for the dynamic topology network, which includes three modules: *network topology discovery*, *adaptive agents re-configuration mechanism* and *active defense*. The model is based on mobile agent technology, and contains two kinds of agents: *topology discovery agent* and *defense* agent. The model uses topology discovery agents to actively probe the current network topology and encodes it. Then the adaptive re-configuration mechanism of the model implements the distribution and migration of the defense agents according to the current topology. Thus, the re-configured defense agents provide active defense for the network. The whole model emerges with the Markov property, which is also analyzed in the paper.

Keywords: Dynamic topology network; Network security; Network defense; Mobile agent

1. Introduction

In the open network, nodes can take part in or depart from the network at any time. There is an emerging phenomenon that network topology is changed during the operation. The Ad-Hoc network is one of the typical examples of dynamic topology networks. Obviously, the security of the dynamic topology network is more serious than that of static network, and the network defense system should adapt for the topology dynamics.

Even though there are many network defense technologies nowadays, they often can only be used on fixed topology in practice. When the network topology is transformed, there is a need to make many manual reconfigurations for the existing network defense system, such as monitoring place, security mechanism, strategy, etc. Therefore, the existing network defense system often cannot function well when the network topology is transformed, and a network defense system aiming at one kind of topology does not work well with another different topology. Therefore, it is crucial to develop a network defense model that can adapt itself to the dynamic topology.

There are existing projects engaged in the adaptation of network defense technology, such as JAM [1], GASSATA [2], AAFID [3], JPA [4], and MAIDS [5]. Although these projects made contribution to the adaptation of network defense technology, and the intrusion detection technology based on mobile agent [7,11] can achieve topology adaptation ability in some degree, all of these related works do not emphasize the adaptation for dynamic network topology and lack systemic research. In order to solve such problem, we first presented an original model that can adjust its agent resource according to the network topology [6]. On the base of our original work in [6], now we present a whole defense model for dynamic topology network based on mobile agent technology. In our research, we want to make an attempt at an explorative study of the dynamic topology network defense.

* Corresponding author. Address: Center of Networking, Information Engineering, Room 409, Yifu Building, Fudan University, 200433 Shanghai, People's Republic of China. Tel.: +86 2165643235; fax: +86 2165647894.
E-mail address: jiangyichuan@yahoo.com.cn (Y.C. Jiang).

0141-9331/$ - see front matter © 2004 Elsevier B.V. All rights reserved.
doi:10.1016/j.micpro.2004.10.007

The rest of the paper is organized as the follows. Section 2 provides the basic architecture. Section 3 addresses the network topology discovery and encoded representation. Section 4 describes the agent reconfiguration mechanism; Section 5 addresses the defense agents briefly. Section 6 analyzes the Markov property of the whole model. Section 7 describes a simulation experiment, and sums up the whole project and proposes the future research direction.

2. Basic architecture

The model described in this paper is based on mobile agent technology. There are two kinds of agents in the system: topology discovery agent and defense agent. Aiming at the dynamic topology, we need first discover the current network topology timely and correctly based on *topology discovery agent*. After discovering the topology, the system makes adaptive reconfiguration to the *defense agents*. The reconfigured defense agents make active defense for the network.

The model includes three modules:

(1) *Network topology discovery*. This module uses agent-based active probing technology to discover the current topology of the protected network and encodes it.
(2) *Adaptive agents re-configuration mechanism*. According to the discovered topology, this module reconfigures the defense agents. This is divided into two sub-parts: one is the initial distribution of agents for network topology; the other is the adaptive migration of agents.
(3) *Active defense*. After distribution and migration, the defense agents implement active defense for the network.

The active defense model architecture for dynamic topology is shown in Fig. 1.

The reconfiguration mechanism adopts a multi-agents architecture with a monitor center. It integrates with the network topology discovery module. The monitor centers of both are located on the same management station. The adaptive reconfiguration mechanism uses genetic algorithms and ant algorithms to make adaptive reconfiguration for achieving the optimal distribution of defense agents according to the current network topology.

In the model, the defense agents include: *intrusion sensor agent*, *intrusion detection agent*, *tracing agent and recovery agent*. Each kind of agents takes care of their respective task.

3. Network topology discovery and encoding

3.1. Topology discovery

On the base of topology discovery method proposed by Hwa-Lin [8] and other relative works (e.g., [9]), we adopt active network topology probing technology based on mobile agent, shown in Fig. 2.

Fig. 2 can be explained as follows: on the Management Station there is DMA1, DMA1 firstly arrives a node N which is never probed by other DMAs. It returns RMA1 that contains the adjacent information of N to the Management Station. Then DMA1 spawns a new DMA for each adjacent node of the node N and terminates itself afterwards. The new DMA goes to the adjacent node, if it discovers that the adjacent node is never probed then it returns a RMA to the Management Station, or else it terminates itself. After the whole topology information returns to the Management Station, the EC encodes it.

3.2. Topology encoded representation

Given that there are n nodes, the complete graph constructed by these nodes has $n(n-1)/2$ edges. According to the number of edges, the length of code is $n(n-1)/2$. Referring to the actual topology, the bit of the code can be 0 if the edge of the graph does not correspond to the one of

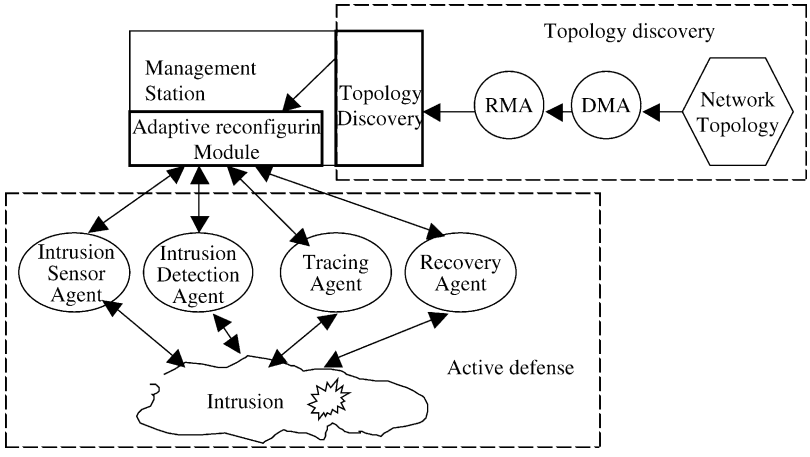

Fig. 1. The architecture of the network defense model for dynamic topology.

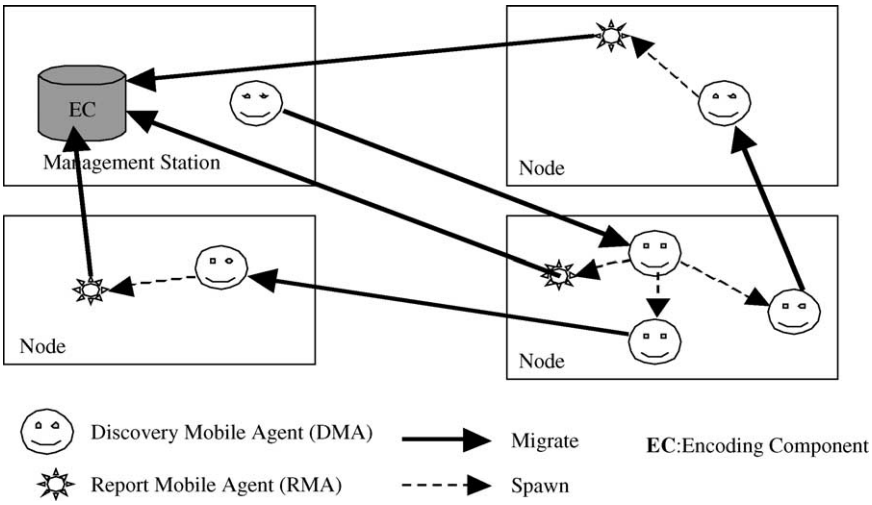

Fig. 2. Architecture of network topology discovery.

the actual topology, or be 1 if the edge of the graph corresponds to the one of the actual topology. For example, a complete graph with four nodes is shown in Fig. 3, the number of each edge is: 1(1,2), 2(1,3), 3(1,4), 4(2,3), 5(2,4), 6(3,4). And the actual network is shown in Fig. 4, which is composed of edges {1,3,4}, therefore we can encode the actual topology as: {101100}.

4. Agent adaptive reconfiguration mechanism

When the network topology is transformed, we need reconfigure the agents in the defense system. Now we apply genetic algorithms and ant algorithms into the reconfiguration mechanism, and propose a module which can adaptively reconfigure defense agents according to the current network topology.

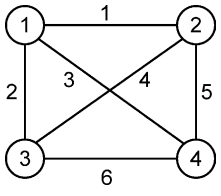

Fig. 3. A complete graph with four nodes.

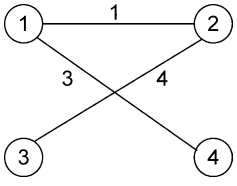

Fig. 4. The actual topology.

4.1. Apply genetic algorithm to implement initial agents distribution

In the model, when there is intrusion in the network, we need first distribute the agents in the network accordingly to achieve optimal distribution of defense agents according to current network topology, intrusion information and former agents distribution. Genetic Algorithms (GAs) are adaptive heuristic search algorithm based on the evolutionary ideas of genetics and natural selection [10]. We use the genetic algorithm to implement the optimal distribution of defense agents. It is well known that the key factors of genetic algorithm are *encoding*, *fitness function*, *production of initial population* and *genetic operation*. Next we will address them in detail.

(1) Encoding

We adopt subsection code where chromosome is parted into three sections of gene. The first section of gene denotes network topology, the second denotes agent distribution state, and the third one denotes intrusion information. Therefore, the chromosome is shown as follows:

$$a_1, a_2, \ldots, a_i, \ldots, a_{n(n-1)/2}$$

$$|I_{11}I_{12}I_{13}I_{14}, I_{21}I_{22}I_{23}I_{24}, \ldots, I_{i1}I_{i2}I_{i3}I_{i4}, \ldots, I_{n1}I_{n2}I_{n3}I_{n4}$$

$$|b_1, b_2, \ldots, b_i, \ldots, b_n$$

where n is the number of nodes in the network.

Among those the first section is addressed in Section 3.2. The length of this section is $n(n-1)/2$.

The second section '$I_{i1}I_{i2}I_{i3}I_{i4}$' shows that on the node i there are No. 1 agents (intrusion sensor agent) with the amount of I_{i1}, No. 2 agents (intrusion detection agent) with the amount of I_{i2}, No. 3 agents (tracing agent) with the amount of I_{i3}, No. 4 agents (disaster

recovery agent) with the amount of I_{i4}. The length of this section is $4n$.

The third section is binary that denotes which node suffered from abnormal activity. $b_i=1$ if node i is suffered from abnormal activity, or else $b_i=0$. The length of this section is n.

Therefore, the total length of the chromosome is shown as follows:

$$n(n-1)/2+4n+n=\frac{n^2}{2}+\frac{9}{2}n \tag{1}$$

(2) *Design the Fitness Function*

When agent i moves from the now location to the destination location, the migration cost includes resource and time cost.

We define the migration cost function of agent i as follows

$$\text{Cost}_i = h(\sigma_1 C_t + \sigma_2 C_r) \tag{2}$$

where C_t is the time cost when agent moves from a node to its adjacent one, C_r is the system resource cost when agent moves from a node to its adjacent one. h is the hops when agent moves from the now location to the destination location. σ_1 and σ_2 are the weight of C_t and C_r. The adaptive act of agent i should make the cost function be minimum.

In genetic algorithms, we often use a non-negative real number to reflect the fitness ability of individual. In order to adjust to the character of genetic algorithms and combining our code and the above cost function, we can define fitness function as follows

$$F^t = F_0^t - \sum_{i=1}^{N} \text{Cost}_i \tag{3}$$

where N is the amount of mobile agents. F_0^t is a positive constant, which changes with the problem size and is used to ensure individual fitness F^t always be non-negative.

(3) Production of Initial Population

In term of the encoding method, we produce initial population with the individual length of $(n^2/2)+(9/2)n$. Among those the first $n(n-1)/2$ bits of gene is binary; the middle $4n$ bits of gene is natural number or 0; the last n bits of gene is binary.

(4) Genetic Operator

Selection: According to (3), we can compute the fitness of individual and select the individual with high fitness value. Crossover: we use crossover operation to produce new individual; Mutation: we also use mutation to produce more robust individual.

Then, through the genetic algorithm operation, we can get the optimal initial distribution of defense agents.

4.2. Apply ant algorithms to implement agent migration

In the defense process, the defense agents should migrate in the network. For effective migration, we apply ant algorithm to implement agent migration.

4.2.1. Residence factor of agent

Firstly we define an array $A[k][i]$ to denote that the number of successful defense while agent k locates at node i After a network defense system is initially installed in a network, $A[k][i]$ is zero. Once the network defense system makes defense successfully, we add $A[k][i]$ by 1, or else we subtract $A[k][i]$ by 1. However, $A[k][i]$ cannot be less than zero.

Definition 1. Residence factor of agent k at node i is defined as follows:

$$\text{res}_k(i) = \text{In}(A[k][i]+1) \tag{4}$$

From (4), we can see that when $A[k][i]$ is zero, then the residence factor of agent k at node i is zero.

The more $\text{res}_k(i)$ is, the better agent k can function at node i, therefore in the new distribution, it has a better chance to migrate to node i.

4.2.2. Apply ant algorithms to realize the agent one-hop migration in the defense progress

Ant algorithm is collective intelligence that studies how the actions and inter-relations of a set of simple agents (for example, bees, ants, etc.) in carrying out global objectives of the system where these agents are immersed [12,13]. The ant algorithm was firstly used to solve the TSP (Travel Salesman Problem). According to the ant algorithm, the ant transition rule is mainly decided by the pheromone left by other ants on the path and heuristic value. In the TSP, the shorter a path is, the more the number of ants that go through the path, then the more the pheromone left by ants. And ants are prone to select the path with more pheromone to travel so as to toward the optimal result.

When the defense system provides defense, agents need to migrate. If agent k want to migrate from i to j, we should consider two factors: one is the pheromone on the path (i,j), the more the number of agents that go through path (i,j), the more pheromone there is; the other is the comparison between $\text{res}_k(i)$ and $\text{res}_k(j)$.

According to the ant algorithm [12], the transition rule of agent is shown as follows

$$p_{ij}(k) = \begin{cases} \dfrac{[\tau(i,j)]^\alpha \times [\eta(i,j)]^\beta}{\sum_{u \in ADJ_k(i)} [\tau(i,u)]^\alpha \times [\eta(i,u)]^\beta} & \text{if } j \in \text{ADJ}_k(i) \text{ and } p_{ij}(k) \geq 0 \\ 0 & \text{otherwise} \end{cases} \tag{5}$$

where $p_{ij}(k)$ denotes the probability that agent k migrate from i to j; $\text{ADJ}_k(i)$ denotes the adjacent nodes of i; $\tau(i,j)$ denotes the pheromone on the path (i,j); $\eta(i,j)$ is a heuristic value; α and β are parameters to control the relative

influence importance between pheromone and heuristic value on agent migration probability.

Pheromone update formula is shown as follows

$$\tau_{ij}(n+1) = \rho\tau_{ij}(n) + \Delta\tau_{ij} \tag{6}$$

where ρ is a parameter, $(1-\rho)$ denotes the waning degree of pheromone from time n to time $n+1$.

$$\Delta\tau_{ij} = \sum_{x} \Delta\tau_{ij}^{x} \tag{7}$$

In (7), x denotes the number of agents; $\Delta\tau_{ij}^{x}$ denotes the pheromone left by agent x on path (i,j)

$$\Delta\tau_{ij}^{x} = \begin{cases} Q/(\sigma_1 d_{ij} + \sigma_2 m_{ij}^{x}), & \text{if agent } x \text{ passes } (i,j) \text{ in this migration process} \\ 0, & \text{otherwise} \end{cases} \tag{8}$$

where Q is a constant which can be got by experiment; d_{ij} denotes the distance between i and j; m_{ij}^{x} denotes the migration cost of agent x from i to j, more detail can be seen in [14]; σ_1 and σ_2 are used to control the relative importance between d_{ij} and m_{ij}^{x}.

In this paper, considering the actual situation of network defense system, we design the heuristic value as follows:

$$\eta_k(i,j) = \mathrm{res}_k(j) - \mathrm{res}_k(i) + C_k \tag{9}$$

C_k is a constant number which is decided by experiment. We can see that only if $\mathrm{res}_k(i) - \mathrm{res}_k(j)$ is more than C_k, then $\eta_k(i,j)$ is negative, therefore migration probability is zero.

According to (6), (9) and (5), then the migration probability of agent k from i to j at time $n+1$ is shown as follows:

$$p_{ij}(k) = \begin{cases} \dfrac{[\rho\tau_{ij}(n) + \Delta\tau_{ij}]^{\alpha} \times [\mathrm{res}_k(j) - \mathrm{res}_k(i) + C_k]^{\beta}}{\sum_{u \in \mathrm{ADJ}_k(i)} [\rho\tau_{iu}(n) + \Delta\tau_{iu}]^{\alpha} \times [\mathrm{res}_k(u) - \mathrm{res}_k(i) + C_k]^{\beta}}, & \text{if } (j \in \mathrm{ADJ}_k(i)) \text{ and } (p_{ij}(k) \geq 0) \text{ and (the denominator} \neq 0) \\ 0, & \text{otherwise} \end{cases} \tag{10}$$

4.2.3. Implement the agent multi-hops migration based on probability theory

In this section, for simplification, we use 'agent' to represent 'agent k', and 'p_{ij}' to represent '$p_{ij}(k)$'.

From Section 4.2.2 we can see that when an agent migrates, its migration probability is only influenced by the pheromone and residence factor produced by last agents' distribution, and not by the previous ones before the last distribution.

We introduce X_n to denote the process of agent migration, $X_n = i$ denotes that agent migrates to node i at the nth migration. Obviously, for any nodes series: $i_0, i_1, \ldots, i_{n-1}, i, j$ and $n \geq 0$, the stochastic process $\{X_n, n = 0,1,\ldots\}$ obeys the following equation:

$$\begin{aligned} p\{X_{n+1} = j | X_0 = i_0, \ldots, X_{n-1} = i_{n-1}, X_n = i\} \\ = p\{X_{n+1} = j | X_n = i\} \end{aligned} \tag{11}$$

From (11), we can conclude that the migration process of agent possesses Markov property, so X_n is a discrete-time Markov Chain. Therefore, we can use Markov Chain theory to analyze the property of agent migration.

We let $p_{ij}^{n,n+1} = p\{X_{n+1} = j | X_n = i\}$, which denotes the conditional probability that agent migrate to node j at the $(n+1)$th migration while the agent locates at node i at the nth migration.

When we analyze the migration process of agent, we only need to consider the interval time between the migrations, and not the start time of the first migration of the agent. Therefore, we can conclude:

$$p\{X_{n+1} = j | X_n = i\} = p_{ij}^{n,n+1} \equiv p_{ij}, \quad n \geq 1 \tag{12}$$

Therefore, the migration process of agent has stable probability and has homogeneous Markov property, and, $\{X_n, n \geq 0\}$ is a homogeneous Markov Chain.

Since the migration process of agent is not negative, the agent either move to a new location or keep residing on current location, therefore we can conclude:

$$p_{ij} \geq 0 \quad \text{and} \quad \sum_{j} p_{ij} = 1, \quad i.j \geq 0 \tag{13}$$

Let the number of nodes in the network be $n+1$, we can define the migration probability matrix of agent: $P = [p_{ij}]$, i.e.:

$$P = \begin{bmatrix} p_{00} & p_{01} & p_{02} & \cdots \\ p_{10} & p_{11} & p_{12} & \cdots \\ & \vdots & & \\ p_{n0} & p_{n1} & p_{n2} & \cdots \end{bmatrix} \tag{14}$$

The ith row of the matrix denotes the probability that the agent on node i migrate to other nodes.

The probability that agent on node i can arrive at node j after n-hops migration is denoted as: $p_{ij}^{(n)} = p\{X_{m+n} = j | X_m = i\}$; and the n-hops migration matrix of agent is denoted as: $P^{(n)} = ||p_{ij}^{(n)}||$.

Theorem 1. The n-hops migration matrix of agent has the following property:

$$p_{ij}^{(n)} = \sum_{k=0}^{\infty} p_{ik} p_{kj}^{(n-1)} \tag{15}$$

We can assume that $p_{ii}^{(0)} = 1$, and if $j \neq i$ then $p_{ij}^{(0)} = 0$.

Theorem 1 is concluded from the principle of stochastic process, and the proof testifying process can be seen in [15].

From Theorem 1, we can see: if an agent want to move from i to j after n-hops migration, the agent should move to k at the first hop migration, and then move from k to j after $(n-1)$-hops migration. The k can also be i or j themselves.

Let the probability distribution of agent at the start time (we can call it as *Initial Probability Distribution*) is:

$$p_i^{(0)} = p\{X_0 = i\}, \quad i = 0, 1, 2, \ldots \tag{16}$$

And the probability distribution of agent at the nth migration is:

$$p_i^{(n)} = p\{X_n = i\}, \quad i = 0, 1, 2, \ldots \tag{17}$$

According to the *Formula of Total Probability* [15], we can conclude:

$$p\{X_n = j\} = \sum_i p\{X_0 = i\} p\{X_n = j | X_0 = i\} \tag{18}$$

So

$$p_j^{(n)} = \sum_i p_i^{(0)} p_{ij}^{(n)} \tag{19}$$

Therefore, we can conclude:

Theorem 2. In the migration process of agent, the n-hops migration probability can be computed by one-hop migration probability.

Definition 2. *Reach-ability*: if there exists $n \geq 0$ and $p_{ij}^{(n)} > 0$, then we say that agent can reach j from i.

Obviously, the reach-ability of agent has the following property: if $i \to j$ and $j \to k$, then $i \to k$.

Therefore, on the base of Markov property of agent migration, we can compute n-hops migration probability of agent to select a best path to move. If the agents migrate along their best paths, they can function well and minimize the cost.

5. Using defense agents to make active defense

After adaptive reconfiguration, the agents are re-distributed in the network optimally to defend the network. As mentioned above, there are four kinds of defense agents: *intrusion sensor agent*, *intrusion detection agent*, *tracing agent*, and *recovery agent*.

Intrusion sensor agents often reside on the hosts. They check the network flow data and host logs to discover abnormal activities; intrusion detection agents make intrusion detection on the location where intrusion sensor agents discover abnormal activities; tracing agents trace the intrusion path in the network and find out the exact location where the network suffered from intrusion first; recovery agents recover the contaminated data according to the system log.

In the system, the coordination among agents adopts Direct Coordination Model. Agent can pass message to the coordinated agent directly. Therefore, the agent that wants to pass message should know the identifier of the agent that will receive message. So every agent should has a unique identifier in the system and the management station should know the exact location of every agent.

6. Analyses for the Markov property of the whole model

In Section 4.2.3, we analyze the Markov property of agent migration, and now we analyze the Markov property of the whole network defense model.

We present a model for test, the whole test model is shown in Fig. 5.

Definition 3. [16] A Stochastic Petri Net (SPN) is a quintuple: SPN$=(P,T;F,M_0,\lambda)$, where:

(1) PN$=(P,T;F,M_0)$ is a Petri Net;
(2) $\lambda=\{\lambda_1,\lambda_2,\ldots,\lambda_n\}$, $n=|T|$, $\lambda_i>0$ is the average run rate of transition t_i, which conforms the exponential distribution.

We can use Stochastic Petri Net to formalize the network defense test model in Fig. 5, shown in Fig. 6 (Table 1).

The average rate of t_4 is the same as the one of t_5; the average rate of t_6 is the same as the one of t_7. For all of the transitions, the corresponding rates that conform the exponential distribution are: $\{\lambda_1,\lambda_2,\lambda_3,\lambda_4,\lambda_4,\lambda_5,\lambda_5\}$.

We can set the token of initial place as 1, and set the token of other places as 0. Under some situation, the output

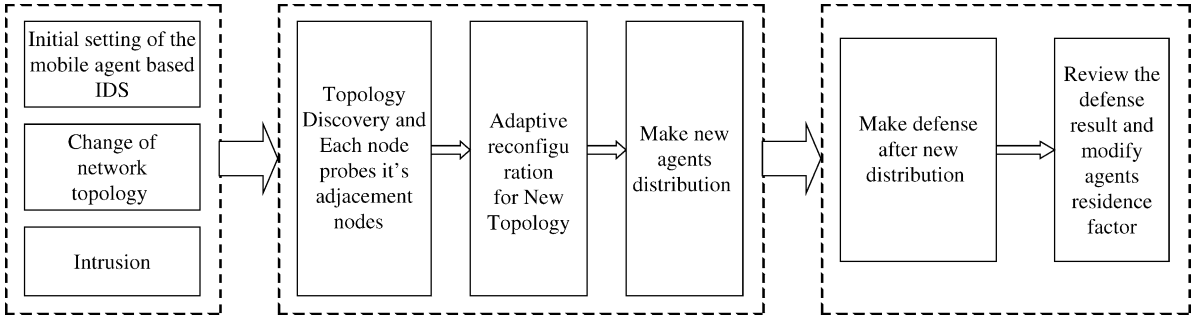

Fig. 5. The test defense model for dynamic topology.

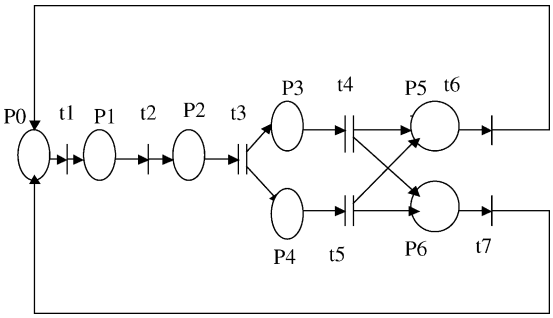

Fig. 6. The Petri Net formalization of the model in Fig. 5.

of Petri Net model can be different according to different conditions. When a transition takes place, of all the output places set only one place's mark can change, so the concept of *selected place transition* was introduced [17]. In Fig. 6, ‖ denotes that when a transition takes place, among the output places we only select a place's mark to change. Therefore, **only** P_3 **or** P_4 can be changed when t_3 takes place; and **only** P_5 **or** P_6 can be changed when t_4 takes place.

In [16], it has been proved that a SPN is congruent to a continuous-time Markov Chain. Every mark of SPN can be mapped to a state of Markov Chain, and the reachability graph is congruent to a state space of a Markov Chain. The reachability graph of our network defense model is shown in Fig. 7. By replacing t_i with λ_i, we can get the Markov Chain for the model, shown in Fig. 8.

We firstly define a transfer matrix $Q=[q_{ij}]$, $0\le i,j\le n-1$. q_{ij} $(i\neq j)$ can be gotten as follows: if there is an arc from M_i to M_j, then q_{ij} $(i\neq j)$ is equal to the rate λ on the arc, or else q_{ij} $(i\neq j)=0$. If $i=j$, then q_{ij} is the negative of the sum of the rate of the arcs out from M_i.

[16] Let the stable state probability of n states in Markov Chain be a row vector: $X=(x_0,x_1,\ldots,x_{n-1})$, then according to the property of Markov stochastic process we can

Table 1
The cutline in Fig. 6

Place	Description	Transition	Description
p_0	Initial state	t_1	Change the network topology
p_1	The network state after changed topology	t_2	Simulation intrusion
p_2	The network state while being intruded	t_3	Adaptive reconfiguration of defense agent
p_3	Agent migrates	t_4	Agents make defense for the network after migration
p_4	Agent keep stable	t_5	The stable agents make defense for the network
p_5	Network defense successfully	t_6	Add the residence factor with 1
p_6	Network defense unsuccessfully	t_7	Subtract 1 from the residence factor

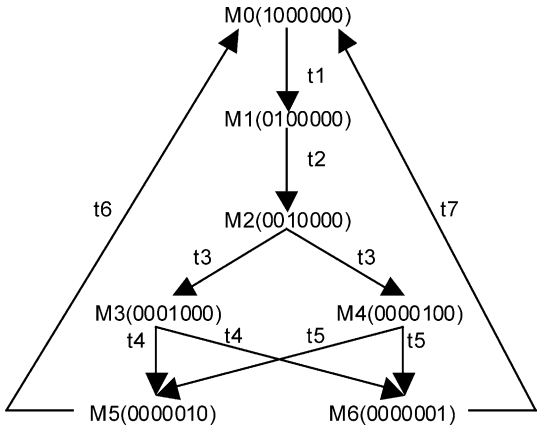

Fig. 7. The reachability graph of the SPN for the adaptive defense model.

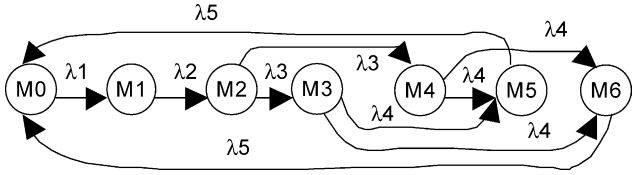

Fig. 8. The Markov chain for the adaptive defense model.

conclude:

$$\begin{cases} XQ = 0, \\ \sum_i x_i = 1, \quad 0 \le i \le n-1 \end{cases} \tag{20}$$

The Markov Chain transfer matrix of our model is shown as follows:

$$\begin{pmatrix} -\lambda_1 & \lambda_1 & 0 & 0 & 0 & 0 & 0 \\ 0 & -\lambda_2 & \lambda_2 & 0 & 0 & 0 & 0 \\ 0 & 0 & -2\lambda_3 & \lambda_3 & \lambda_3 & 0 & 0 \\ 0 & 0 & 0 & -2\lambda_4 & 0 & \lambda_4 & \lambda_4 \\ 0 & 0 & 0 & 0 & -2\lambda_4 & \lambda_4 & \lambda_4 \\ \lambda_5 & 0 & 0 & 0 & 0 & -\lambda_5 & 0 \\ \lambda_5 & 0 & 0 & 0 & 0 & 0 & -\lambda_5 \end{pmatrix} \tag{21}$$

From (20) and (21), we can conclude:

$$\begin{cases} -\lambda_1 x_0 + \lambda_5 x_5 + \lambda_5 x_6 = 0 \\ \lambda_1 x_0 - \lambda_2 x_1 = 0 \\ \lambda_2 x_1 - 2\lambda_3 x_2 = 0 \\ \lambda_3 x_2 - 2\lambda_4 x_3 = 0 \\ \lambda_3 x_2 - 2\lambda_4 x_4 = 0 \\ \lambda_4 x_3 + \lambda_4 x_4 - \lambda_5 x_5 = 0 \\ \lambda_4 x_3 + \lambda_4 x_4 - \lambda_5 x_6 = 0 \\ x_0 + x_1 + x_2 + x_3 + x_4 + x_5 + x_6 = 1 \end{cases} \tag{22}$$

Therefore, we can get the stable probability of every state, shown as (23)

$$\begin{cases} P\{M_0\} = x_0 = \dfrac{2\lambda_2\lambda_3\lambda_4\lambda_5}{2\lambda_2\lambda_3\lambda_4\lambda_5 + 2\lambda_1\lambda_3\lambda_4\lambda_5 + \lambda_1\lambda_2\lambda_4\lambda_5 + \lambda_1\lambda_2\lambda_3\lambda_5 + 2\lambda_1\lambda_2\lambda_3\lambda_4} \\ P\{M_1\} = x_1 = \dfrac{2\lambda_1\lambda_2\lambda_3\lambda_4\lambda_5}{2\lambda_2^2\lambda_3\lambda_4\lambda_5 + 2\lambda_1\lambda_2\lambda_3\lambda_4\lambda_5 + \lambda_1\lambda_2^2\lambda_4\lambda_5 + \lambda_1\lambda_2^2\lambda_3\lambda_5 + 2\lambda_1\lambda_2^2\lambda_3\lambda_4} \\ P\{M_2\} = x_2 = \dfrac{2\lambda_1\lambda_2\lambda_3\lambda_4\lambda_5}{4\lambda_2\lambda_3^2\lambda_4\lambda_5 + 4\lambda_1\lambda_3^2\lambda_4\lambda_5 + 2\lambda_1\lambda_2\lambda_3\lambda_4\lambda_5 + 2\lambda_1\lambda_2\lambda_3^2\lambda_5 + 4\lambda_1\lambda_2\lambda_3^2\lambda_4} \\ P\{M_3\} = x_3 = \dfrac{2\lambda_1\lambda_2\lambda_3\lambda_4\lambda_5}{8\lambda_2\lambda_3\lambda_4^2\lambda_5 + 8\lambda_1\lambda_3\lambda_4^2\lambda_5 + 4\lambda_1\lambda_2\lambda_4^2\lambda_5 + 4\lambda_1\lambda_2\lambda_3\lambda_4\lambda_5 + 8\lambda_1\lambda_2\lambda_3\lambda_4^2} \\ P\{M_4\} = x_4 = \dfrac{2\lambda_1\lambda_2\lambda_3\lambda_4\lambda_5}{8\lambda_2\lambda_3\lambda_4^2\lambda_5 + 8\lambda_1\lambda_3\lambda_4^2\lambda_5 + 4\lambda_1\lambda_2\lambda_4^2\lambda_5 + 4\lambda_1\lambda_2\lambda_3\lambda_4\lambda_5 + 8\lambda_1\lambda_2\lambda_3\lambda_4^2} \\ P\{M_5\} = x_5 = \dfrac{2\lambda_1\lambda_2\lambda_3\lambda_4\lambda_5}{4\lambda_2\lambda_3\lambda_4\lambda_5^2 + 4\lambda_1\lambda_3\lambda_4\lambda_5^2 + 2\lambda_1\lambda_2\lambda_4\lambda_5^2 + 2\lambda_1\lambda_2\lambda_3\lambda_5^2 + 4\lambda_1\lambda_2\lambda_3\lambda_4\lambda_5} \\ P\{M_6\} = x_6 = \dfrac{2\lambda_1\lambda_2\lambda_3\lambda_4\lambda_5}{4\lambda_2\lambda_3\lambda_4\lambda_5^2 + 4\lambda_1\lambda_3\lambda_4\lambda_5^2 + 2\lambda_1\lambda_2\lambda_4\lambda_5^2 + 2\lambda_1\lambda_2\lambda_3\lambda_5^2 + 4\lambda_1\lambda_2\lambda_3\lambda_4\lambda_5} \end{cases} \quad (23)$$

From all of above analyses, we can conclude: (1) of all the marks of the reachability graph, the max number is 1, so the model is bound and secure; (2) from the reachability graph, we can see that every transition can take place, so the model does not produce deadlock; (3) the sum of the marks of the model is constant, so the model is conservational; (4) the model has reach-ability.

Therefore, our model is feasible theoretically.

7. Experiments and conclusion

We have developed a network defense prototype system based on mobile agent technology. We embed the *topology-adapted network defense model* provided by this paper into the prototype system and make simulation test.

In our experiment, we make two kinds of tests: (1) Running the original prototype without the introduction of the model provided by this paper (Method 1); (2) using our topology-adapted network defense model (Method 2). We make comparison between the intrusion detection efficiencies of Method 1 and Method 2. The *intrusion detection efficiency* is defined as the proportion of the number of successful intrusion detection to the total number of simulation tests.

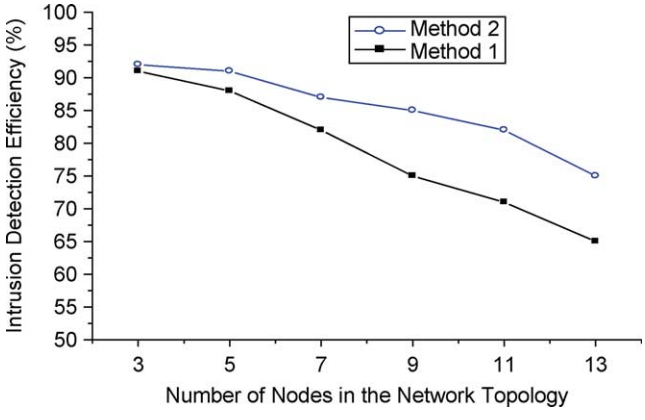

Fig. 9. The comparison between the two methods in the experiment.

In the simulation experiment, we adopt Expert, an Unix tool, to simulate the intrusion. In our experiment, we get different network topology by change the amount of network nodes from 3 to 13. In the network, we implement full inter-connection among the nodes.

From Fig. 9, we can see that: when the nodes number is 3, the contrast is not obvious, the reason is that since the topology is not so complex that our model does not show its advantage; however, with the increase of node number, then the network topology is more complex, the performance contrast between the two methods becomes obvious.

Therefore, the simulation result proves that our model is feasible, especially when the network topology is complex. Our model can adapt itself for the change of network topology.

This paper provides a defense model for dynamic topology network, and makes detail explanation for the architecture and principle of the model. In our research, we made an explorative attempt for the defense of dynamic topology network. However, now our work is still mainly theoretical and elementary. Our future task will focus on the further development of the system and achieving a network defense system that can be applied in real large-scale dynamic topology network.

References

[1] W. Lee, S.J. Stolfo, K.W. Mok, A data mining framework for building intrusion detection models, in: Proceeding of IEEE Symposium on Security and Privacy 1999, Berkeley, California, pp. 120–132.

[2] M.E. Ludoric, GASTATA. A genetic algorithm as an alternative tool for security audit trails analysis, 1998. URL: www.inf.tu-dresden.de/publications/1998/wv-1998-01.ps.gz

[3] J. Balasubramaniyan, J.O. Garcia-Fernandez, D. Isacoff, E.H. Spafford, D. Zamboni, An architecture for intrusion detection using

autonomous agents [Technical Report], Department of Computer Science, Purdue University; Coast TR 98-05, 1998.
[4] M. Asaka, S. Okazawa, A. Taguchi, S. Goco, A method of tracing intruders by use of mobile agents, in: Proceeding of INET99, San Jose, CA, USA, June 1999.
[5] S. Parikh, A Framework of System Integrator for MAIDS. [Report for the Degree of Master of Science], Iowa State University, Ames, IA, 2001.
[6] Y.C. Jian, Y.P. Zhong, S.Y. Zhang. A topology-adapted network defense model based on mobile agent. Grid and cooperative computintg. Lecture Notes in Computer Science. Vol. 3252. pp. 335–342. Springer-verlag Heidelberg, 2004.
[7] C.A. Carver, Jr., Adaptive agent-based intrusion response [PhD Thesis], Texas A&M University, Texas, 2001.
[8] H.-C. Lin, C.-H. Wang, Automatic topology discovery using mobile agents, in: Proceedings of the International Workshop on Agent Technologies over Internet Applications, September 26–28, 2001.
[9] B. Lowekamp, D.R. O'Hallaron, T.R. Gross, Topology discovery for large Ethernet networks, in: Proceeding of SIGCOMM'01. San Diego, California, August 27–31, 2001.
[10] J.H. Holland, Genetic algorithms, Scientific American 267 (1992) 60–78.
[11] M. Crosbie, G. Spafford, Defending a computer system using autonomous agents, in: Proceedings of the 18th National Information Security Conference. Baltimore, Maryland, October, 1995.
[12] A. Colorni, M. Dorigo, V. Maniezzo, Distributed optimization by ant colonies [A], in: Proceedings of the First European Conference Artificial Life [C], Pans, Elsevier, France, 1991, pp. 134–142.
[13] J. Aguilar, A general ant colony model to solve combinational optimization problems, Revista Colombiana de Computaction 2(1), pp. 7–18. 2001.
[14] W. Jansen, P. Mell, T. Kargiannis, D. Marks, Mobile agents in intrusion detection and response, in: Proceedings of the 12th Annual Canadian Information Technology Security Symposium, Ottawa, Canada, June, 2000.
[15] S.-Y. Hua, Stochastic Processes, South-East University Press, Nanjing, 1988. pp. 49–75 (in Chinese).
[16] L. Chuang, Computer Network and Computer System Performance Evaluation, Tsinghua University Press, Beijing, 2001. pp. 176–185 (in Chinese).
[17] W. He, H.-W. Wang, Petri-net-based modeling support, Journal of Huazhong University of Science and Technology 23 (5) (1995) 52–56 (in Chinese).

Y.C. Jiang was born in 1975. He received his MS degree in computer science from Northern Jiaotong University, China in 2002. He is currently a PhD candidate in computer science of the Department of Computing and Information Technology, Fudan University, China. His research interests include mobile agent system, network security, and artificial intelligence.

Z.Y. Xia was born in 1974. He received his MS degree in fuse technology from Nanjing University of Science and Technology in 1999, and received his PhD degree in computer science from Fudan University in 2004. He is currently a lecturer in the Department of Computer, Nanjing University of Aeronautics and Astronautics, China. His research interests include information security, mobile agent and active network.

S.Y. Zhang was born in 1950. He is now a professor and PhD supervisor, and also the director of the Center of Networking and Information Engineering, Fudan University, China. His research interests include network system, mobile agent system and network security.