Autonomic Network Applications Designed after Immunological Self-Regulatory Adaptation

Chonho Lee and Junichi Suzuki
Department of Computer Science
University of Massachusetts, Boston
{chonho and jxs} @ cs.umb.edu

Abstract—*As Internet applications have been rapidly increasing in complexity and scale, they are expected to be autonomous and adaptive to dynamic changes in the network. Based on the observation that various biological systems have already overcome these requirements, this paper describes a biologically-inspired framework, called iNet, to design autonomous and adaptive Internet applications. It is designed after the mechanisms behind how the immune system detects antigens (e.g., viruses), specifically produces antibodies to eliminate them, and self-regulates the production of antibodies against its anomaly (e.g., immunodeficiency and autoimmunity). iNet models a set of environment conditions (e.g., network traffic and resource availability) as an antigen and a behavior of applications (e.g., migration and reproduction) as an antibody. iNet allows each application to autonomously sense its surrounding environment conditions (i.e., an antigen) to evaluate whether it adapts well to the sensed conditions based on an evaluation policy, and if it does not, adaptively invoke a behavior (i.e., an antibody) suitable for the conditions. iNet also allows each application to dynamically configure its own evaluation policy so that it can trigger the behavior invocation at the right time. Simulation results show that iNet allows applications to autonomously adapt to changing environment conditions and to dynamically self-regulate the behavior invocation by configuring the evaluation policy when the evaluation fails.*

1. INTRODUCTION

Large-scale network applications such as data center applications and grid computing applications face several critical challenges, particularly autonomy and adaptability, as they have been increasing in complexity and scale[1]. They are expected to autonomously adapt to dynamic environmental changes in the network (e.g., workload surges and resource extinction) in order to improve user experience, expand applications' operational longevity and reduce maintenance cost [1, 2, 3, 4].

As inspiration for a new design paradigm for network applications, the authors of the paper observe that various biological systems have developed the mechanisms necessary to meet the above requirements (i.e., autonomy and adaptability) [5]. For example, bees act autonomously, influenced by local conditions and local interactions with other bees. A bee colony adapts to dynamic environmental conditions. When the amount of honey in a hive is low, many bees leave the hive to gather nectar from flowers. When the hive is full of honey, bees rest in the hive. Based on this observation, the authors of the paper believe that, if network applications are designed after certain biological principles and mechanisms, they may be able to increase their autonomy and adaptability.

BEYOND[2] is an architecture that applies biological principles and mechanisms to design autonomous and adaptive network applications. In BEYOND, each application is designed as a decentralized collection of software agents. This is analogous to a bee colony (network application) consisting of multiple bees (agents). Each agent provides a particular functionality of a network application, and implements biological behaviors such as migration, replication, and death.

This paper focuses on a self-regulatory adaptation mechanism for agents, called iNet. iNet is designed after the mechanisms behind how the immune system detects antigens (e.g., viruses), specifically produces antibodies to eliminate them, and self-regulates the production of antibodies against its anomaly (e.g., immunodeficiency and autoimmunity). iNet models a set of environment conditions (e.g., network traffic and resource availability) as an antigen and a behavior of applications (e.g., migration and death) as an antibody.

iNet allows each application to autonomously sense its surrounding environment conditions (i.e., an antigen) to evaluate whether it adapts well to the sensed conditions based on its own evaluation policy and if it does not, adaptively invoke a behavior (i.e., an antibody) suitable for the conditions. For example, agents may invoke the replication behavior at the network hosts that accept a large number of user requests for their applications. This leads to the adaptation of agent population; agents can improve their throughput. Also, agents may invoke the migration behavior to move toward network hosts that receive a large number of user requests for their applications. This results in the adaptation of agent locations; agents can improve their response time.

[1] For example, Google reportedly runs over 450,000 servers in its data centers [6, 7].

[2] Biologically-Enhanced sYstem architecture beyond Ordinary Network Designs

iNet also allows each application to dynamically configure its own evaluation policy so that it can trigger the behavior invocation only as needed. For example, when an agent realizes that the evaluation result of the environment conditions was wrong (e.g., even though an agent adapts well to the current environment conditions, its evaluation policy may notice that it does not, and then it performs one of behavior reluctantly; or, even though an agent do not adapt well to the current environment conditions, it may never invoke behaviors because its evaluation policy keeps saying that it does.), the agent configures the evaluation policy to regulate the behavior invocation (i.e., so that it can perform a behavior at the right time). This may avoid the degradation of agent's adaptability and the waste of resource consumption and execution overhead in its behavior invocation. Also, this self-regulation process frees agent developers from configuring an evaluation policy for all possible environment conditions at design time. This significantly simplifies the implementation and configuration of agents.

Simulation results show that iNet allows applications to autonomously adapt to changing environment conditions and dynamically self-regulate the behavior invocation by configuring the evaluation policy according to the detection of defect in the policy. This paper is organized as follows. Section 2 overviews the design principles in BEYOND. Section 3 describes the design details of iNet. Section 4 shows a series of simulation results to evaluate the autonomous adaptability of agents. Sections 5 and 6 conclude with comparison with several related work.

2. BEYOND ARCHITECTURE

In BEYOND, agents run on a middleware platform in a network host. Each platform provides a set of runtime services that agents use to perform their services and behaviors. There are no central entities to control and coordinate agents. Decentralization allows agents to be scalable and simple by avoiding performance bottleneck and any central coordination in deploying them [8, 9].

Each agent consists of *attributes*, *body* and *behaviors*. Attributes carry descriptive information regarding an agent (e.g., agent ID and energy level). The body implements a functional service that an agent provides. For example, an agent may implement a web service in a data center, while another agent may implement a scientific simulation model in a grid computing system. Behaviors implement the actions inherent to all agents:

- Migration: Agents may move between platforms.
- Energy exchange and storage: Agents may store and expend *energy* as biological entities strive to gain energy by seeking and consuming food. Each agent gain energy in exchange for providing services to other agents or users. They may also expend energy for services that they receive from other agents and for resources available on a platform (e.g., memory space).
- Replication: Agents may make their copies in response to higher energy level, which indicates higher demand for the agents. A replicated agent is placed on the platform that its parent agent resides on, and it receives the half amount of the parent's energy level.
- Communication: Agents may communicate with each other for the purposes of, for example, requesting a service, or exchanging energy.
- Death: Agents die due to energy starvation. If energy expenditure of an agent is not balanced with energy gain, the agent cannot pay for the resources it needs; it dies from lack of energy. When an agent dies, an underlying platform removes the agent and releases all resources allocated to the agent.

3. DESIGN OF iNET ADAPTATION MECHANISM

3.1. Natural Immune System

The immune system is an adaptive defense mechanism to regulate the body against dynamic environmental changes such as antigen invasions. Through a number of interactions among various white blood cells (e.g., macrophages and lymphocytes) and molecules (e.g., antibodies), the immune system evokes two responses to antigens: *innate* and *adaptive* responses.

In the innate response, the immune system performs self/non-self discrimination. This response is initiated by macrophages and T-cells, a type of lymphocytes. Macrophages move around the body to ingest antigens and present them to T-cells. T-cells are produced in thymus that performs the negative selection. In the negative selection process, thymus removes T-cells that strongly react with the body's own (self) cells. The remaining T-cells are used as detectors to identify foreign (non-self) cells. When a T-cell(s) detects a non-self antigen presented by a macrophage, the T-cell(s) secrete chemical signals to activate the second immune response: adaptive response.

In the adaptive response, B-cells, another type of lymphocytes, are activated by T-cells. Some of the activated B-cells who strongly react to an antigen start to replicate themselves (called affinity maturation) and produce antibodies that specifically react to the antigen identified by T-cells. Antibodies form a network and communicate with each other [10]. This network is formed with stimulation and suppression relationships among antibodies. By these relationships, antibodies dynamically change their population, i.e. proliferation and death, and their network structure. Thus, the adaptive response is offered by multiple types of antibodies, although a single type of antibody (the best matched with an antigen) may play the dominant role. The antibody network also helps to keep the quantitative balance of antibodies. Through the stimulation and suppression interactions, the population of specific antibodies rapidly increases following the recognition of an antigen and, after eliminating the antigen, decreases again. Performed based on this self-regulation mechanism, the adaptive immune response is an emergent product from many interactions among antibodies.

The human body may suffer from non-self antigens (e.g., viruses) as well as the anomalies of the immune system such as immunodeficiency and autoimmunity. Immunodeficiency is a phenomenon that the immune system cannot respond to foreign pathogens, so antibodies cannot be produced to eliminate them. On the other hand, autoimmunity is the failure of an organism to recognize its own constituent parts as self. This results in self-attacking by autoantibodies produced by overreaction of immunological cells. When the body faces threats by such anomalies, the damaged or dying cells by the threats produce danger signals [11] that alert immune system to respond to the threats.

Some research work introduces two types of danger signals, Uric acid [12, 13] and Heat Shock Proteins [14, 15]. Two different compound chemicals produced as danger signals have different functionalities. Uric acid will stimulate macrophages (or dendritic cells), which induces the adaptive response by activating T-cells, so that the immune system correctly respond to the antigens. This will accelerate the production of antibodies. On the other hand, heat shock protein (HSP) will help other proteins fold into the right shape and locate the right place (called Chaperonin-floding [15]). For example, autoimmunity may occur due to apoptosis of macrophages and cytokines dysregulation of T-cells [16, 17]. HSP reforms broken proteins in macrophages and T-cells so that they stop overreacting to self cells any more. This will suppress the antibody production.

3.2. iNet Adaptation Mechanism

The iNet artificial immune system consists of the environment evaluation (EE) facility and behavior selection (BS) facility, which implement the innate and adaptive immune responses, respectively (Figure 1). The EE facility allows an agent to continuously sense a set of current environment conditions as an antigen and classify the antigen to self or non-self. A self antigen indicates that the agent adapts to the current environment conditions well, and a non-self antigen indicates it does not. When the EE facility detects a non-self antigen, it activates the BS facility. The BS facility allows an agent to choose a behavior as an antibody that specifically matches with the detected non-self antigen.

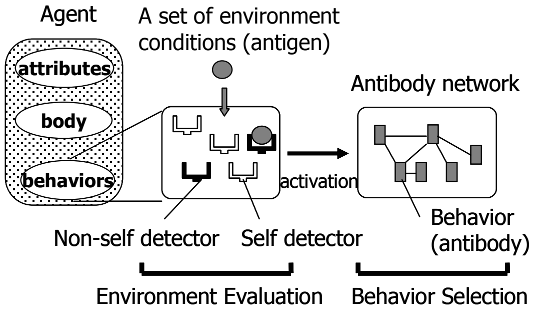


Figure 1 – Design of iNet adaptation mechanism

Environment Evaluation Facility (EE)

The EE facility performs two steps: initialization and self/non-self classification. The initialization step produces detectors (i.e. evaluation policy) that identify self and non-self antigens. Each antigen is represented as a feature vector (X), which consists of a set of environment conditions (features) (F) and a class value (C).

$$X = (F_1, F_2, ..., F_n, C)$$

Each environment condition F_i has a value; C indicates whether a given antigen (i.e., a set of environment conditions) is self (0) or non-self (1). If an agent monitors resource utilization and workload (the number of user requests) on the local host and each value is lower than a certain threshold a user specified, an antigen is represented as follows.

$$X_{current} = ((Low: Resource\ Utilization, Low: Workload), 0)$$

The initialization step in the EE facility is designed after the negative selection process in the immune system (Figure 2). As the immune system randomly generates T-cells first, the EE facility generates detectors (feature vectors) randomly. Then, the EE facility separates the detectors into self detectors, which closely match with self antigens, and non-self detectors, which do not closely match with self antigens. This separation is performed via similarity measurement between randomly generated feature vectors (R) and self antigens (S) that human users supply. After the vector matching, both self and non-self detectors are stored in the detector table[3] (Figure 2).

The second step in the EE facility is self/non-self classification of an antigen (a set of current environment conditions). It is performed with a decision tree built from detectors (i.e. evaluation policies) in the detector table and classifies an antigen into self or non-self[4]. The decision tree is built using the information gain technique [18]. First, consider one node as a root of decision tree, and it contains all detectors in the detector table. Then, divide the detector based on one of feature into two subsets of detectors. (Assume that each feature has two distinct values) Each subset goes to one of two child nodes. If the subset of detectors contains only one detector, then the node becomes a leaf node with the class label; otherwise, divide the subset again based on one of the other features into the subsets. Information gain technique suggests how to select a feature at each dividing step so that

[3] The immune system removes non-self detectors through negative selection. However, in iNet, both self and non-self detectors are used to perform self/non-self classification

[4] The reasons for using decision trees as an antigen classifier are implementation simplicity and algorithmic efficiency. Decision trees perform classification much faster than other algorithms such as clustering, support vector machine and Markov model algorithms [18, 19]. The efficiency of classification is one of the most important requirements in iNet because each agent periodically senses and classifies its surrounding environment conditions.

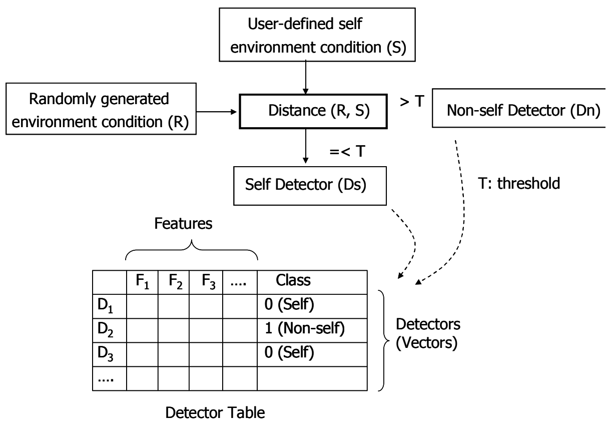


Figure 2 – Initialization step in the EE facility

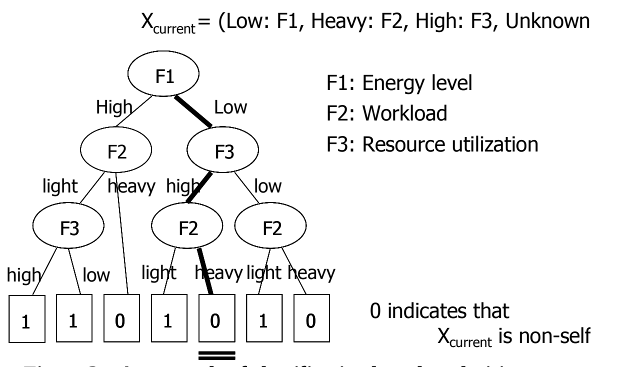


Figure 3 – An example of classification based on decision tree

the number of paths to leaf nodes and the height of tree can be minimized.

Figure 3 shows an example decision tree. Each node in the tree specifies which feature (environment condition) is considered. Based on the feature values in a given antigen, the EE facility travels through tree branches. If the EE facility classifies the antigen to non-self, then it activates the BS facility.

Behavior Selection Facility (BS)

The BS facility selects an antibody (i.e., agent's behavior) suitable for the detected non-self antigen (i.e., environment conditions). Each antibody consists of three parts: a *precondition* under which it is selected, *behavior ID* and *relationships* to other antibodies. Antibodies are linked with each other using stimulation and suppression relationships. Each antibody has its own concentration value, which represents its population. The BS facility identifies candidate antibodies (behaviors) suitable for a given non-self antigen (environment conditions), prioritizes them based on their concentration values, and selects the most suitable one from the candidates. When prioritizing antibodies (behaviors), stimulation relationships between them contribute to increase their concentration values, and suppression relationships contribute to decrease it. Each relationship has an affinity value, which indicates the degree of stimulation or suppression.

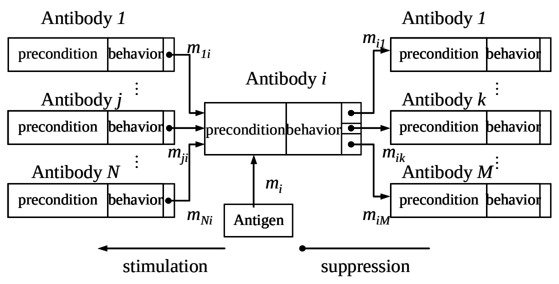


Figure 4 - A Generalized Antibody Network

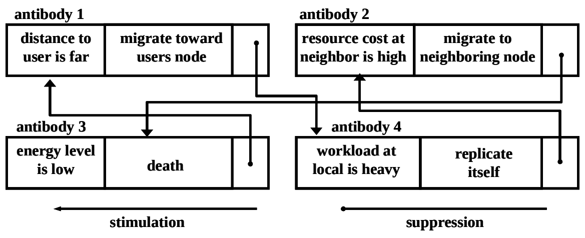


Figure 5 - An Example Antibody Network

Figure 4 shows a generalized network of antibodies. The antibody *i* stimulates M antibodies and suppresses N antibodies. *mji* and *mik* denote affinity values between antibody *j* and *i*, and between antibody *i* and *k*. *mi* is an affinity value between an antigen and antibody *i*. The concentration of antibody *i*, denoted by *ai*, is calculated with the following equations. In Equation (1), the first and second terms in a bracket denote the stimulation and suppression from other antibodies. *mji* and *mik* are positive between 0 and 1. *mi* is 1 when antibody *i* is stimulated directly by an antigen, otherwise 0. *k* denotes the dissipation factor representing the natural death of an antibody. Equation (2) is a sigmoid function used to squash the *Ai(t)* value between 0 and 1.

$$\frac{dA_i(t)}{dt} = \left(\frac{1}{N} \sum_{j=1}^{N} m_{ji} \cdot a_j(t) - \frac{1}{M} \sum_{k=1}^{M} m_{ik} \cdot a_k(t) + m_i - k \right) a_i(t) \ldots (1)$$

$$a_i(t) = \frac{1}{1+\exp(0.5 - A_i(t))} \;..... \; (2)$$

Every antibody's concentration is calculated 200 times repeatedly. This repeat count is obtained from a previous simulation experience [20]. If no antibody exceeds a predefined threshold during the 200 calculation steps, the antibody whose concentration value is the highest is selected (i.e., winner-takes-all selection). If one or more antibodies' concentration values exceed the threshold, an antibody is selected based on the probability proportional to the concentration values (i.e., roulette-wheel selection).

Figure 5 shows an example network of antibodies. It contains four antibodies, which represent the migration, replication and death behaviors. Antibody 1 represents the migration behavior invoked when the distance to users is far from an agent. Antibody 1 suppresses Antibody 3 and stimulates Antibody 4. Now, suppose that a (non-self) antigen indicates (1) the distance to users is far, (2) workload is heavy on the local host and (3) resource utilization is low on a neighboring host. This antigen stimulates Antibodies 1, 2 and 4 simultaneously. Their populations increase, and Antibody 2's concentration value becomes the highest because Antibody 2 suppresses Antibody 4, which in turn suppresses Antibody 1. As a result, the BS facility would select Antibody 2.

Self-regulation Process

As Section 3.1 describes, the immune system may suffer from its own anomalies such as immunodeficiency and autoimmunity. The damaged or dying cells caused by the anomalies produce danger signals that alert the immune system to get rid of the anomalies. Similarly, iNet allows each agent to self-regulate the behavior invocation by configuring the EE facility when it detects anomalies. In iNet, the degradation of agent's fitness caused by failure of self/non-self classification is considered as the anomaly.

Figure 6 describes the flow of self-regulation process in each agent. Corresponding to danger signals such as Uric acids and Heat shock proteins, each agent responds to two types of signals. Signal 1 is produces when the current fitness decreases by classifying the current environment conditions as Self and by not performing any behaviors even though an agent does not adapt well to the conditions (i.e. this corresponds to immunodeficiency). Signal 2 is produces when the current fitness decreases by classifying the current environment conditions as Non-self and by performing inappropriate behaviors although there is no necessity to perform behaviors because an agent adapts well to the conditions (i.e. this corresponds to autoimmunity).

When an agent receives either of signals, it flips (Self <-> Non-self) the class value of the detector, which indicates the miss-classified environment conditions. The strength of the danger signals is represented as a probability that an agent configures the class value. The probability is calculated the weighted sum of the agent's previous fitness (i.e., the degree of adaptation) and the decay of the current fitness as follow:

$$Probability = \alpha * Fitness(t-1) + (1\text{-}\alpha) * \{Fitness(t-1) - Fitness(t)\}$$

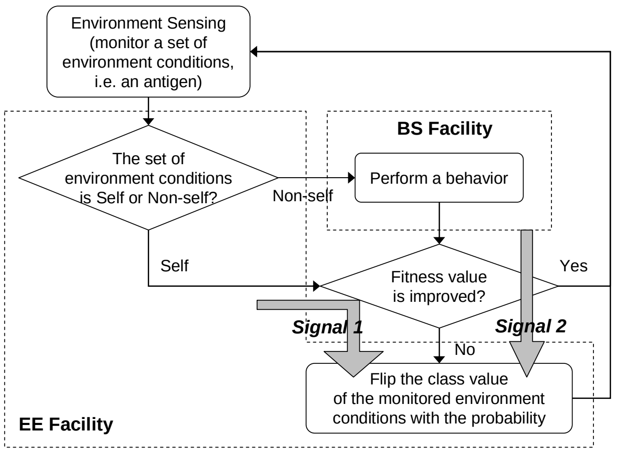
Fitness of an Agent

Each agent periodically keeps track of its *fitness*, which quantifies how much it adapts to the current environment conditions. Agents strive to increase their fitness values by performing their behaviors. Fitness is calculated as a weighted sum of fitness factors (*fi*):

$$Fitness = \sum wi * fi \qquad ..... (3)$$

Currently, iNet considers the following six fitness factors. Each factor value is non-negative between 0 and 1.

Response time (*f1*): The response time of an agent for users. *R* is the time for each agent to process a single user request.



Environment Sensing (monitor a set of environment conditions, i.e. an antigen)
BS Facility
The set of environment conditions is Self or Non-self?
Non-self
Perform a behavior
Self
Fitness value is improved?
Yes
Signal 1
Signal 2
No
Flip the class value of the monitored environment conditions with the probability
EE Facility

Figure 6 – The flow of self-regulation process

$$f_1 = \frac{R}{Response\ Time} \qquad .....(4)$$

Throughput (*f2*): indicates how many user requests agents process.

$$f_2 = \frac{\#\ of\ user\ requests\ processed\ by\ all\ agents}{Total\ \#\ of\ user\ requests} \quad .....(5)$$

Energy utility (*f3*): indicates the rate of an agent's energy expenditure to its energy gain.

$$f_3 = 1 - \frac{Energy\ expenditure\ of\ an\ agent}{Energy\ gain\ of\ an\ agent} \qquad .....(6)$$

Load balance (*f4*): indicates how user requests (workload) are distributed over agents. *m* denotes the number of user requests that an agent processes in a unit time. *um* denotes the average number of user requests that each agent is expected to process. *Mmax* denotes the maximum number of user requests that an agent can process in a unit time.

$$f_4 = 1 - \frac{m - \mu_m}{M_{\max}},\ where\ \mu_m = \frac{Total\ \#\ of\ user\ requests}{Total\ \#\ of\ agents} \quad ...(7)$$

Resource utilization balance (*f5*): indicates how resource utilization is distributed over hosts. *r* denotes the resource utilization rate on the local host that an agent resides on. This is measured as the ratio of the amount of resources consumed by agents on the host to the amount of resources available on the host. *ur* denotes the expected average of resource utilization rate over all hosts that agents reside on.

$$f_5 = 1 - (r - \mu_r)$$
$$where\ \mu_r = \frac{The\ sum\ of\ resource\ utilization\ rate\ on\ all\ hosts}{\#\ of\ hosts\ that\ agents\ resides\ on} \quad ...(8)$$

Age (f6): denotes the lifetime of an agent. \$S\$ is the total simulation time.

$$f_6 = \frac{Lifetime\ of\ an\ agent}{S} \quad .....(9)$$

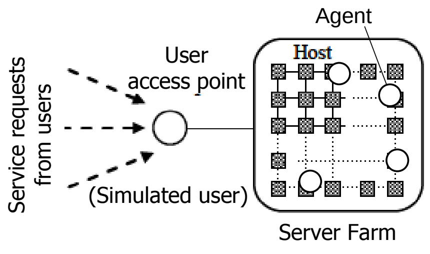


Figure 7 – A simulated network

4. SIMULATION RESULTS

This section presents a series of simulation results to evaluate the autonomous adaptability of agents by measuring of the number of agents in the network, response time for users, and agent throughput according to the workload trace. The simulations are carried out on the BEYOND simulator. Figure 7 shows a simulated network as a server farm consisting of network hosts connected in a 10x10 grid topology. Each agent implements a web service that receives an HTTP user request and returns a certain HTML file. User requests travel from users to agents via user access point (see Figure 7). This simulation study assumes that a single (emulated) user runs on the access point and sends user requests to agents. In this simulation study, a user issues requests for agents as described in Figure 8. At the beginning of each simulation, four agents are deployed on the network, and each agent contains an antibody network in the BS facility, which is manually configured as described in Section 3.2.

In order to investigate how a self-regulation process impacts the adaptability of agents, three types of agents are evaluated. (1) *Manual*: an agent with the manually configured EE facility as described in Section 3.2., (2) *Random*: an agent with the randomly configured EE facility, and (3) *Random+ Self-regulation*: an agent with the randomly configured EE facility and self-regulatory mechanism (i.e. it responds to danger signals and dynamically configure the EE facility).

Figure 9 shows how agents autonomously adapt their population to the workload changes. When agents receive requests, they start to provide their service for users and gain more energy from users. Agents (*Manual*) with the manually configured EE facility successfully adapt their population in timely manner. For example, at 2:00 and 4:00 when the workload surges, they increase their population; subsequently, at 6:00 when the workload drops, they immediately decrease their population by performing death behavior. On the other hand, agents (*Random*) with the randomly configure EE facility could not perform any behavior and did not adapt their population because the EE facility classified the environment conditions as Self although the workload dramatically changed. However, agents (*Random+Self-regulation*) with self-regulation process dynamically configure their EE facility so that they perform behaviors to adapt their population. For example, before 0:30, the EE facility classified the

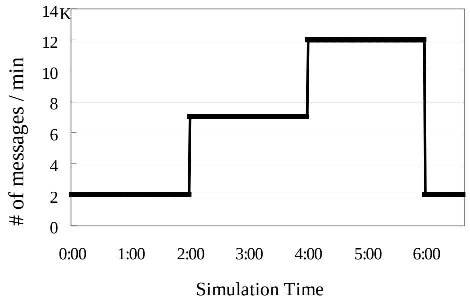


Figure 8 – Workload

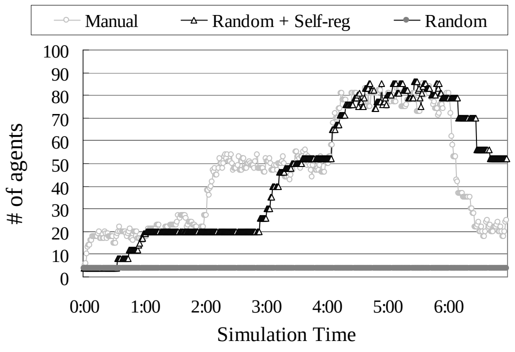


Figure 9 – Agent population

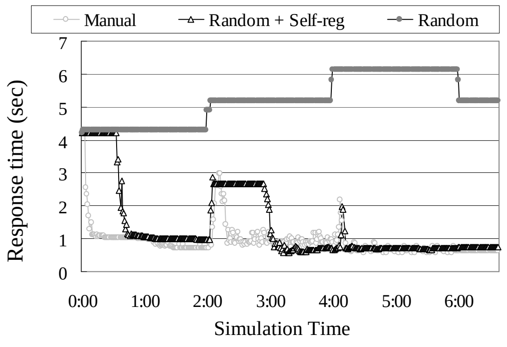


Figure 10 – Response time

environment conditions (e.g. workload is high) as Self, so agents could not perform replication behavior. Also, before 3:00, even though they did not adapt well to the environment conditions (e.g. resource utilization at local host is high), agents could not perform migration behavior because the EE facility evaluated the conditions as Self. At that time, agents respond to danger signals and dynamically configure the EE facility. Once the agents regulate the behavior invocation, they adaptively perform their behaviors in timely manner.

Figure 10 shows how agents autonomously reduce response time for a user. At the beginning of simulation, response time becomes very high because only four agents process 2,000 user requests a minute and a distance between the agent and users is long. However, after the agents accumulate enough energy from users and start to replicate themselves and migrate toward the user location, they rapidly decrease response time. For example, Agents (*Manual*) successfully reduce response time by increasing their population (i.e., increasing service availability) even when the workload increases at 2:00 and 4:00. On the other hand, agents (*Random*) did not perform any behaviors because the EE facility did not evaluate the environment conditions correctly, so they could not reduce response time. However, when agents (*Random+Self-regulation*) recognize that the EE facility wrongly evaluated the environment conditions (i.e., respond to danger signals), they try to configure the EE facility and regulate the behavior invocation so that they perform behaviors to reduce response time at 0:30 and 3:00.

Figure 11 shows how three different types of agents dynamically adapt their throughput to the workload trace. It is measured as the number of responses that a user receives a minute from agents. Agents (*Manual*) autonomously maintain high throughput by dynamically adjusting their population and locations through migration and replication behaviors while agents (*Random*) cannot improve their throughput because they did not increase their population (i.e., service availability). However, by self-regulation process, agents (*Random +Self-regulation*) recognize the defect of the EE facility and dynamically configure the EE facility. As a result, they start to increase their population and improve throughput (e.g., at 0:30 and 3:00) according to the workload.

Finally, Figure 12 shows the average fitness value of agents (i.e., the degree of adaptation to the environment) as described in Section 3.2. Agents (*Manual*) improve their fitness value to about 0.6 from 0.3 during the simulation while agents (*Random*) could not improve the fitness value (although the fitness value slightly increases because of energy utility (*f*3) and age (*f*6) factors). Agents (*Random+Self-regulation*) keep trying to improve the fitness value by regulating their behavior invocation; eventually improve their fitness value to that of *Manual* agents.

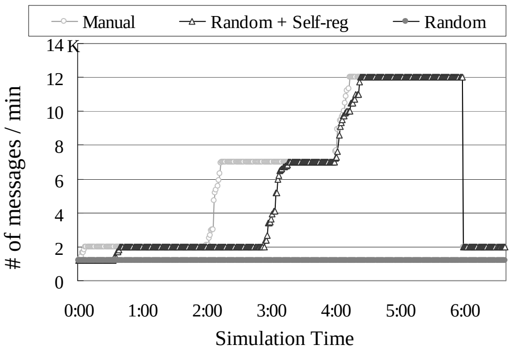


Figure 11 – Throughput

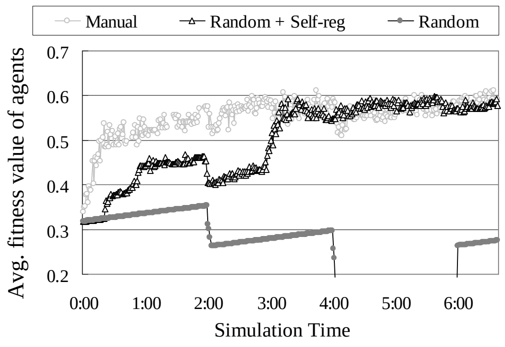


Figure 12 – Average fitness value of agents

5. RELATED WORK

This paper describes several extensions to the prior work on iNet [21, 22]. [21] mainly focus on the antibody network in the BS facility and its evolutionary mechanism; however, it does not investigate the iNet EE facility. [22] does not investigate the self-regulation process in the EE facility. Thus, agent designers needed to manually and carefully configure the EE facility by anticipating all possible self/non-self environment conditions at the design time. In contrast, the iNet with the self-regulatory mechanism allows agents to autonomously adjust their EE facility configurations at runtime; iNet requires no manual configurations for agent designers.

The Bio-Networking Architecture (BioNet) [23, 24] is similar to BEYOND in that it applies biological principles and mechanisms to allow network applications (agents) to autonomously adapt to dynamic changes in the network. The BEYOND architecture employs a different approach to design the adaptation mechanism for agents. It implements an artificial immune system as its adaptation mechanism while [23, 24] uses a factor-based weighted sum equation. In [23, 24], a behavior selection strongly depends on the values of weights and threshold, which are hard to configure as the scale of environment conditions varies. In [23, 24], each agent does not have a mechanism equivalent to the EE facility, so each agent may periodically performs one of its behaviors if the configuration was not appropriate. This results in wasting resources caused by unnecessary behavior selection. In BEYOND, each agent contains the EE facility, which examines whether it adapts well to the current environment conditions. It activates the BS facility only when the agent does not adapt to the current environment conditions. This way, agents can reduce resource consumption and execution overhead in their adaptation activities.

Artificial immune systems have been proposed and used in various application domains such as anomaly detection [25] and pattern recognition [26]. [25] focuses on the generation of detectors for self/non-self classification and improve the negative selection process of the artificial immune system. [26] focuses on the accuracy for the matchmaking of an antigen and antibody. Unlike those work, this paper proposes an artificial immune system to improve autonomous adaptability of network applications.

In addition, some research work [27] using artificial immune systems extend their work with the concept of danger signals. [27] proposes the mechanism to detect misbehaving nodes as antigens based on event sequences of routing process in ad hoc network. Danger signals contribute to reduce the number of false positives (i.e., the system evaluates a correctly working node as a misbehaving node) by dynamically updating the definition of normal event sequences (self). On the other hand, iNet self-regulation process allows agents to respond false positives as well as false negatives (i.e. the system cannot catch unknown non-self antigens).

6. Conclusion

This paper describes and evaluates an immunologically inspired adaptation mechanism, called iNet, which allows network applications to autonomously adapt to dynamic changes in the network. iNet also allows each network application to self-regulate the behavior invocation by dynamically configuring the EE facility when the adaptability drops. Simulation results show that iNet agents improve the performance such as response time for user request and throughput according to the workload by adapting their location and population.

Reference

[1] P. Dini, W. Gentzsch, M. Potts, A. Clemm, M. Yousif and A. Polze, “Internet, Grid, Self-adaptability and Beyond: Are We Ready?,” *Proc. of IEEE Int’l Workshop on Self-Adaptable and Autonomic Computing Systems*, Aug 2004.

[2] R. Sterritt and D. Bustard, “Towards an Autonomic Computing Environment,” *Proc. of 14th IEEE Int’l Workshop on Database and Expert Systems Applications*, Sep 2003.

[3] J. Rolia, S. Singhal, and R. Friedrich, “Adaptive internet data centers,” *Proc. of International Conference on Advances in Infrastructure for Electronic Business, Science, and Education on the Internet*, 2000.

[4] Large Scale Networking Coordinating Group of the Interagency Working Group for Information Technology Research and Development (IWG/IT R&D), *Report of Workshop on New Visions for Large-scale Networks: Research and Applications*, Mar 2001.

[5] S. Camazine, J. L. Deneubourg, N. R. Franks, J. Sneyd, G. Theraula and E. Bonabeau, *Self Organization in Biological Systems*, Princeton University Press, 2003.

[6] D. F. Carr, “How google works,” *Baseline*, 2006.

[7] J. Markoff, and S. Hansell, “Google’s not-so-very-secret weapon.,” *International Herald Tribune*, 2006.

[8] N. Minar, K. H. Kramer and P. Maes, “Cooperating Mobile Agents for Dynamic Network Routing,” In *A. Hayzelden and J. Bigham (eds.) Software Agents for Future Communications Systems*, Springer, 1999.

[9] G. Cabri, L. Leonardi and F. Zambonelli, “Mobile-Agent Coordination Models for Internet Applications,” In *IEEE Computer*, Feb 2000.

[10] N. K. Jerne, “Idiotypic Networks and Other Preconceived Ideas,” In *Immunological Review*, vol. 79, 1984.

[11] P Matzinger, “The Danger Model: A Renewed Sense of Self,” *Science*, 2002.

[12] K. R. Jerome, and L. Corey, “The Danger Within,” *The New England Journal of Medicine*, Vol. 350, 2004.

[13] W. R. Heath1, and F. R. Carbone1, “Immunology: Dangerous Liaisons,” *Nature* 425, 460-461, Oct 2003.

[14] B. Goldman, “White Paper: Heat Shock Proteins' Vaccine Potential From Basic Science Breakthroughs to Feasible Personalized Medicine,” *Antigenic*, July 2002.

[15] W. A. Fenton, and A. L. Horwich, ”Chaperonin-mediated protein folding: fate of substrate polypeptide,” *Quarterly Reviews of Biophysics*, May 2003.

[16] R. Bacchetta, F. Cattaneo, and M. G. Roncarolo, “Immunodeficiencies and Autoimmunity: Dysregulation of Regulatory T cells,” *Allergy and Clinical Immunology*, Vol. 15(3), Sep 2005.

[17] M. J. Clemens, W. J. van Venrooij, L. B. A. van de Putte, “Apoptosis and Autoimmunity,” *Nature*, Vol.7, Jan 2000.

[18] T. Mitchell, Machine Learning, McGraw-Hill, 1997.

[19] P. Berkhin, “Survey of Clustering Data Mining Techniques,” Accrue Software, Inc., San Jose, CA, 2002.

[20] J. Suzuki and T. Suda, “Adaptive Behavior Selection of Autonomous Objects in the Bio-Networking Architecture,” *Proc. of 1st Annual Symposium on Autonomous Intelligent Networks and Systems*, May 2002.

[21] C. Lee and J. Suzuki, “An Immunologically-inspired Adaptation Mechanism for Evolvable Network Applications,” In *Proc. of the 4th IEEE Consumer Communications and Networking Conference (CCNC)*, Las Vegas, NV, January 2007.

[22] C. Lee and J. Suzuki, “Biologically-Inspired Design of Autonomous and Adaptive Grid Services,” In *Proc. of the 2nd IEEE International Conference on Autonomic and Autonomous Systems (ICAS)*, Santa Clara, CA, July 2006.

[23] T. Suda, T. Itao and M. Matsuo, “The Bio-Networking Architecture: The Biologically Inspired Approach to the Design of Scalable, Adaptive, and Survivable/Available Network Applications,” In *K. Park (ed.) The Internet as a Large-Scale Complex System*, Oxford University Press, June 2005.

[24] T. Nakano and T. Suda, “Self-Organizing Network Services with Evolutionary Adaptation,” In *Special Issue on Adaptive Learning Systems in Communication Networks, IEEE Transactions on Neural Networks*, Vol.16, No. 5, Sep 2005.

[25] F. A. Gonzalez and D. Dasgupta, “Anomaly Detection Using Real-valued Negative Selection,” *Genetic Programming and Evolvable Machines*, 2003

[26] L. N. de Castro and J. Timmis, “Artificial Immune Systems: A Novel Paradigm to Pattern Recognition. Artificial Neural Networks in Pattern Recognition,” In *Proc. of Symposium on Soft Computing*, 2002.

[27] S. Sarafijanovic and J.-Y. Le Boudec, “An Artificial Immune System Approach with Secondary Response for Misbehavior Detection in Mobile Ad-Hoc Networks,” *IEEE Transactions on Neural Networks, Special Issue on Adaptive Learning Systems in Communication Networks*, Vol. 16(5), 2005.