An Immunologically-inspired Adaptation Mechanism for Evolvable Network Applications

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

Abstract—**Large-scale network applications are expected to be more autonomous and adaptive to dynamic changes in the network to improve user experience, expand applications' operational longevity and reduce maintenance cost. Based on the observation that various biological systems have already met the requirements (i.e., autonomy and adaptability), this paper describes a biologically-inspired framework, called iNet, to design autonomous and adaptive network applications. iNet is designed after the mechanisms behind how the immune system works. iNet models a set of environment conditions (e.g., network traffic and resource availability) as an antigen and a behavior of network applications (e.g., migration and reproduction) as an antibody. iNet allows network applications to autonomously sense its surrounding environment conditions (i.e., antigens) and adaptively invoke a behavior (i.e., antibody) suitable for the conditions. The configuration of antibodies evolves via genetic operations (e.g., mutation and crossover). Simulation results show that iNet allows agents to autonomously adapt to changing environment conditions by invoking their behaviors suitable for the current environment condition and evolving their antibody configurations.**

I. Introduction

Large-scale network applications such as grid computing applications and data center applications face several challenges, particularly autonomy and adaptability, as they have been rapidly increasing in complexity and scale [1, 2, 3]. They are expected to autonomously adapt to dynamic 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. As inspiration for a new design paradigm for network applications, the authors of the paper observe that various biological systems have already developed the mechanisms necessary to meet the above requirements (i.e., autonomy and adaptability) [4]. For example, bees act autonomously, influenced by local conditions and local interactions with other bees. A bee colony adapts to dynamic environmental conditions. For example, 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 autonomy and adaptability.

BEYOND[1] is an architecture that applies biological principles and mechanisms to design autonomous and adaptive network applications. In BEYOND, each network 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, reproduction and death.

This paper focuses on the autonomous adaptability for agents in BEYOND, and describes a biologically-inspired adaptation mechanism for the agents. The proposed mechanism, called iNet, is designed after the mechanisms behind how the immune system specifically produce antibodies to eliminate antigens (e.g., viruses) and evolves antibodies. iNet models a set of environment conditions (e.g., network traffic and resource availability) as an antigen and an agent behavior as an antibody. Each agent contains its own immune system (i.e., iNet), and the configuration of iNet (antibodies) defines the agent's behavior policy (i.e., when to invoke which behavior). iNet allows each agent to autonomously sense its local environment conditions (i.e., an antigen) and 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 services. This leads to the adaptation of agent population, and 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 services. This results in the adaptation of agent locations, and agents can reduce response time for users.

iNet also allows each agent to dynamically evolve its own iNet configuration (antibody configuration) when it replicates or reproduces. Agents perform this evolution process by generating behavioral diversity and executing natural selection. Behavioral diversity means that different agents maintain different iNet configurations. This is generated via mutation and crossover, which alter iNet configurations during agent replications and reproductions. Natural selection retains the agents that adapt well to environment conditions (i.e., the agents that have iNet configuration suitable for the environment conditions) and eliminate the agents that does not adapt to the conditions. This evolution mechanism frees application developers (agent developers) from anticipating all possible environmental changes and tuning their agents to the changes at design time, thereby significantly simplifying the process to design and configure agents.

[1] BEYOND: Biologically-Enhanced sYstem architecture beyond Ordinary Network Design

This paper describes the design of iNet and evaluates how iNet contributes for agents to autonomously adapt to environmental changes in the network. Simulation results show that iNet allows network applications to evolve through generations and to autonomously adapt to changing environment conditions.

II. 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 [14, 15].

Each agent consists of *attributes*, *body* and *behaviors*. Attributes carry descriptive information regarding an agent (e.g., agent ID). The body implements a functional service the agent provides. For example, an agent may implement a web service in a data center, while another agent may implement a physical model for scientific simulations 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).

Replications: 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.

Reproduction: Agents may select another agent as a mating partner and reproduce a child which inherits the traits (e.g. iNet configurations) of its parents by crossover and mutation of genes. Agents are *evolvable* by generating behavioral diversity and executing natural selection, performed based on the concept of energy. For example, an abundance of stored energy indicates higher demand to an agent; thus, the agent may be designed to favor reproduction or replication to increase its availability. A scarcity of stored energy (i.e., an indication of lack of demand) eventually causes elimination of the agent.

Communication: Agents may communicate each other for the purposes of, for example, requesting a service, exchanging energy, or reproducing a child.

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.

III. Design of iNet Adaptation Mechanism

This section overviews how the natural immune system works (Section III.A) and describes how iNet is designed after the natural immune system (Section III.B).

A. *Natural Immune System*

The immune system is an adaptive defense mechanism to regulate the body against dynamic environment changes (e.g. antigen invasions). When the immune system detects an antigen invasion, it activates B-cells (a certain type of immune 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 structure and communicate with each other [9]. This network is formed with stimulation and suppression relationships among antibodies. By these relationships, antibodies are dynamically changing their population, i.e. proliferation and death, and change their network structure. Thus, the adaptive immune 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.

Besides, the immune system inherits immune cells from parents to children via maternal milk and placenta. Even single-cell organisms inherit genes to their offspring. Also, during affinity maturation, a primary repertoire of roughly 10^9 different B-cell receptors can be generated from individual genes by recombining existing gene segments or mutation. The different combinations of a set of genes give rise to receptors that can bind unlimited numbers of foreign invaders, and specific antibodies are produced from the receptors [10]. This process helps to react to a variety of antigens, so the immune system needs to generate vast numbers of antibodies.

B. *iNet Artificial Immune System*

iNet allows an agent to continuously sense a set of current environment conditions as an antigen and to select a behavior as an antibody that specifically matches with the sensed antigen. Each antibody consists of three parts: *precondition* under which it is selected (one of environment conditions), *Behavior ID*, one of agent behaviors, and *relationships* to other antibodies (one or more links). Antibodies are linked with each other using stimulation and suppression relationships. Each antibody has its own concentration value corresponding to the number of the antibody. The value is used to prioritize antibodies (behaviors) in behavior selection. iNet 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 antigens (behaviors), stimulation relationships between them contribute to increase their concentration values, and suppression relationships contribute to decrease it. Each relationship has its own strength (affinity),

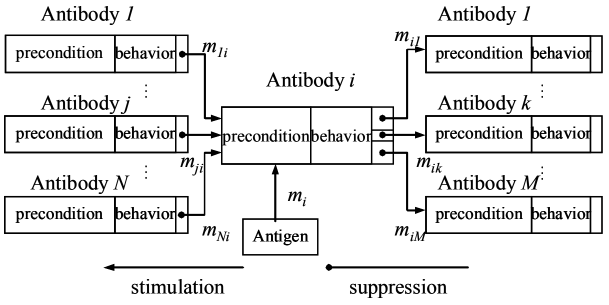

Figure 1. A Generalized Antibody Network

which indicates the degree of stimulation or suppression.

Figure 1 shows a generalized network of antibodies. The antibody i stimulates M antibodies and suppresses N antibodies. m_{ji} and m_{ik} denote affinity values between antibody j and i, and between antibody i and k. m_i is an affinity value between an antigen and antibody i. The concentration of antibody i, denoted by a_i, is calculated with the following equations.

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

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

In the equation (1), the first and second terms in a big bracket denote the stimulation and suppression from other antibodies. The affinity values between antibodies (i.e. m_{ji} and m_{ik}) are positive between 0 and 1. m_i 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. This value is 0.1. The initial concentration value for every antibody, $a_i(0)$, is 0.01. The equation (2) is a sigmoid function used to squash the $A_i(t)$ value between 0 and 1.

Every antibody's concentration is calculated 200 times repeatedly. This repeat count is obtained from a previous simulation experience [12]. If no antibody exceeds a predefined threshold (0.7) during the 200 calculation steps, the antibody whose concentration value is the highest is selected (i.e. winner-tales-all selection). If one or more antibodies' concentration values exceed the threshold, an antibody is selected based on the probability proportional to the current concentrations (i.e. roulette-wheel selection).

Figure 2 shows an example network of antibodies. It contains four antibodies, which represent migration, replication and death behaviors. Antibody 1 represents the migration behavior invoked when distance to users is far. Antibody 1 suppresses Antibody 3 when it is stimulated Now, suppose that an antigen indicates (1) user location is far, (2) workload is heavy on the local platform and (3) resource cost is high at neighboring platform. This antigen stimulates Antibodies 1, 2 and 4 simultaneously. Their population increases, and it is likely that Antibody 2's concentration value becomes highest because Antibody 2 suppresses Antibody 4, which in turn suppresses Antibody 1. As a result, Antibody 2 (i.e. migration to neighboring node behavior) would be selected.

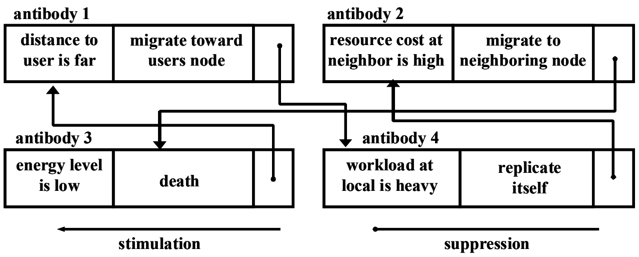

Figure 2. An Example Antibody Network

Evolutionary Mechanism of iNet: iNet allows agents to select a mating partner and produce a child by performing reproduction behavior. The generated children inherit the parameters and structure of antibody network (described in section III.B) from their parents by genetic operations, crossover and mutation. iNet employs the concept of fitness for survival of agents who adapt to current environment well. The fitness indicates the degree of adaptation to current environment and application requirements. Through the evolutionary process, agents improve their fitness from generation to generation. In other words, they autonomously adjust their iNet configuration by finding optimal values of antibody network. This results in the evolution of iNet and survival of agents with the high adaptability to environment changes.

(1) Chromosome and Genes: Each agent has one antibody network as its phenotype, which consists of a set of antibodies stimulating or suppressing each other (affinity relationship). The genotype is defined by the *antibody genes* which specify the presence of antibodies and the *affinity genes* which specify affinity relationship. The phenotype is determined by genotype on a chromosome, so each agent has a sequence of genes composed of a binary string and a set of affinity values representing the genotype (Figure 3). For example, an agent whose genotype is {(1, 0, 1, 1), (0, 1, 2, 0, 0, 0, .5, 0, 1, 0, 0, .5)} has an antibody network consists of 3 antibodies (ID:1,3,4) and 5 affinity relationships.

(2) Fitness Value: In order to evaluate the adaptability of agent, the degree of adaptation (how well an agent adapts to current environment) is quantified with fitness values based on following six factors.

***Response Time (f1)**:* Each agent tries to reduce the response time for users requesting a service. *R* is minimum response time taken to provide services. (e.g. in our simulation *R*=0.2sec. An agent running on the same node as users takes 0.2sec to provide its service).

$$f_1=\frac{R}{ResponseTime}\ \ldots..(3)$$

***Throughput (f2)**:* Each agent tries to improve its throughput according to workload (e.g. the number of requested messages). *Throughput* is represented as a ratio of provided messages by an agent to users and requested messages to the agent from users.

$$f_2=\frac{\#ofMessages.provide}{\#ofMessages.request}\ \ldots..(4)$$

***Energy* Efficiency *(f3)**:* Each agent tries to improve its *En-*

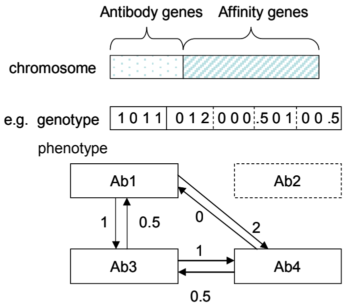

Figure 3 . Phenotype and genotype

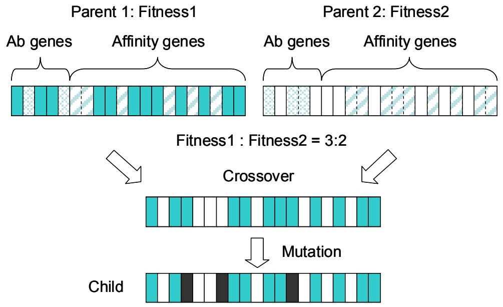

Figure 4. An example of genetic operations

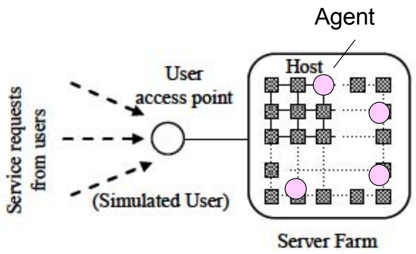

Figure 5. Simulated network

ergy Efficiency, which is measured as consumption rate, *CR,* the ratio of the consumed energy and gained energy by an agent. *E* is the desired consumption rate in [0,1] according to environment conditions (e.g. *E*=0.1 indicates agents prefer to stay rather than invoking behaviors while E=0.7 indicates agents often perform behaviors to get more energy).

$$f_3 = \min\left(\frac{E}{CR}, \frac{CR}{E}\right), where\ CR = \frac{Energy.consume}{Energy.gain} \text{.....(5)}$$

***Resource* Efficiency *(f4)*:** Each agent tries to efficiently utilize available resources (e.g. memory spaces). *Resource Efficiency* is measured as resource utilization, *RU*, over workload ratio *WL* (a fraction of # of request messages on a platform and # of messages that agents can handle on the platform during a particular time period), and also it is influenced by queue utilization, *QU,* the fraction of the number of waiting messages to be processed in the queue and the maximum queue size (e.g. max resource for a node is as 64M and max queue size is 1000 messages).

$$f_4 = 1 - QU * \left(\frac{RU}{WL}\right)^2 \quad \text{..... (6)}$$

Age (f5): Each agent has its lifetime, *L*, which indicates agents' experience (e.g. older agents have survived longer in the system). Through the evolutionary process, older agents may have adapted antibody network. *S* is the desired agent lifetime in second (e.g. *S*=3600, 7200, etc. When workload is changed every one hour, agents survived in more than one hour imply that they adapt to the environment change). *Age* factor is increased along a sigmoid curve as follows.

$$f_5 = \frac{1}{1 + \exp(-L + 0.5)}, where\ L = \frac{Time.curr - Time.born}{S} \text{...(7)}$$

Load Balancing (f6): Each agent tries to balance the workload into neighboring nodes. *Load Balancing* is measured as variance of agent distribution on local and neighboring nodes. *N* is # of neighboring nodes and one for local node.

$$f_6 = 1 - \frac{LB}{N}, where \quad \text{..... (8)}$$

$$LB = \#ofAgents(local) - \frac{\#ofAgents(local\ and\ neighbors)}{N}$$

Finally, the fitness value *F* of each agent will be computed as equation (9) in a particular time period. *w* is the weight value for each fitness factors. For simplicity, *w*s are defined as 1/6 in the simulation section.

$$F = \sum wi * fi\ ,\ 1 \le i \le 6 \text{..... (9)}$$

(3) Genetic Operations (Crossover and Mutation): When an agent invokes a reproduction behavior, it finds a mating partner agent and generates a child. The agent will select its partner running on the same platform and whose fitness value is greater than itself. The generated child will contain a set of genes inherited from both parents according to the ratio of parents' fitness values as described in Figure 4. If the agent cannot find such partner, then it replicates itself.

The generated children or replicated agent may mutate after the crossover. With mutation probability *M*, they may change values in both antibody genes (i.e. 0 or 1) and affinity genes (i.e. affinity values).

IV. Simulation Results

This section presents several simulation results to evaluate the autonomous adaptability of agents with iNet and their evolutionary process. The simulations are carried out on the BEYOND simulator[2].

Figure 5 shows a simulated network as a server farm consisting of network hosts connected in a 10x10 grid topology. BEYOND platform is running on each network host, and each agent implements a web service. Service requests travel from users to agents via user access point. This simulation study assumes that a single (emulated) user runs on the access point and sends service requests to agents. When a user issues a service request, request messages are broadcasted to search a target agent that can process the issued service requests.

In order to show that iNet agents effectively adapt to the environmental changes regardless of workload scale, this simulation generates a random workload for web service agents described in figure 6 (a). The workload trace is designed based on a daily request rate for the www.ibm.com site in February, 2001 [11]. The request rate peaks to about 5,500 requests per min in the morning and 9,000 requests per min in the evening. At the beginning, four agents whose antibody network is randomly configured are randomly deployed. In order to evaluate how the evolutionary process impacts the adaptability of agents, two different types of agents—agents with a reproduction behavior and without it, i.e. *with evolution* and *without*

[2] The BEYOND simulator contains 13,490 lines of Java code. It is available for researchers who investigate autonomous and adaptive grid applications (http://dssg.cs.umb.edu).

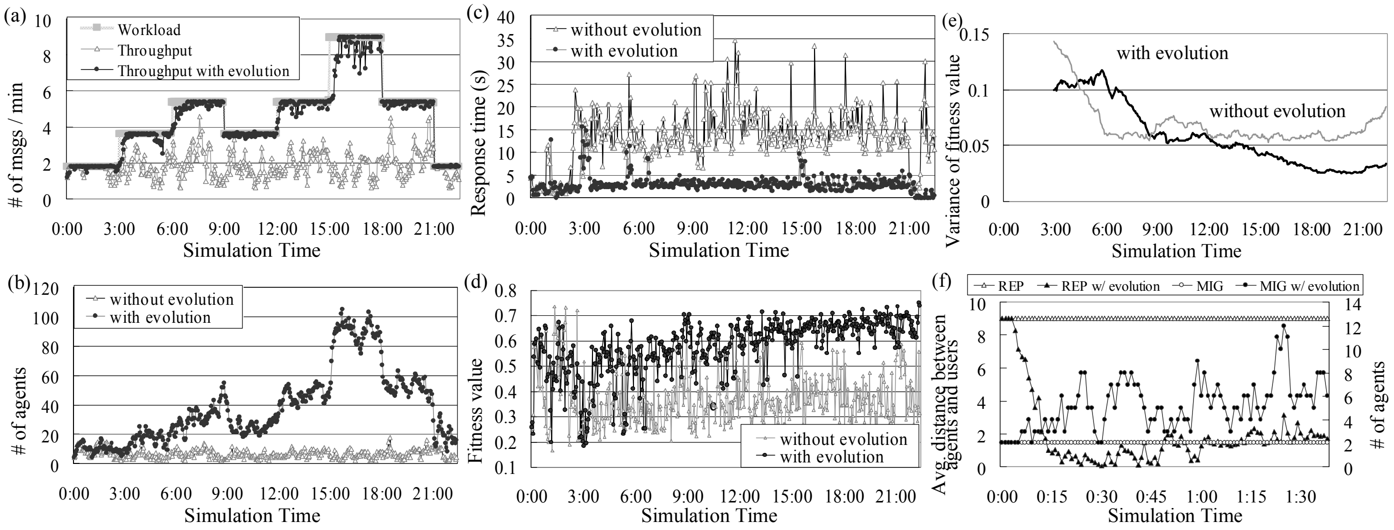

Figure 6. Performance evaluation of iNet with/without evolutionary process

evolution—are compared.

Figure 6 (b) shows how agents autonomously adapt their population to workload. When agents receive requests, they start providing their service for users. Agents gain more energy from users and try to perform replication or reproduction behavior to increase their population. However, agents without evolution cannot perform a replication behavior appropriately; moreover, they incorrectly invoke a death behavior despite receiving user requests. It follows that randomly configured agents do not have a proper antibody network suitable for the environment. On the other hand, agents with evolution successfully increase their population. This is because they reproduce children having the adapted antibody network where a replication behavior is appropriately selected according to the workload.

Figure 6 (c) shows how agents autonomously adapt response time for a user. At the beginning of simulation, response time becomes very high because only four agents process 2,000 requests a minute and a distance between the agent and users is long. However, after the agents accumulate enough energy from users and start migrating towards users and replicating themselves, they rapidly decrease response time. For agents with evolution, when workload is generated, the response time spikes up to 10 seconds, but they decrease it to 2 seconds in 30 minutes. It follows that they reproduce children who successfully invoke migration and replication behaviors according to the workload. On the other hand, agents without evolution cannot reduce their response time because they do not properly migrate towards users and increase their population.

Consequently, figure 6 (a) also shows the throughput achieved by two different types of agents. It is measured as the number of responses that users receives a minute from agents. Agents with evolution autonomously meet given workload by dynamically adjusting their population and locations through migration and replication behaviors while agents without evolution cannot achieve their throughput to workload.

Figure 6 (d) shows the average fitness value of agents. While agents without evolution do not improve their fitness value, agents with evolution reproduce children who obtain the adapted antibody network and improve fitness value to about 0.6~0.7 by increase their population and reducing response time.

Figure 6 (e) shows the variance of agents' fitness values, how the fitness values are spread around the average. The variance for agents without evolution has not converged well while the variance for agents with evolution has gradually converged to 0. The lower variance implies that all agents' fitness values are close each other. Together with the results in figure 6 (d), we can conclude that "Good" configuration (genes) of antibody network is successfully spread out to other agents by evolutionary process. Thus, agents adapts to the environment conditions well through generations.

Finally, in order to show how the evolutionary process impacts the agents' activity, two different types of agents are intentionally added. For example, the first type of agents (called REP agents) does not have a migration behavior in their antibody network, so they can only perform a replication or reproduction behavior. On the other hand, the second type of agents (called MIG agents) does not have a replication or reproduction behavior, so they can only perform a migration behavior. Figure 6 (f) shows the average distance between agents and users on the left y-axis and the number of agents on the right y-axis. It is clearly shown that REP agents cannot reduce the distance while those agents with evolution reproduce children who obtain a migration behavior and migrate towards use locations. Similarly, MIG agents cannot replicate themselves; however, those agents with evolution can obtain a replication behavior by mutation and increase their population according to the workload.

V. Related Work

This paper proposes an extended version of iNet adaptation mechanism inspired by the natural immune system. The previous version of iNet [7, 8] did not contain the evolutionary mechanism. For the previous iNet, application developers (e.g. agent designers) are required to configure the antibody network manually and intentionally before deployment (i.e. at design time) so that agents adaptively perform a suitable behavior for local environment conditions. However, since the current iNet allows agents to adjust their own iNet configuration by tuning parameters such as affinity values between antibodies, agents can be randomly configured and deployed without predicting possible environmental changes and tuning parameters at design time. The optimal iNet configurations are autonomously spread out by genetic operations to surviving agents in the network. This contributes to achieve the high adaptability of agents to dynamic environment changes over generations and to reduce the maintenance cost of agents, i.e. network applications.

This work is an extension to the Bio-Networking [5, 6], inspired by the biological principles and mechanisms that allow network systems to scale, adapt, and survive; however, the BEYOND architecture employs a different approach to design the adaptation mechanism for agents. There is a difference in how agents select a behavior (i.e. a behavior policy). iNet implements an artificial immune system as an adaptation mechanism while [5, 6] implements the weighted sum of environment conditions for each behavior.

Although [6] also includes an evolutionary mechanism which adjusts parameters (i.e. weight values) at runtime, application developers still have to manually design the weighted sum equation itself (i.e. which environment conditions should be considered as weighted sum factors) and configure the threshold value for all behaviors at offline. It is hard to find optimal threshold values without knowing network configuration beforehand. But, iNet will autonomously find the optimal affinity values (i.e. stimulation and suppression strength) and form the adaptive antibody network.

Similar work has been proposed in Organic Grid project [13]. [13] attempts to the decentralized task scheduling for large-scale computation on grid environment over the Internet. Agents autonomously find and migrate to the hosts that provide enough resources to execute subtasks allocated to the agent. iNet is designed for and can be applied into any types of application services (e.g. web service, database service, etc), while [13] is designed for ones specific to task scheduling and computation. iNet is proposed as one of design approaches for network applications, so it is unrestrained in doing anything. For this purpose, iNet includes more various behaviors and is capable of sensing more environment conditions than that of [13].

VI. Conclusion

This paper describes and evaluates an immunologically-inspired adaptation mechanism, called iNet. Each agent (application component) embeds iNet to adaptively perform their behaviors suitable for changing environment conditions. Each agent's configuration of antibodies evolves via genetic operations when the agent performs replication or reproduction. Simulation results show that iNet allows agents to autonomously adapt to dynamic changes in environment conditions by invoking their behaviors suitable for the current environment condition and evolving their antibody configurations.

References

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

[2] 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.

[3] R. Sterritt and D. Bustard, "Towards an Autonomic Computing Environment," In *Proc. of 14th IEEE Int'l Workshop on Database and Expert Systems Applications*, Sep 2003.

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

[5] 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.

[6] 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.

[7] C. Lee and J. Suzuki, "An Autonomic Adaptation Mechanism for Decentralized Grid Applications," In *Proc. of the IEEE CCNC*, Las Vegas, NV, Jan 2006.

[8] C. Lee and J. Suzuki, "Biologically-Inspired Design of Autonomous and Adaptive Grid Services," In *Proc. of the IEEE ICAS*, Santa Clara, CA, July 2006

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

[10] Claudia Berek, "Somatic hypermutation and B-cell receptor selection as regulators of the immune response," In *Transfusion Medicine and Hemotherapy*, 32(6):333, 2005.

[11] J. Chase, D. Anderson, P. Thakar, A. Vahdat, and R. Doyle, "Managing Energy and Server Resources in Hosting Centers," In *Proc. of the 18th Symposium on Operating Systems Principles*, pages 103, Oct 2001.

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

[13] A.J. Chakravarti, G. Baumgartner, M. Lauria, "The Organic Grid: Self-organizing Computational Biology on Desktop Grid," In *A Zomaya, Parallel Computing for Bioinformatics*, John Wiley & Sons, 2005.

[14] 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.

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