ASIA-PACIFIC INSTITUTE OF INFORMATION TECHNOLOGY

Computer Networks and Distributed Systems

(*Research Paper*)

Biological Architectures for Computer Networking

By:

Oon Thiam Teck

KL003833_HF0272SE

1. Introduction:

The usage of computer network is growing in a high speed. We can foresee that the scale and complexity of future network will keep increase and the network will finally cover all human and most of the human made electronic devices.

This universal network will spans locations engaged in every human endeavor, including the home, workplace, cars, trains, airplanes, train stations, airports, and space. The universal network also supports various types of information that diverse devices collect (ranging from the temperature of the refrigerator to user personal information), creating a complex web of information [2]. Any fraud in the network will cost a higher lost than ever.

So there is a need for radical change in the current computer network architecture to ensure its abilities to face the challenges in the future. The future network architecture should have the following features [2]:

- emergence of a useful network-wide behavior from autonomous and local interaction of various system architectural components
- self-organization (requiring no central coordination and administration)
- self-adaptation and self-evolution to short and long term changes in user and network conditions
- built in support for scalability, mobility, security and survivability from massive failures and attacks.

Biological system such as bee colony and, immune system exists on the earth for millions of years and it go through series of evolutions to survive well in the ever changing environment. We believe these biological systems already develop some mechanisms that are required by the future network. The implementation of biological architecture for computer network is also believed to solve most of the problems face by current network design.

So the idea of adopting key principles and mechanisms of biological system in networking architecture being proposed by the Network Research Group of University of California, Irvine - *Bio-Networking Architecture*, the research topic of this research paper.

However, there are problems arise to apply biological mechanism in the computer network. The main challenges among them are the security of energy level in *Bio-Networking Architecture*, and difficulties in implementing authentication on cyber-entities.

After the first section of introduction on research topic, the Section 2 of the paper will provide an overview on the *Bio-Networking Architecture.* Section 3 and 4 will discuss about the major problems faced by the development of *Bio-Networking Architecture* with its solution and issues arise from it. Section 5 will be the conclusion.

2. Overview:

The *Bio-Networking Architecture* is a paradigm as well as middleware for the design and implementation of scalable, adaptive, and survivable/available network applications [1]. It is adopting principles and mechanism of various biological systems with careful selection.

This middleware consists of the *Bio-Networking Platform* and the *Cyber-Entities (CE).*

In the *Bio-Networking Architecture*, a network application is implemented as a decentralized collection of autonomous objects called *cyber-entities* [5]. Each *cyber-entity* follows simple behavior rules (e.g., replication, reproduction and migration) similar to biological entities. *Cyber-entities* store and expend energy for living. They gain energy in exchange for providing their services, and expend energy for performing their biological behaviors, invoking other cyber-entities' services and utilizing resources (e.g. CPU cycles and memory space) [1].

Bio-Networking Platforms provide an execution environment for the *super-entities* that comprised of multiple autonomous *cyber-entities*. It is responsible for abstract low-level operating and networking details such as I/O, concurrency, messaging, and network connection management and provides a series of runtime services that cyber-entities frequently use for performing their services and invoking their biological behaviors (e.g. death, reproduction, migration, etc).

The following is some of the biological principle and mechanism that is adopted by the *Bio-Networking Architecture*: [1] [2] [7] [8]

- **Emergence**. Desired characteristic of biological system doesn't appear on individual *cyber-entity* but emerge from groups of interacting entities.
- **Behavior selection based on local information and interaction.** *Cyber entities* influence by local environment factor and the interaction with other cyber entities. No 'master' entity is exist in the *Bio-Networking Architecture*
- **Birth and Death.** When there were *cyber-entities* crashed or die, other *cyber-entities* is replicate to ensure the availability of application.
- **Energy as Natural Selection Mechanism**. *Cyber-entities* get energy from human users and other *cyber-entities* but give energy to *Bio-Networking Platform* in order to run. *Cyber-entities* adapt them self to maximize energy gain and minimize energy spend in order to survive.
- **Evolution.** It occurs as a result of genetic diversity and natural selection. It happens in the form of mutation, crossover and inheritance (from parent entities).
- **Pheromones.** Biological entities use pheromones to attract partners, find partners, deter enemies, signify danger/weakness, and sense danger.
- **Genetic Selection and Replication.** Biological entities use genetic selection and replication to ensure diversity and evolve towards more desirable behaviors and functions.
- **Migration.** *Cyber-entities* migrate from network node to node when energy gain from migrating is higher than energy needed for migrate.

- **Immune System.** Biological immune systems detect intrusion by randomly generating receptors which bind to non-self entities.
- **Diversity.** *Cyber Entities* that perform same task is design in different behavior. There were various level of diversities such as bits, object, messages, software, hosts, and network as discuss in [9].

The following Section 2.1 and 2.2 will further explain the *Cyber Entities* and *Bio-Networking Platform*.

2.1 Cyber-Entities

Cyber-entities are autonomous mobile agents, which are used to construct network applications [1]. *Cyber-entities* monitor their environment, autonomously behave and adjust their behavior to evolve and adjust to both short term and long term environmental changes (such as dynamically changing network conditions, user preference and usage patterns) [2].

Multiple *cyber-entities* comprise a *super-entity* that implement any services built on *Bio-Networking Architecture*.

Cyber-entity consists of three major parts: attributes, body, and behaviors [2]. *Cyber-entity*'s attribute contain information about the *cyber-entity*. Its body contain content of the service provide, either data (non-executable) or application code (executable code). *Cyber-entity* behaviors decide (migration, reproduction, replicate) or execute (service, relationship establishment, etc) autonomous action of *cyber-entity*.

2.2 Bio-Net Platform

A *Bio-Networking Platform* is any network hardware device that runs a Java Virtual Machine (JVM) and the *Bio-Networking Platform* software [1].

B*io-Networking Platform* consists of four major components: *Bionet Class Loader, Bionet Message Transport, Bionet Container, and Bionet Services*.

The *Bionet services* provide a set of runtime services that cyber-entities frequently use [5]. The *Bionet container* dispatches incoming messages to cyber-entities running on a local Bio-Networking platform [5]. The *Bionet message* transport abstracts low-level networking and operating details such as network I/O, concurrency, messaging, and network connection management [5]. The *Bionet class loader* dynamically loads class definitions of *Cyber-Entities* into a JVM when they migrate from a *Bio-Networking Platform* to another [5].

The nine key *Bionet services* are briefly described below [5]:

- **Bionet lifecycle service.** Allows *cyber-entities* to change their internal state, replicate, reproduce, die, etc.

- **Bionet relationship management service.** Allows *cyber-entities* to establish, examine, update and eliminate their relationships with other cyber-entities from time to time. Organized either by keyword similarity or small world clustering [3].
- **Bionet resource sensing service.** Allows *cyber-entities* to inquire the type, amount and cost of available resources (CPU cycle and memory space) for various type of decision making (e.g. replicate, reproduce, migrate, etc).
- **Bionet energy management service.** Keeps track of the energy levels of *cyber-entities* running on a local platform, and allows the *cyber-entities* to pay energy units for receiving services from another *cyber-entity*, for utilizing resources (CPU cycle and memory space), and for performing their behaviors.
- **Bionet discovery service.** Allows *cyber-entities* to search for other *cyber-entities* on a remote node through their relationships. This discovery is called cyber-entity level discovery because the discovery is performed with the knowledge of *cyber-entities* (i.e. relationships). Relationship history is referred to enhance the performance of frequent search but slow down the search for cyber-entities seldom access [6].
- **Bionet cyber-entity sensing service.** Allows *cyber-entities* to search for other cyber-entities running on a local and neighboring platform. This discovery is called platform level discovery because the discovery is performed with the knowledge of platforms (i.e. platform connectivity).
- **Bionet pheromone emission service.** Allows *cyber-entities* to emit their pheromones (traces) and to sense pheromones emitted by other *cyber-entities*. Help *cyber-entities* to find suitable *cyber-entities* of same application to reproduce child *cyber-entities* that have desired (high demand) characteristics.
- **Bionet topology sensing service.** Allows *cyber-entities* to sense the existence of remote platforms within N hops.
- **Bionet migration service.** Helps *cyber-entities* to migrate to another platform. *Cyber-Entities* only make decision whether to migrate or not, the *Bio-Networking Platform* executes the migration.

3. Major Problems of Bio-Networking Architecture:

Bio-Networking Architecture is not very mature by now, many mechanisms (discovery mechanism, behaviors selection mechanism, etc) used still have room to improve. We will discuss two of the security-related problems faced by the *Bio-Networking Architecture* here.

3.1 Cyber Entities Energy Level Security Threat

Energy serves as the natural selection mechanism in the Bio-Net Architecture. Unlike the natural biological system where energy is stored in entities body physically (e.g. fat in human body), energy of the *cyber-entities* does not exist physically but it is just a value stored in a field to indicate the energy level. It is modified from time to time as the energy gain or spends (just like money transferring in and out from your bank account where you do not see any 'physical' money).

This brings a risk that the energy level of cyber-entities will be modified maliciously (attacked) by themselves or competition service providers in order to compete with their competitor unfairly by cheating the *Bio-Networking Platform* natural selection mechanism.

When the natural selection mechanism of the *Bio-Net Platform* does not perform fairly, other desired characteristic of biological system such as evolution, replicate and reproduction will not exist. So this is one of the most serious problems of implementing biological architecture in computer network.

In the current proposed platform, energy level can be stored in three places: in the *cyber-entities* it self, in trusted node and, in the platform in which a *cyber-entity* resides [3].

We will discuss about the types of possible attack on energy level and solution proposed for each type of attack in the following section 3.1.1 and 3.1.2.

3.1.1 Types of Attack on Energy Level:

According to [3], there were four type of attack on energy level in *Bio-Networking Architecture* identified currently. We will discuss all four type of attack and the possible purpose of such attack in this section.

The first type is the **malicious modification by the cyber-entity itself** [3]. The purpose of a *cyber-entity* to intentionally modified (increase the energy level in most of the case) its own energy level normally is to increase its chance of survivability. By doing this, the *cyber-entity* can avoid from die of starvation and keep replicate and reproduction even the services provided have a low demand.

The second type of attack is the **malicious modification by other cyber-entities** [3]. In this case, the modification will normally be decrease the energy level of *cyber-entities* of competing service provider. By doing this, they may decrease the population of competing *cyber-entities* and win in this 'unnatural' selection.

The third type is the **malicious modification by a malicious platform**. This type of attack only happens when *cyber-entities* migrate from node to node. Since the TCP/IP protocol suite is vulnerable to a variety of attacks during packet transfer, a malicious platform of a competing service provider may intercept a migrating *cyber-entity* and modify its energy value, eliminating competing *cyber-entities* [3].

The forth type is the **malicious modification by the platform in which a cyber-entity resides**. *Cyber-entities* are practically impossible to escape from this type of attack since the platform has authority of access everything of the cyber entities.

3.1.2 Solution to Different Types of Attack

The first type of attack is difficult to prevent when the energy level is stored in the cyber-*entities* itself because cyber-entities have full access to its attributes include energy level. We may need to store the energy level of *cyber-entities* in the place, either in a platform or a trusted node that cannot be accessed directly by a *cyber-entity* [3]. To prevent *cyber-entities* form providing incorrect energy-related information to the trusted node by request transaction information from the counter part of the transaction or the counter part may automatically send this information to the trusted node for verification [3].

The second type of attack can be prevent if the platform is a Java ™ platform since under the Java ™ memory management scheme, memory of each *cyber-entity* and platform is protected from each other. However, there exists a threat of modification of energy level by another *cyber-entity* when a *cyber-entity* object is serialized for migration [3]. Solution proposed in [3] is use existing techniques to make object serialization secure. For instance, if a variable is declared as private transient, the variable will not be written out in object serialization. After all non-transient data being written out, the transient data can be written out in any encrypted format. Thus, the private transient keyword associated with a variable would not allow other *cyber-entities* to read or write that particular variable in object serialization. *Cyber-entities* also need to be authenticated by trusted node to prevent malicious node collect other *cyber-entities* energy-related information by masquerade as a trusted node. We will discuss the problem of implementing authentication in Section 3.2.

The third type of attack, cryptography technique or ordinary secure connection such as SSL may be used to prevent tamper of energy level during migration [3].

The forth type of attack may only be prevent by installing a platform in trustworthy parties or by using specialized, tamper-resistant hardware such as *secure coprocessor* to ensure the integrity of a platform [3].

Table 1 is the summary of solution for different type of energy level attack for different energy level stored location.

Table 1: Security on energy level in three storage models [3]

	In cyber-entity	In trusted node	In platform
Malicious modification by a cyber-entity itself	Difficult to prevent	Checking counterpart, Secure connection	JAVA security
Tampering during migration	Secure connection	Safe	Secure connection
Malicious modification by another cyber-entity	JAVA security Encrypt in object serialization	Requires authentication	JAVA security
Malicious platform	Secure coprocessor	Secure coprocessor	Secure coprocessor

3.2 Authentication Mechanisms for Cyber-Entities

To design a security-sensitive application based on the *Bio-Networking Architecture,* it is necessary for *cyber-entities* to authenticate each other [4].

Mobile agent based distributed system need the security mechanisms perform the following tasks [4]

- to protect the execution environments against potentially malicious mobile agent under execution,
- to protect the mobile agent against potentially malicious hosts or execution environments,
- to authenticate the mobile agent or host to ensure that the counterpart is the one who he wants.

Many mobile agent authentication mechanisms have been developed and proposed. These mechanisms include *sand-boxing*, *digital shrink-wraps*, and *proof-carrying code* for protecting the execution environment, *limited black-box security*, *computing with encrypted functions*, and *tamper-proof devices* for protecting mobile agent [4].

There were also plenty of theory, approach, and mechanism proposed by researchers for mobile agent authentication. But, these mechanisms focused mainly on the migration of mobile agents and described authentication mechanisms that are necessary when a mobile agent migrates [4].

Cyber-entities are autonomous mobile agents that comprise the network application. It have behaviors that more than just migration, its behaviors such as replicate, reproduction, pheromone emission make the authentication process much more complex. Most of the existing mechanism is not suitable for the *Bio-Networking Platform*.

So there is a desperate need for an authentication mechanism that is capable of handles not only migration, but also replicate and reproduction behaviors of *cyber-entities* efficiently because many cyber-entities are created dynamically.

The following section 3.2.1 and 3.2.2 will explain the mechanisms proposed by Y.I. Eom and T. Suda in [4]. On-line registration mechanism for clone cyber-entities is for registering newly created *cyber-entities* to the *Certificate Authority* (*CA*, a trusted third party). Peer-peer authentication mechanism between *cyber-entities* is for *cyber-entities* to authenticate each other to make sure the counterpart is the one it wants before they start interact.

These mechanisms are based on the following assumption [4]:

- *cyber-entities* do not carry security sensitive information such as keys and registration states, and all security sensitive information are managed by the hosts or *CA*
- to control *cyber-entity* creation, we let *cyber-entity* launchers be able to register additional cloning policy of their *cyber-entities* in their home site if necessary.
- each host has a security facility that can manage its private and public keys and also can provide security services for the *cyber-entities* in the *Bio-net platform*

- the security facility is secure in the viewpoint that it can be protected from the attacks by malicious *cyber-entities* or malicious hosts. It may be possible by constructing the security facility with tamper-proof hardware.

3.2.1 On-line registration mechanism for clone cyber-entities

This mechanism has two schemes for two different network environments: Single-Domain Environment and Multiple-Domain Environment.

In single-domain environment, there are several sites and cyber-entities, with a single CA. In this case, all cyber-entities should be registered in the CA [4].

The authentication in single-domain environment is relatively simple. When a parent *cyber-entity* P creates a child *cyber-entity* C, P sends a creation request to the *Bio-Networking Platform* where P resides. Then, the Bio-net platform, with its security facility, sends the registration request to the Certificate Authority (CA, a trusted third party) instead of P and C. Finally, the CA replies with the result of the registration, and the host activates the child cyber-entity when the registration is successful [4].

In multiple-domain environment, the entire network is partitioned into several domains and there is a CA in each domain. For authentication, each cyber-entity should be registered in the home CA or in the CA of the domain that it is currently located [4].

The authentication in multiple-domain environment is relatively complicated. When a parent cyber-entity creates a child cyber-entity C, P sends a creation request to *Bio-Networking Platform* where P reside. Then, the *Bio-Networking Platform* sends the request to the local CA of the domain it currently resides. The local CA processes the request and decides whether to accept the request or not. When the request is accepted, the child *cyber-entity* C is registered temporarily. C can do some jobs with some constraints after temporary registration. Local CA will immediately starts the full registration procedure by contacting the CA of the home domain of the *cyber-entity*. After full registration, the child *cyber-entity* can do its job without any constraints [4].

3.2.2 Peer-peer authentication mechanism between cyber-entities [4]

In the case where a cyber-entity X in domain CA_D wants to contact another cyber-entity in domain CA_E and the home domain (domain where there *cyber-entities* is created) of X and Y is CA_G and CA_H. It will go through the a few steps explained below.

X will first send an encrypted connection request with its private key to Y. After Y receive X's request, Y sends an encrypted authentication request with Y's private key and X's connection request to the CA_E (Y's local domain).

CA_E obtain private key and registration key of X and Y from CA_G (X's home domain) and CA_H (Y's home domain). (Mechanism for searching the CAs to obtain the public key of a cyber-entity is necessary but that is not in the scope of this paper.)

CA_E decrypts the authentication request with private key of X and Y obtains from CA_G and CA_H and knowing X and Y wants to authenticate and interact with each other. CA_E generated a new session key that is to be used for secure communication between X and Y, CA_E sends a encrypted message containing the session key, registration states of X and Y to Y.

Y uses the registration keys to determine the reply from CA_E is for which request (Y may have more than a request not reply yet).

After decrypting and verifying the message, Y obtains the session key and checks the registration state of the X. Y sends a message that containing the CA_E-signed portion of the message and the authenticator to X. This authenticator helps X to ensure that the counterpart is Y because only Y knows the session key.

After verifying the message form Y, X has authenticated the cyber-entity Y and creates a security association to Y with a session key. This security association may be restrictive according to the registration state of Y. X sends an encrypted acknowledgement message Y with the session key.

Y decrypts the message with the session key from X and verifies it. Finally, Y has also authenticated the X and creates a security association to X with the session key according to the registration state of X. By now, the mutual peer-peer authentication between cyber-entities is finished. The authenticated cyber-entities can now communicate using the secure channel.

4. Issues Arise from Problems:

4.1 Unknown Type of Attack on Energy Level

As discuss in section 3.1, the current technology can solve all four type of malicious modification on the energy level. However, there might be other type of attacks which is unidentified and unimagined yet.

Although by now we think that all type of attack on energy level can be group under one of the four type of malicious modification on energy level, this might not be true in the future. For example, the unknown type of attack might be the combination of different types of known attack.

The major origin of this idea is based on my analysis on the purpose of various type of attack on energy level. As we discuss in section 3.1, all type of attack is with the intention of cheat the natural selection mechanism of *Bio-Networking Platform* to beat the competing service provider. What if the attack is tend to turn down the whole network? Imagine all *cyber-entities* energy level are change to zero or even worst, a negative value. The whole network will down and survivability, the desired characteristic of biological system will not be there.

You might think that no one will launch such an attack that does bring benefits to anyone. Even a competing platform won't do this since all networks are interdependence in the future to provide better service and to be more competitive. I agree that there were no reason to do it, but the risk is still there.

So I suggest that there should be an emergency mechanism to prevent the network massive failure after any expected or unexpected type of attack. This emergency mechanism should be capable of ensure all *cyber-entities* energy level is keep at the level that it will not die of starvation but also not high enough for the *cyber-entities* to replicated or reproduction. It means the population of *cyber-entities* will be static while this mechanism is executed. In another words, the natural selection mechanism will be disable.

So the emergency mechanism should only be used when it is necessary and should be turn off after the network is secure from the unknown attack.

This mechanism is not a mechanism that protects the network directly but it allocated time for network administrator to identify the source of attack and finding solution for the attack without letting the network down.

Of course it is better that the emergency mechanism will have more than one implementation to increase its diversity. This enhances the reliability of the emergency mechanism.

4.2 Limitation of Proposed Authentication Mechanism

Since both registration of new *cyber-entities* and authentication among *cyber-entities* is carried out by CA and both of these activities is very frequent in the *Bio-Networking Architecture*. Y.I. Eom and T. Suda realize that the authentication mechanism discussed in [9] and Section 3.2 of this paper is having its performance bottleneck due to the possible heavy congestion around CAs

Besides that, the authentication mechanism that required the centralized CA is obviously doesn't fit well in the *Bio-Networking Architecture* which is hoping to achieve the decentralized characteristic of biological system.

So the proposed mechanism needs to be enhanced in order to fulfill user demand in the future. I would like to suggest two ways for enhancing the current mechanism in this section.

First, I suggest that we may design some *cyber-entities* that are capable of assist the CA. These *cyber-entities* serve as the agents of CA and there are distributed around the CA and they are not allowed to migrate. We will named them as '*CA-agents*' in the following discussion to differentiate them from others cyber-entities. *CA-agent* can only carry out one specific task of the CA. Of course, more than one set of *CA-agents* should be design to have better diversity. Like other *cyber-entities*, *CA-agents* may replicate, reproduce,

and death from time to time. The design of *CA-agents* should be very careful to prevent malicious *cyber-entities* masquerade as *CA-agents*. By implement this, CAs can perform better since service requests is handle by different *CA-agents* concurrently with the central control of CA. However, the mechanism is still a centralized mechanism since is still heavily depend on the individual CA in each domain.

So here is my second suggestion, we may add the replicate behavior to the CA. By doing this, the quantity of CA in the same domain will change frequently by demand. These CAs should be distributed equally in its own domain and they should be capable to move so that it may move to the area where service demand is high. To implement this, a mechanism for *cyber-entities* to locate nearest CA will be necessary. This is relatively harder to implement since it is difficult to synchronize data contain in all the CA in the same domain. However, it is worth to put into consideration since this way are more decentralize and it ensure the *Bio-Networking Architecture* survivability when one of the CA is experienced a massive failure.

5. Conclusion:

As conclusion, the development of network will cause the scale of network become unmanageable and the *Bio-Networking Architecture* which adopting key principle and mechanisms will become a trend in the future network design once it is mature enough.

In this architecture, network application is consists of interactive *cyber-entities* that employ on the *Bio-Networking Platform*. *Cyber-Entities* are autonomous mobile agent that have biological characteristic. *Bio-Networking platform* is any networking hardware that installs the *Bio-Networking Platform* software. It is more preferable that the platform is running a Java Runtime Environment (JVM).

As shown in the simulation result of Aphid (a web content distribution application run on *Bio-Networking Platform*) recorded in [1], applications which follow the biological principles and mechanisms are more adaptive, survivable, and scaleable. This result is very encouraging and it brings lots of confidence to the researcher on the potential of this technology.

However, adopting biological principles and mechanism may not be answers to all problem faced by current network design and these problems may need to be solve by ourselves. This paper is focus on the type of problem that doesn't exist in biological system but exist in the computer network. These problems include the security threat of *cyber-entities* energy level and the difficulties of implement authentication mechanism among *cyber-entities* as discuss in Section 3.

Although there are solution proposed for the problems mentioned above, but these solutions are still quite theoretical and not fully tested in the real environment yet. Therefore, the solutions also may have some known or unknown limitation.

I have tried to identify the limitations of proposed solution and express my opinion to enhance the existing solutions. My suggestion is very theoretical too and it may need time to be proof on its effectiveness. These limitations and suggestions are included in the Section 4 of this paper.

Bio-networking architecture is still new and a lot more effort is required to make it a successful technology. More researchers from not only the IT industry but also the biological fields is encourage to contribute to the project in order to speed up the development of the *Bio-Networking Architecture* since not much IT specialist is having high knowledge about biological system.

The future direction of the development will be mainly focus on refine the current proposed architecture and discover more biological principles that is suitable for implementing in the *Bio-Networking Architecture*.

The interaction with other research team with similar research topic (e.g. Artificial Life, Mobile Agent, etc) is important in order to develop a more stable and open standard of *Bio-Networking Architecture*. It also ensures the *Bio-Net Architecture* to be more acceptable by the public.

Besides solving all these technical problems of the system, effort is also needed for standardization of *Bio-Networking Architecture*. The research team has worked on standardizing key concepts and mechanisms in the *Bio-Networking Architecture* in standard specifications of OMG SDO DSIG (Object Management Group Super Distributed Objects Domain Special Interest Group).

Since the *Bio-Networking Architecture* is hardware independence, so its future is bright since its usage doesn't limited to any networking hardware and transmission medium. Its abilities to serve as an experiment platform for others similar technology also increase its chance of success.

The major challenge might not be the technical difficulties but the human behavior of resistance to change. This is because the implementation of the Bio-Networking Architecture will need a major redesign of existing network applications. A new philosophy of network application design will need to be established. Time is the only thing that capable of solving this problem of human resistance to change once this technology has convinced the public about its superiority in performance and security.

(4,885 words)

Reference:

[1] M. Wang, T. Suda, *The Bio-Networking Architecture: A Biologically Inspired Approach to the Design of Scalable, Adaptive, and Survivable / Available Network Applications*, Technical Report, February, 2000, Available on WWW at: "http://netresearch.ics.uci.edu/bionet/publications/bionet-TR00-03.pdf"

[2] T. Suda, *Adaptive Networking Architecture for Service Emergence*, Available on WWW at "http://www.ngi-supernet.org/lsn2000/UC-Irvine.pdf"

[3] S. Song, T. Suda, *Security on Energy Level in the Bio-Networking Architecture*, Technical Report, March, 2001, Available on WWW at: "http://netresearch.ics.uci.edu/bionet/publications/TR-security-on-energy.pdf"

[4] Y. I. Eom and T. Suda, *Authentication for Clone Cyber-entities in Bio-net Computing Environments,* July, 2000, Available on WWW at: "http://netresearch.ics.uci.edu/bionet/publications/eom.pdf"

[5] T. Suda, *The Bio-Networking Architecture Bi-weekly report #11 (October 28, 2002): Bio-Networking Platform*, Available on WWW at: "http://netresearch.ics.uci.edu/bionet/publications/darpa/biweekly/darpa_biweekly_1028.pdf"

[6] M. Moore and T. Suda, *A Decentralized and Self-Organizing Discovery Mechanism*, 2002, Available on WWW at: "http://netsearch.ics.uci.edu/bionet/publication/discovery_ains.pdf"

[7] J. Suzuki, T. Suda, *Adaptive Behavior Selection of Autonomous Objects in the Bio-Networking Architecture*, February, 2002, Available on WWW at: "http://netresearch.ics.uci.edu/bionet/publications/ains02_behavior.pdf"

[8] *A Biologically Inspired Network for Ubiquitous Computing*, Larry T. Chen, Network Research Group, Dept. of Information and Computer Science, University of California, Irvine, Available on WWW at "http://netresearch.ics.uci.edu/bionet/resources/ubicomp.html"

[9] *Diversity in Computing*, Larry T. Chen, Network Research Group, Dept. of Information and Computer Science, University of California, Irvine, Available on WWW at: "http://netresearch.ics.uci.edu/bionet/resources/diversity.html"