# Self-Organization based Network Architecture and Control Technologies for New Generation Networks

Naoki Wakamiya, Shin'ichi Arakawa, and Masayuki Murata
*Graduate School of Information Science and Technology, Osaka University*
*Suita, Osaka 565-0871, Japan*
*Email: {wakamiya, arakawa, murata}@ist.osaka-u.ac.jp*

***Abstract*—An emerging new generation network is requested to accommodate an enormous numbers of nodes with high diversity and a wide variety of traffic and applications. To achieve higher scalability, adaptability, and robustness than ever before, in this paper we present new network architecture composed of self-organizing entities. The architecture consists of the physical network layer, service overlay network layer, and common network layer mediating them. All network entities, i.e., nodes and networks, behave in a self-organizing manner, where the global behavior emerges through their operation on local information and direct and/or indirect mutual interaction intra- and inter-layers. We show several results to demonstrate how self-organizing network control behaves based on our architecture.**

***Keywords*-new generation network; network architecture; self-organization; bio-inspired algorithms**

## I. Introduction

To satisfy a wide range of requirements and desire of people and to support our daily life in many aspects, a variety of fixed devices such as PCs, servers, home electric appliances, and information kiosk terminals, mobile devices such as that equipped with people and vehicles, and small and scattered devices such as RFID tags and sensors, are and will be distributed in the environment. They are and will be connected with each other and organize networks. They cooperate with each other in sharing and exchanging obtained or generated information and controlling each other to provide people with desired services for safe and comfortable environment.

Those devices generate a great variety of traffic including voice, video, computer, sensing, identification, control, and management data in accordance with a type of device, application, service, and context. Traffic characteristics also have the diversity, e.g., constant/intermittent, low/high rate, and small/large amount. Furthermore, the number, type, location, and usage of devices, condition of communication environment, and traffic characteristics dynamically and considerably change every moment.

It means that new generation networks is requested to accommodate an enormous numbers of heterogeneous devices and a wide variety of traffic with substantial fluctuation under dynamically changing communication environment [1]. Therefore, a network would often face unexpected or unpredictable user behavior, usage of network, and traffic pattern, which are beyond the scope of the assumption in designing and building the network. As a result, the performance considerably deteriorates or at worst the network completely collapses. Consequently, the conventional network design methodology, where structures, functionalities, algorithms, and control parameters are optimized to accomplish the best performance assuming certain operating environment, and fault detection, avoidance, and recovery mechanisms are prepared and preprogrammed for expected failure, is no longer feasible.

Taking into account requirements for a new generation network stated above, in [2] we propose new network architecture, which is more scalable to the number of nodes and scale of network, more adaptive to a wide variety of traffic patterns and their dynamic change, and more robust to expected and unexpected failure independently of size and duration, than ever before. Our basic idea is to organize and control the whole network system in a self-organizing manner where the global behavior emerges from mutual interaction among localized behavior of small entities. A network has a layered architecture; the physical network layer, the service overlay network layer, and the common layer, which mediates inter and intra layer interaction. Behavior of all entities constituting a network system, i.e., node, network, and layer, is self-organized. A node performs MAC, scheduling, routing, congestion control, and other control by using nonlinear functional modules called *self-organization engines*, which operate based on local information obtained through observation of environment and information exchange with neighboring nodes. Nodes further organize and control a network through localized behavior and mutual interaction among them. Networks within a layer also behave in a self-organizing way and interact with each other directly by exchanging messages and/or indirectly by changing operating environment shared among them. In addition to the intra-layer interaction, service overlay networks and physical networks interact with each other through mediation of the common network layer.

In the following sections, we introduce the self-organizing network architecture first starting with basic concept and

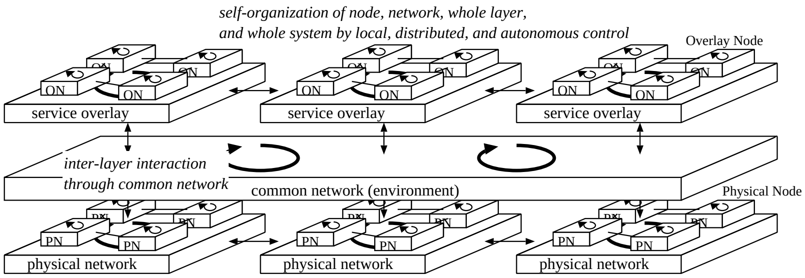


Figure 1. Self-organizing network architecture

followed by node architecture and components. We also show examples of combination of multiple self-organization engines and hierarchical control of network by a single self-organization engine. Then, we conclude the paper by mentioning related work and future issues.

## II. Self-organizing network architecture

In this section, we briefly describe our network architecture together with some sample biological models for self-organized control of new generation networks.

### A. Basic Concept

As the number of nodes and the size of network increase, a centralized mechanism becomes ineffective for considerable maintenance overhead to collect up-to-date information on the whole network system and distribute the decision to all nodes. Especially when we consider wireless communication, such control overhead occupies the limited bandwidth and disturbes regular data communication. Even semi-distributed mechanisms, such as a table-driven routing protocol, where nodes in a network perform a distributed control algorithm, requires nodes to keep the same and consistent up-to-date view of network. Therefore, we need fully distributed and autonomous control mechanisms, which enable a node to operate without the need for global information, but purely on local information obtained through observation of its surroundings and messages exchanges with neighbors. With such autonomous mechanisms, it also is possible to avoid letting a single and local failure, e.g., link disconnection, involve the whole system by propagating the failure information to update the topology information that all nodes maintain.

In addition, a conventional adaptation mechanism where the whole system is periodically re-optimized based on the up-to-date status information puts too much burden on a large-scale network to adapt to frequent changes in the operating environment. As the temporal order of changes in network condition becomes small in new generation networks, frequency of information update increases and becomes more harmful. Therefore, we need self-adaptive and self-configuration control mechanisms, which are local and distributed. Each node should autonomously and locally adapt control parameters, behavior, and even algorithm and mechanism in accordance with the state of surrounding environment.

Furthermore, a conventional network system acquires the robustness by implementing a variety of detection, avoidance, and recovery mechanisms against failure, error, abuse, extreme operating condition, and critical event. Such design methodology makes a network system complicated, monolithic, and even fragile. Therefore, we need simple and module-based control mechanisms where a node, network, and network system are constituted by autonomous, simple, and interacting functional control modules. When a part of the modules halts for unexpected failure, the remaining modules provide the minimum level of network service and provoke adaptive behavior of other modules and entities. Consequently, the whole network system adapts to the new environment.

In summary, we need a self-organizing network system where a node consists of autonomous and simple control mechanisms, mutual and local interaction among nodes organizes a network, and inter and intra layer interaction among networks organizes the whole network system. Such a network system can keep providing network services to users and applications independently of the size of system and condition of operating environment, the degree of their diversity and dynamic change, and the scale and duration of failure. The self-organizing network architecture we propose has three layers. They are the physical network layer consisting of wireless and wired access networks and optical core networks, the service overlay network layer consisting of service or application-oriented overlay networks, and the common network layer, which mediates interaction among entities within a layer and interaction among the two layers. These layers are self-organized through inter and intra-layer mutual interaction among entities. The architecture is illustrated in Figure 1.

### B. Node Architecture

In the self-organizing network architecture, each of physical and overlay nodes is composed of communication and sensing module, knowledge database module, and self-organization engines, and network control functionalities

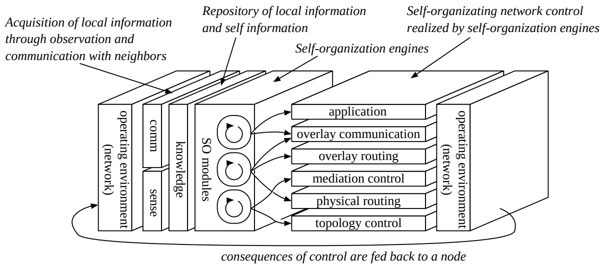

Acquisition of local information through observation and communication with neighbors
Repository of local information and self information
Self-organizing network control realized by self-organization engines
Self-organization engines
operating environment (network)
comm
sense
knowledge
SO modules
application
overlay communication
overlay routing
mediation control
physical routing
topology control
operating environment (network)
consequences of control are fed back to a node

Figure 2. Node architecture

(see Figure 2). The communication and sensing module obtains local information through message exchange with neighboring nodes and observation of environmental condition by probing or sensing for example. The module also collects status information of node itself. Obtained information is deposited into the knowledge database to be used by self-organization engines.

A self-organization engine is a basic component for self-organizing behavior of node. It operates on local information in the knowledge database and reacts to its dynamic change. By using self-organization engines, a node realizes and performs MAC, scheduling, routing, congestion control, and other network control functionalities.

Behavior of a node changes the operating environment and affects neighbors. For example, emission of a message consumes the wireless bandwidth and may cause collisions and congestion. Neighbors would react to the change and behave accordingly. Such changes in the operating environment and neighbors' behavior are observed by the communication and sensing module and fed back to the node itself.

### C. Self-Organization Engines

A self-organization engine is a nonlinear functional module and a core of self-organization. It operates on a nonlinear mathematical model in the form of differential equation. Examples of nonlinear models include a pulse-coupled oscillator model [3], a reaction-diffusion model [4], a response threshold model [5], and an attractor selection model [6]. All of these models are derived from self-organizing behavior of biological systems, which are inherently fully-distributed, autonomous, and self-organizing. As a typical example, it is well known that a group of social insects such as ants, termites, and honey bees often exhibits sophisticated and organized behavior, e.g., ant trail, cemetery formation, brood sorting, and division of labor, which is beyond mere collection of simple behavior of individuals. Such collective intelligence, called *swarm intelligence*, emerges from mutual and local interaction among simple agents [7]. By adopting such bio-inspired mathematical models to network control, we can expect to achieve a robust, scalable, and adaptive self-organizing network system.

In this subsection, we first introduce the above mentioned four nonlinear functions and how they are applied to specific network control. Then we give two examples of extended model, i.e., a combination of two nonlinear modules and a layered nonlinear module. We should note here that not all modules are incorporated to the proposed architecture and presented results are obtained by independent implementation and experiments.

*1) Pulse-coupled Oscillator Model:* A pulse-coupled oscillator model explains synchronized behavior observed in a group of flashing fireflies [3]. A firefly periodically flashes based on its biological timer keeping its intrinsic frequency when it is alone. When fireflies form a group, a flash of firefly stimulates non-flashing fireflies. A stimulated firefly advances its timer by a small amount. When the advanced timer reaches a certain threshold, the stimulated firefly flashes as well. At this moment, these fireflies are considered synchronized. By repeatedly stimulating each other through flashes, a group of fireflies eventually get synchronized and they begin to flash at the same time and at the same frequency.

In a pulse-coupled oscillator model, an oscillator maintains a timer. It fires when the phase of timer $\phi$ reaches one and then the phase goes back to zero. The dynamics of phase $\phi$ is formulated as,

$$\frac{d\phi_i}{dt} = \frac{1}{T_i} + \frac{\Delta(\phi_i)}{|\{j|j \in \mathcal{N}_i, \phi_j = 1\}|} \sum_{j \in \mathcal{N}_i} \delta(1 - \phi_j). \quad (1)$$

In (1), $T_i$ stands for the intrinsic interval of oscillator $i$'s timer. $\mathcal{N}_i$ is a set of oscillators coupled with oscillator $i$. Oscilalltors coupled with a firing oscillator are stimulated. $\Delta(\phi_i)$ is a monotonically increasing nonlinear function, which determines the amount of stimulus when oscillator $i$ receives a fire. The global synchronization, where all oscillators flash simultaneously at the same frequency, can be accomplished without all-to-all coupling. As far as the network of oscillators is connected, oscillators with similar intrinsic interval are eventually synchronized. Depending on parameters and functions, so-called phase-lock condition, where oscillators fire alternately keeping the constant phase difference, can also be accomplished and a traveling wave appears.

A direct application of the pulse-coupled oscillator model is synchronization or scheduling. By regarding a wireless sensor node as a firefly and radio signal transmission as flash of a firefly, we can self-organize synchronization in a wireless sensor network. In Figure 3, phase transition in a network of 100 nodes randomly distributed in the region of 100 m×100 m is shown. The communication range is identical among nodes and set at 20 m. Intrinsic frequency $F_i = 1/T_i$ is chosen within the range of [0.9,1.1] at random considering timer drift and error. Initial phase $\phi_i$ is also chosen at random taking into account error and asynchronous power activation. The x-axis of the figure corresponds to time and the y-axis shows state transition. Initially, phases

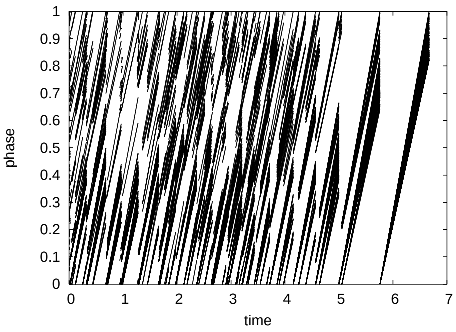


Figure 3. Synchronization in a wireless sensor network

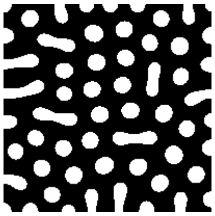


Figure 4. Spot pattern

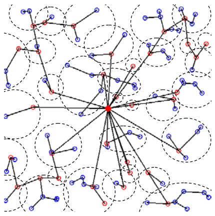


Figure 5. Clustered wireless sensor network

are different among nodes. At different rate depending on intrinsic frequency, phases shift toward one. As one of the nodes broadcasts a message, other nodes are stimulated and some of them are brought to broadcast a message. As a result of chain of stimulation, nodes are eventually merged into several groups, in which nodes broadcast a message simultaneously. Finally, the global synchronization emerges as a consequence of mutual interactions among nodes. Although delay is not considered in the experiment, a pulse-coupled oscillator model has been applied to a variety of network control such as clock and timer synchronization [8] and scheduling [9], [10], where communication delay exists.

*2) Reaction-Diffusion Model:* Next a reaction-diffusion model describes emergence of periodic patterns such as spots, stripes, and maze on the surface of animal coat through chemical interaction among cells [4]. In a reaction-diffusion model, two hypothetical morphogens, i.e., activator and inhibitor, are considered. The activator activates both of the activator and inhibitor to increase their concentrations. On the other hand, the inhibitor inhibits generation of morphogens. The dynamics of morphogen concentrations is formulated as,

$$\frac{du}{dt} = F(u, v) + D_u \nabla^2 u \tag{2}$$

$$\frac{dv}{dt} = G(u, v) + D_v \nabla^2 v \tag{3}$$

where $u$ and $v$ are concentrations of activator and inhibitor, respectively. The first term of right-hand side of the equations is called a reaction term and expresses chemical reactions, i.e., activation and inhibition among morphogens. The second term called a diffusion term is for interaction among neighboring cells. To generate a pattern, the condition $D_u < D_v$, i.e., the speed of diffusion of inhibitor is faster than that of activator, must be satisfied.

Let us consider the field of morphogens where the morphogen concentrations are the same and stable. Now assume that small perturbation makes the activator concentration slightly larger than the inhibitor concentration at a point in the field. The increased activator activates morphogenesis generation at the point. Although both of activator and inhibitor increases at the point, the generated inhibitor diffuses around for the faster diffusion rate than the activator. As a consequence, the concentration of inhibitor at the point becomes smaller than that of activator. At the same time, the diffused inhibitor inhibits generation of morphogens and then the concentration of activator becomes small at the diffused area. Eventually, we observe the spatial distribution of nonuniform morphogen concentrations, i.e., a pattern.

Autonomously generated patterns can be used in several network controls where a pattern of communication and control appears, such as routing, clustering, and placement of nodes, agents, or contents. For example, a spot pattern generated by the reaction-diffusion model in Figure 4 resembles to the clustered structure of a wireless sensor network in Figure 5. In [11], a node evaluates the reaction-diffusion equations by using the morphogen concentrations of itself and neighboring nodes. Eventually a spot pattern appears where each spot is centered at a node with the highest activator concentration in the proximity, which becomes a cluster head. Neighboring nodes, i.e., cluster members, send their sensor data following the gradient of activator concentration to a cluster head. By taking account of the residual energy in the morphogen concentrations, energy-efficient clusters can be formed in a self-organizing manner. It is shown that the near-optimal clusters can be formed by a reaction-diffusion based mechanism with only 8.4% increase in the energy×delay cost from the optimal solution. Another example of applications of a reaction-diffusion model is scheduling of spatial TDMA MAC protocol [12].

*3) Response Threshold Model:* A response threshold model explains division of labor in a colony of social insects [5]. The ratio of individuals engaged in a certain task is autonomously controlled in accordance with the demand. The demand $s$ of a task changes as,

$$\frac{ds}{dt} = \delta - \frac{\alpha N_{act}}{N}, \tag{4}$$

where $\delta$ corresponds to the per-time increase in demand, $N$ is the total number of individuals, and $N_{act}$ means the number of individuals engaged in the task. When $N_{act}$ is not sufficiently large, the demand increases. The probabilities that individual $i$ starts or stops performing the task are given as,

$$P(x_i = 0 \rightarrow x_i = 1) = \frac{s^2}{s^2 + \theta_i^2}, P(x_i = 1 \rightarrow x_i = 0) = p. \tag{5}$$

$x_i$ indicates the state of individual $i$, where $x_i = 1$ corresponds to performing the task. $\theta_i$ is a response threshold of individual $i$ against the task, which implies the willingness or hesitation in doing the task. $p$ is a constant. Adaptive division of labor or specialization emerges from the following learning function.

$$\frac{d\theta_i}{dt} = \begin{cases} -\xi & \text{if } x_i = 1 \\ \varphi & \text{if } x_i = 0 \end{cases} \tag{6}$$

This adaptation leads to division of labor in two groups, specialists actively participating in task having a small threshold and idle ones having a large threshold. When some of specialists accidentally die, the demand begins to increase. Then, individuals belonging to the latter group eventually start to perform the task. Finally the appropriate ratio $\frac{N_{act}}{N}$ recovers.

Examples of application of a response threshold model include task allocation for mobile sensor network coverage [13] and sensor and actuator networks [14]. In II-C6, we incorporate the response threshold model with the pulse-coupled oscillator model to achieve an energy-efficient and adaptive surveillance control.

*4) Attractor Selection Model:* Finally, an attractor selection model duplicates non-rule adaptation of *E. coli* cells to dynamically changing nutrient condition in the environment [6]. A mutant *E. coli* cell has a metabolic network consisting of two mutually inhibitory operons, each of which synthesizes different nutrient, i.e., glutamine and tetrahydrofolate. When a cell generates one nutrient more, it does the other nutrient less. If a cell is in a neutral condition where both nutrients exist, the concentrations of mRNAs dominating protein production are at a similar level. Therefore, a cell synthesize either of nutrients. Once one of nutrient becomes insufficient, the level of gene expression of operon for the missing nutrient eventually increases so that a cell can live in the new environment by compensating for the insufficient nutrient. However, there is no signal transduction, i.e., embedded rule-based mechanism, from the environment to the metabolic pathway to switch between two operons.

The dynamics of concentration of mRNAs is formulated in a general form as,

$$\frac{d\vec{x}}{dt} = f(\vec{x}) \cdot \alpha + \vec{\eta}, \tag{7}$$

where $\vec{x}$ corresponds to the concentrations of mRNA. $f(\vec{x})$ is a function for chemical reaction on the metabolic network. $\alpha$ represents the cellular activity such as growth rate and expresses the goodness of current behavior, i.e., gene expression. Finally, $\vec{\eta}$ expresses internal and external noise affecting the cell behavior. In the case of the `E. coli`'s experiments, the first term is formulated as [6],

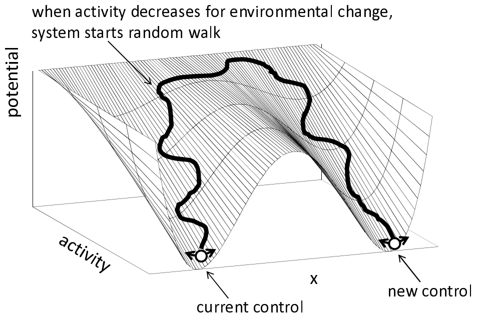

when activity decreases for environmental change, system starts random walk
potential
activity
x
current control
new control

Figure 6. Attractor selection model

$$\frac{dm_1}{dt} = \frac{s(\alpha)}{1 + m_2^2} - d(\alpha)m_1 + \eta_1 \tag{8}$$

$$\frac{dm_2}{dt} = \frac{s(\alpha)}{1 + m_1^2} - d(\alpha)m_2 + \eta_2. \tag{9}$$

$m_1$ and $m_2$ are mRNA concentrations and $s(\alpha)$ and $d(\alpha)$ are functions of synthesis and decomposition, respectively. Due to the mutually inhibiting relationship, the dynamics of the above nonlinear equations has two attractors, $m_1 > m_2$ and $m_1 < m_2$. An attractor is a state where a nonlinear dynamical system converges and becomes stable. If the current attractor, i.e., morphogen generation, is suitable for the current nutrient condition, the activity becomes high. Then the basin of the current attractor becomes deep and the cell stays there. Now the environment changes and the current attractor becomes inappropriate. The activity decreases and the basin becomes shallow accordingly. Being driven by the noise term, the mRNA concentrations of the cell change randomly. When the mRNA concentration corresponding to the missing nutrient becomes larger, the activity slightly increases. It makes the basin of the appropriate attractor deeper. As a consequence of entrainment, the cell eventually reaches the new appropriate attractor and adapts to the new environment (see Figure 6).

Since a new generation network would often face environmental changes and even unexpected condition, adaptation is one of fundamental mechanisms that self-organizing network controls should have. In applying to network control, $\vec{x}$ represents control parameters or control policy. When the current control is appropriate for the environment, activity $\alpha$ reflecting the goodness of the control becomes high and the deterministic control $f(\vec{x})$ dominates the system behavior. Once the environmental condition changes and the control becomes inappropriate, activity $\alpha$ decreases and relative influence of the noise term $\vec{\eta}$ becomes dominant. The system looks for new appropriate control, i.e., a good attractor, by being driven by random and stochastic control. Eventually the system finds and reaches a new good attractor.

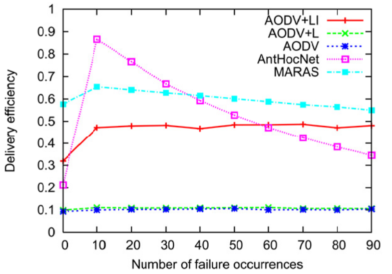


Figure 7. Performance comparison among ad-hoc routing protocols

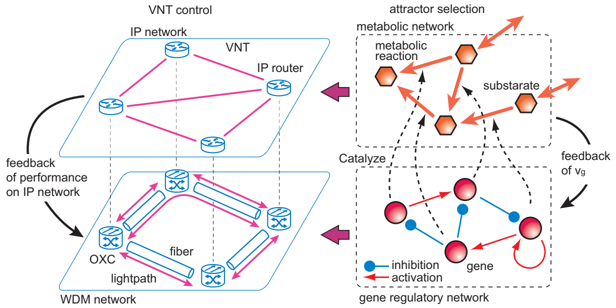


Figure 8. VNT control by hierarchical attractor selection

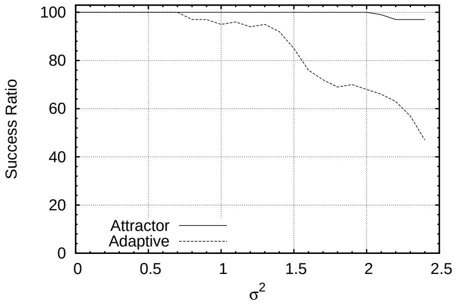


Figure 9. Adaptability of VNT control to dynamic traffic changes

An attractor selection model has been applied to multipath routing in overlay networks [15] and adaptive routing in mobile ad-hoc networks [16].

In Figure 7 a simulation result demonstrating the robustness of an attractor-selection based ad-hoc routing is shown. In our proposal, each node evaluates attractor selection equations to decide a next-hop node, to which a packet to be forwarded. Attractors correspond to neighbor nodes and the mRNA concentrations express the goodness of a neighbor node as the next hop. The activity is derived at a destination based on the number of hops that a packet travelled and it is fed back to all intermediate nodes. When a node chooses a neighbor, which contributes to establishment of a shorter path, the activity increases and a node keeps choosing the neighbor as next hop. When the path becomes longer or is broken, the activity decreases and a node begins to find a new good neighbor. In simulation experiments, 256 immobile nodes are uniformly distributed in a region of $1500 \times 1500$ m$^2$. Each node can communicate with neighbors within about 510 m using IEEE 802.11b. Between a fixed pair of source and destination nodes, a CBR session of 8 kbps, which sends 10 packets per second, is established. The x-axis corresponds to the number of failure occurance during a 1000-seconds simulation run. For example, with 10 failure occurance, each failure lasts for 100 seconds where randomly chosen 25% of nodes remain halted. At the end of the duration of 100 seconds, they resume operation and another set of randomly chosen 25% of nodes halt. In comparison to the standard AODV [17], AODV with local route repair feature (AODV+L), AODV with both local route repair and RREP response by intermediate node (AODV+LI), and another bio-inspired routing protocol AntHocNet [18], it is apparent that our proposal, i.e., MARAS, is more robust to failures than the others.

As one may notice, those models take the form of nonlinear temporal differential equations. It means that a system operating on self-organization engines always adapts to temporal changes in the environment. Adaptation is inherent in regular network control. In addition, no global information is required and each entity can determine its behavior by itself and in relation to neighbors. In the conventional approach, adaptation is implemented as an additional mechanism to regular network control. In the case of routing for example, next hop selection and routing table update are different and independent mechanisms. A routing table is updated on receiving intermittent control messages whereas next hop is selected on a per-packet basis.

*5) Layered Attractor Selection Model:* In [19], we adopt a hierachical attractor selection model of interacting a gene regulartory network and a metabolic network to virtual network topology control (see Figure 8). Genes form a gene ragulartory network of activation-inhibition relationships. A metabolic network expresses a series of production of substrates from other substrates. Chemical reaction is catalyzed by proteins, whose expression levels are controlled by genes. The dynamics of expression level of proteins is described in the form of (7), where the activity corresponds to the cell growth rate. The cell growth rate is determined as an increasing function of concentrations of substrates. A gene regulartory network adaptively and dynamically controls expression levels to achieve the high growth rate in accordance with nutrient condition.

By regarding a WDM network as a gene network, an IP network as a metabolic network, and IP-level performance, i.e., inverse of the maximum link utilization, as growth rate or activity, a WDM network adaptively and dynamically configures virtual network topology (VNT) by setting lightpaths between IP routers. Figure 9 shows a result of preliminary experiments, where the x-axis corresponds to the degree of change and the y-axis shows the probability

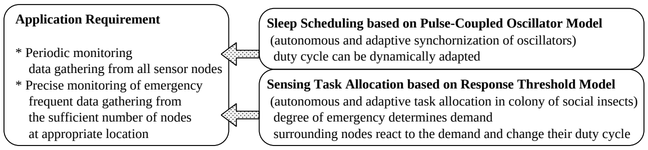

Application Requirement
* Periodic monitoring
data gathering from all sensor nodes
* Precise monitoring of emergency
frequent data gathering from
the sufficient number of nodes
at appropriate location
Sleep Scheduling based on Pulse-Coupled Oscillator Model
(autonomous and adaptive synchornization of oscillators)
duty cycle can be dynamically adapted
Sensing Task Allocation based on Response Threshold Model
(autonomous and adaptive task allocation in colony of social insects)
degree of emergency determines demand
surrounding nodes react to the demand and change their duty cycle

Figure 10. Self-adaptive application-oriented sensing

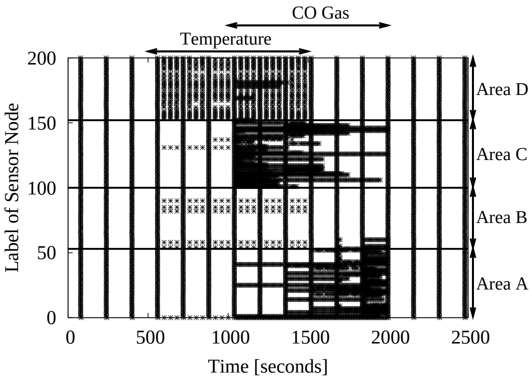


Figure 11. Adaptation of timing of message emission

that a WDM network successfully accommodates IP traffic and suppresses the maximum link utilization. As shown, our VNT control outperforms a conventional method, called ADAPTIVE, where lightpath establishment is done heuristically [20]. That is, our proposal is more robust and adaptive to dynamically changing conditions.

*6) Combination of Multiple SO Engines:* Now, we show an example of combination of multiple self-organization engines. Assume an application of periodic data gathering in a wireless sensor network consisting of a variety of sensor nodes, e.g., thermometer and CO gas sensor, in a plant. Under a usual condition, all sensors obtain and send their sensor data to a sink at the regular and same intervals. However, once an unusual event occurs, some sensors begin to report sensor data more frequently. The number of sensors for frequent sensing should be adapted in accordance with the degree of emergency. For example, temperature changes slowly in the order of hours and, once it becomes high, it stays high for a long period. Therefore, sensors are required to monitor temperature frequently when changes are detected, while they can decrease the sensing frequency under stable conditions. On the other hand, since gas existence itself is harmful, CO gas sensors should perform frequent sensing if CO gas exists.

In [21], taking remote surveillance of a shaft furnace in a steel plant as an application, we used a pulse-coupled oscillator model to accomplish energy-efficient sleep scheduling adaptive to sensing frequency, which is dynamically controlled by a response threshold model, see Figure 10. In Figure 11, we show how sensor nodes adapt their sensing frequency to dynamically changing sensing requirements. 200 sensor nodes are distributed in the monitored region, which further divided into four areas for the sake of explanation. Each cross indicates the time instant that a sensor node wakes up, obtaines sensor information, and emits a message. At the beginning, nothing happens in the region and sensor nodes report sensor information at the regular data gathering intervals of 160 seconds. At 500 seconds, temperature begins to increase in the area D. Detecting the increase, sensor nodes in the region D begins to operate more frequently as dense crosses show. At 1000 seconds, CO gas leakage is observed in the region C and it moves to the area A as time passes. Therefore, sensor nodes in the region first start frequent sensing. As the CO gas moves, those nodes that perform frequent sensing change as the movement of the dense area in Figure 11 indicates. At 1500 seconds, temperature stops increasing and stays high. Since there is no change in the temperature, sensor nodes, which adopt the high sensing frequency resume the normal operation.

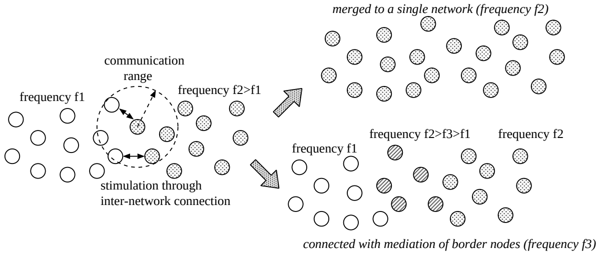


Figure 12. Fusion and connection of wireless sensor networks

### D. Intra-layer Interaction

Nodes operating on self-organization engines directly interact with neighboring nodes by exchanging messages for stimulation in a pulse-coupled oscillator model and morphogen diffusion in a reaction-diffusion model, for example. Furthermore, they indirectly interact with each other through environmental change. The autonomous behavior of node would change the environment, by consuming the bandwidth for example. In reaction to such environmental changes, other nodes would change their behavior. Such indirect interaction induced by environmental change is called *Stigmergy* [7] and it is one of important principles of self-organization. Through direct and/or indirect mutual interaction among nodes, a network is self-organized.

Physical networks and service overlay networks also interact with each other in the physical network layer and the service overlay network layer, respectively. Direct interaction among networks is accomplished by direct message exchanges or mediation of the common network layer. In case that there is no means of direct message exchanges, e.g., communication among different node devices belonging to different networks, the common layer having multiple interfaces to those networks relays messages between them. Examples of cooperative networking can be found in some

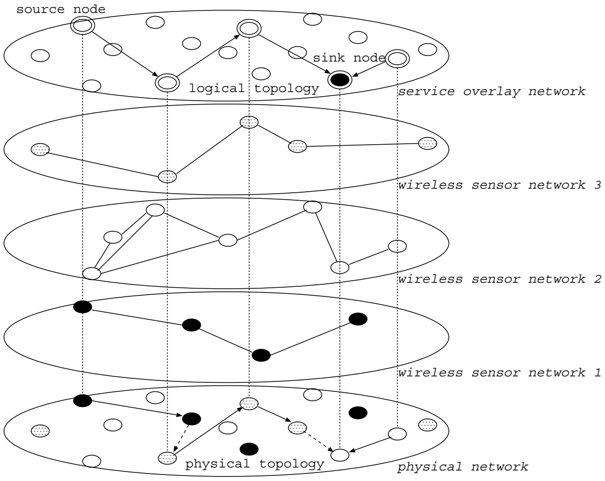


Figure 13. Layered sensor-overlay network

literatures [22], [23], [24], where networks interact with each other, they are connected with each other, and even they are merged into one depending on degree of cooperation and benefit.

For example, wireless sensor networks deployed in the same region or meeting with each other, e.g., sensor network in a room and that carried by a user entering the room, need to exchange information to provide users or applications with information or environmental control appropriate for time, place, and occasion. It is a natural assumption that they operate on the different operational frequency for energy-efficient and application-oriented control. When these networks adopt the pulse-coupled oscillator model as a self-organizing engine for frequency control, fusion, connection, and seperation of networks can easily be accomplished (see Figure 12).

### *E. Inter-layer Interaction*

Recently especially in the field of wireless network, a concept of cross-layer design has been attracting many researchers [25]. In a cross-layer architecture, each layer optimizes its behavior taking into account information and status of other layers. For example, route establishment based on the wireless link quality expressed by the received signal strength and the amount of residual energy on nodes incorporate network layer, physical layer, and even management plane.

In the self-organizing network architecture, the common network layer allows entities belonging to different layers communicate with each other in order to exchange and share control information, get feedback from the other layer, and even control the other layer. We should note here that inter-layer interaction should be kept ’loose’ not to introduce unnecessarily strong interdependency, which makes a system fragile and causes unintended consequences. For example, if an overlay network and a physical wireless network are strongly coupled with each other, the overlay network becomes too sensitive to small perturbation in the quality of wireless links and the physical topology. It changes its topology and routes actively and frequently and the resultant performance drastically deteriorates.

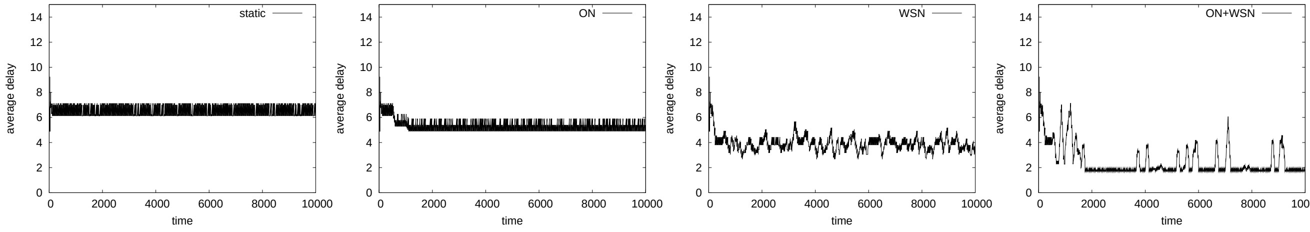
In a new generation network constructed on the layered self-organization architecture, small-scale perturbation such as local congestion, link disconnection, and node failure is handled by localized and prompt reaction of surrounding nodes. On the contrary, a network system adapts to large-scale variation, such as injection of the vast amount of traffic by flooding and spatial and simultaneous failure, by a series of reactions induced by mutual interaction among nodes and networks and spreading over the whole network, layer, and network system. Furthermore, from an inter-layer control viewpoint, influence of small-scale physical failure is absorbed in the physical network layer and hidden from the service overlay network layer. On the other hand, against large-scale physical failure, the physical network layer tries to avoid affecting the performance and control of the service overlay network layer, while the service overlay network layer adapts to changes in physical network configuration. As a result of such cooperative and self-organizing behavior, the system-level adaptability, stability, and robustness can be accomplished.

As an example of inter-layer interaction, we consider a layered sensor-overlay network. Assume that there are multiple wireless sensor networks consisting of heterogeneous sensor nodes. For the sake of energy saving, they adopt sleep control and their interval are different from each other. We consider that an overlay network is deployed over the wireless sensor networks for periodic data gathering from all or some of sensor nodes to an observatory point as illustrated in Figure 13. If all nodes involved in data gathering belong to the same wireless sensor network, the data gathering delay is the minimum. Otherwise, the delay becomes considerably large, because a node having a message to send has to wait for a next-hop node belonging to a different network to wake up. A possible way that an application can do for delay reduction without knowledge of the wireless sensor network is to adapt and find the overlay network topology leading to the minimum delay. The other adaptation in the wireless sensor network layer is synchronization. By allowing a node to addtionally synchronize with the sleep schedule of other network, the data gathering delay can be reduced very much at the sacrifice of additional energy consumption. Their adaptive behavior can be modeled by the attractor selection as,

$$\frac{d\vec{x}_O}{dt} = f(\vec{x}_O) \cdot \alpha + \vec{\eta}_O, \tag{10}$$

$$\frac{d\vec{x}_W}{dt} = f(\vec{x}_W) \cdot \alpha + \vec{\eta}_W, \tag{11}$$

where $\vec{x}_O$ and $\vec{x}_W$ corresponds to selection of overlay topology in the overlay network layer and selection of



(a) Static (b) ON (c) WSN (d) ON+WSN

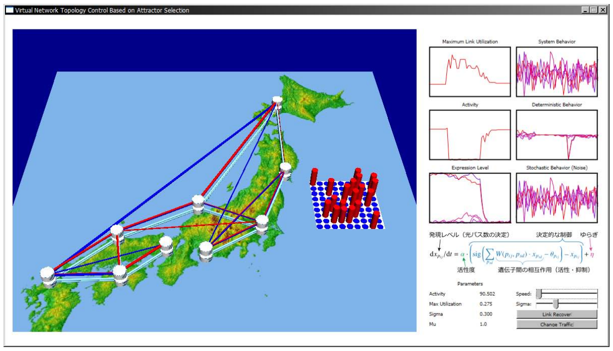
Figure 14. Average data gathering delay

schedule to synchronize in the wireless sensor network layer, respectively. These layers share the same information, i.e., activity $\alpha$, which is defined by the data gathering delay. Both layers behave in an adaptive manner to minimize the data gathering delay as a whole. We call this model sharing the same activity in multiple attractor selection-based controls the attractor composition model.

We evaluate the effectiveness of layered control based on the attractor composition model. 150 sensor nodes are randomly distributed in 200 m×200 m region. The communication range is set at 25 m. Each sensor node has its own intrinsic operational interval randomly chosen among 5, 10, and 15 minutes. That is, there are three groups of sensor nodes. An overlay network consists of one randomly chosen sink node and four source nodes randomly chosen among remaining 149 nodes. The data gathering interval is set at 10 minutes. Sensor data are obtained every 10 minutes at source nodes and wait for transmission.

We compare four different scenarios, i.e., Static, ON, WSN, and ON+WSN, depending on whether the attractor selection or composition is performed or not. In the Static scenario, the topology of overlay network is kept the same and sensor nodes follow their intrinsic operational intervals only. In the case of ON, an overlay network alone tries to minimize the data gathering delay by changing the logical topology. On the contrary, sensor nodes adaptively synchronize with other operational intervals in the WSN scenario. Finally, the attractor composition-based dynamic adaptation is performed in the case of ON+WSN.

Figure 14 shows the transient behavior of layered adaptation control, where the adaptation intervals of overlay network and wireless sensor network are set at 500 minutes and 50 minutes, respectively. Comparing the figures in Figure 14, it is apparent that the data gathering delay gradually decreases from the left figure to the right by introduction of adaptation mechanisms. In the case of ON and ON+WSN, an overlay network stays at a certain attractor after about 1000 minutes and 1500 minutes, respectively.



$$dx_{p_{ij}}/dt = \alpha \cdot \left( \mathrm{sig}\left( \sum_{p_{sd}} W(p_{ij}, p_{sd}) \cdot x_{p_{sd}} - \theta_{p_{ij}} \right) - x_{p_{ij}} \right) + \eta$$

Parameters

| | |
|---|---|
| Activity | 90.502 |
| Max Utilization | 0.275 |
| Sigma | 0.300 |
| Mu | 1.0 |

Figure 15. Screenshot of simulator

### F. Evaluation Methodology

The purpose of the self-organizing network architecture is not to improve performance in terms of conventional measures such as packet delivery ratio, response time, and throughput, but to acquire higher scalability, adaptability, and robustness than ever before. However, quantitative evaluation of such *-ties and *-ness property is not trivial.

Since self-organization engines are based on nonlinear mathematical formulas, some basic characteristics such as stability, convergence, and adaptability of each control mechanism can be theoretically or analytically discussed. For example, when we consider reaction-diffusion equations of $F(u, v) = \max\{0, \min\{au - bv + C, M\}\} - du$ and $G(u, v) = \max\{0, \min\{eu + f, N\}\} - gv$ [26], the discrete step $\Delta$ in implementing the model must satisfy $0 < \Delta t < \min\{\frac{2}{d+4D_u(\Delta x^{-2}+\Delta y^{-2})}, \frac{2}{g+4D_v(\Delta x^{-2}+\Delta y^{-2})}\}$ for a pattern to converge from mathematical analysis of a bistable reaction diffusion model. In a case of the pulse coupled oscillator model, a stimulus function $\Delta(\phi_i)$ determines the speed of synchronization, but faster synchronization results in higher vulnerability to small perturbation in timer drift. However, these analytical results are obtained for a single and independent mathematical model. Basically it is not trivial to mathematically analyse and predict interaction among different self-organizing control mechanisms and their emergent

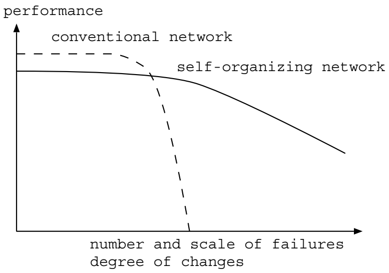


Figure 16. Comparison of self-organizing network and conventional network

behaviors. Therefore, we consider incorporating mathematical analysis for fundamental understanding of nonlinear control and simulation experiment for in-depth analysis of emergence of self-organization. For this purpose, we are now developing a novel simulator where the behavior of entities is defined by nonlinear equations and we can observe and investigate their behavior visually (see Figure 15).

Another issue is definition of range of parameters and conditions to consider. Do we need to explore the unlimited range to show the robustness against unexpected failure and condition? This still remains as future work.

## III. Discussion and future issues

In this paper, we present the self-organizing network architecture where each of node, network, layer, and network system is self-organized through intra and inter-layer mutual interaction.

Hierarchical architecture of self-* modules can also be found in autonomic computing [27] and autonomic network [28] and there are worldwide efforts for a "clean-slate" design such as [29]. Although the main goal is the same or similar, but our architecture is different from them in organizing the whole network system by self-organization principle based on nonlinear mathematical models.

Because of self-organization, each node does not need to obtain and maintain the global information and they only need to communicate with neighbor nodes to obtain the local information. This contributes to the robustness of control [30] and the scalability where the complexity of control does not depend on the number of nodes or the size of network. Since each node only needs to calculate a set of differential equations to determine its behavior, the protocol to implement is easy, simple, and lightweight.

Although our preliminary result of a specific application scenario demonstrates the superiority of our architecture, our knowledge and experience suggest that a self-organizing system is not necessarily optimal and does not always guarantee the best performance. However, we consider it is worth sacrificing performance to some extent to achieve scalability, adaptability, and robustness.

In addition to the suboptimal performance, there are some drawbacks in self-organization based control. One is that in some class of self-organization, it takes time for a system to converge and become stable. For example, as shown in Figure 3, the synchronization does not emerge at once. Therefore, the pulse-coupled oscillator model cannot be used for synchronization in the frequently changing environment, e.g., with very high mobility. Another drawback is that it would be difficult to maintain and control the whole system. Since there is no central control unit, which collects up-to-date global information, nobody knows the current status of the system. Of course it is possible to make all entities report their status to a center, it only wastes bandwidth and energy. As stated in Section I, our approach is to leave from the conventional control relying on the global or consistent information.

As Figure 16 implies, when the operational conditions stay within the expected range, a conventional and optimally designed network achieve better and even optimal performance than a self-organizing network. However, we argue that this lower performance will eventually be compensated by the advancements in network technologies, such as the increase in channel capacities and the development of new devices. Instead, we should direct our attention to the adaptability and robustness of self-organizing systems rather than their performance. In fact, self-organizing biological systems never intend to achieve the optimal performance since they slowly evolve while adapting to a dynamically changing environment. There is always some amount of spare or even idle resources left and sometimes even inefficient control can be observed. Such unused resources are the actual reason for their adaptability and robustness, and similar strategies to tradeoff between quantity and quality are also essential properties for future network technologies.

Although it is also possible for a conventional system to improve its adaptability and robustness by introducing additional and redundant network resources and more sophisticated recovery mechanisms, this would only result in slightly shifting the critical point to the right in Figure 16, which is still far below the range of adaptation capability and tolerance that a self-organizing system possesses.

## Acknowledgment

This work was supported in part by National Institute of Information and Communications Technology, Japan and "Global COE (Centers of Excellence) Program" of Ministry of Education, Culture, Sports, Science and Technology, Japan.

## References

[1] S. Paul, J. Pan, and R. Jain, "Architectures for the future networks and the next generation internet: A survey," Washington University in St. Louis, Tech. Rep. WUCSE-2009-69, October 2009.

[2] N. Wakamiya, S. Arakawa, and M. Murata, “Self-organization based network architecture for new generation networks,” in *Proceedings of the First International Conference on Emerging Network Intelligence*, Sliema, Malta, October 2009.

[3] R. E. Mirollo and S. H. Strogatz, “Synchronization of pulse-coupled biological oscillators,” *Society for Industrial and Applied Mathematics Journal on Applied Mathematics*, vol. 50, no. 6, pp. 1645–1662, December 1990.

[4] A. Turing, “The chemical basis of morphogenesis,” *Philosophical Transactions of the Royal Society of London*, vol. B. 237, no. 641, pp. 37–72, August 1952.

[5] E. Bonabeau, A. Sobkowski, G. Theraulaz, and J.-L. Deneubourg, “Adaptive task allocation inspired by a model of division of labor in social insects,” in *Proceedings of Biocomputing and Emergent Computation*, 1997, pp. 36–45.

[6] A. Kashiwagi, I. Urabe, K. Kaneko, and T. Yomo, “Adaptive response of a gene network to environmental changes by fitness-induced attractor selection,” *PLoS ONE*, vol. 1, no. 1, December 2006.

[7] E. Bonabeau, M. Dorigo, and G. Theraulaz, *Swarm Intelligence: From Natural to Artificial Systems*. Oxford University Press, 1999.

[8] O. Simeone and U. Spagnolini, “Distributed time synchronization in wireless sensor networks with coupled discrete-time oscillators,” *EURASIP Journal on Wireless Communications and Networking*, vol. 2007, 2007, doi:10.1155/2007/57054.

[9] Y.-W. Hong and A. Scaglione, “A scalable synchronization protocol for large scale sensor networks and its applications,” *IEEE Journal on Selected Area in Communications*, vol. 23, no. 5, pp. 1085–1099, May 2005.

[10] N. Wakamiya and M. Murata, “Synchronization-based data gathering scheme for sensor networks,” *IEICE Transactions on Communicatios*, vol. E88-B, no. 3, pp. 873–881, March 2005.

[11] N. Wakamiya, K. Hyodo, and M. Murata, “Reaction-diffusion based topology self-organization for periodic data gathering in wireless sensor networks,” in *Proceedings of Second IEEE International Conference on Self-Adaptive and Self-Organizing Systems (SASO 2008)*, October 2008.

[12] M. Durvy and P. Thiran, “Reaction-diffusion based transmission patterns for ad hoc networks,” in *Proceedings of IEEE INFOCOM 2005*, March 2005, pp. 2195–2205.

[13] K. H. Low, W. K. Leow, and J. Marcelo H. Ang, “Task allocation via self-organizing swarm coalitions in distributed mobile sensor network,” in *Proceedings of AAAI-04*, July 2004, pp. 28–33.

[14] T. H. Labella and F. Dressler, “A bio-inspired architecture for division of labour in sanets,” in *Proceedings of International Conference on Bio-Inspired Models of Network, Information and Computing Systems (BIONETICS 06)*, December 2006, pp. 28–33.

[15] K. Leibnitz, N. Wakamiya, and M. Murata, “Biologically-inspired self-adaptive multi-path routing in overlay networks,” *Communications of the ACM*, vol. 49, no. 3, pp. 62–67, March 2006.

[16] ——, “A bio-inspired robust routing protocol for mobile ad hoc networks,” in *Proceedings of 16th International Conference on Computer Communications and Networks (ICCCN 2007)*, August 2007, pp. 321–326.

[17] C. Perkins, E. Belding-Royer, and S. Das, “Ad hoc on-demand distance vector (AODV) routing,” RFC 3561, July 2003.

[18] G. D. Caro, F. Ducatelle, and L. M. Gambardella, “Anthocnet: An adaptive nature-inspired algorithm for routing in mobile ad hoc networks,” *European Transactions on Telecommunications (Special Issue on Self-Organization in Mobile Networking)*, vol. 16, no. 5, pp. 443–455, September/October 2005.

[19] Y. Koizumi, T. Miyamura, S. Arakawa, E. Oki, K. Shiomoto, and M. Murata, “Adaptive virtual network topology control based on attractor selection,” *IEEE/OSA Journal of Lightwave Technology*, 2008, (Conditionally accepted).

[20] A. Gencata and B. Mukherjee, “Virtual-topology adaptation for WDM mesh networks under dynamic traffic,” *IEEE/ACM Transactions on Networking*, vol. 11, pp. 236–247, April 2003.

[21] Y. Taniguchi, N. Wakamiya, M. Murata, and T. Fukushima, “An autonomous data gathering scheme adaptive to sensing requirements for industrial environment monitoring,” in *Proceedings of 2nd IFIP International Conference on New Technologies, Mobility and Security (NTMS 2008)*, November 2008, pp. 52–56.

[22] P. Kersch, R. Szabo, and Z. L. Kis, “Self organizing ambient control space – an ambient network architecture for dynamic network interconnection,” in *Proceedings of the 1st ACM workshop on Dynamic Interconnection of Networks (DIN’05)*, September 2005, pp. 17–21.

[23] E. D. Poorter, B. Latré, I. Moerman, and P. Demeester, “Symbiotic networks: Towards a new level of cooperation between wireless networks,” *International Journal of Wireless Personal Communications*, vol. 45, no. 4, pp. 479–495, June 2008.

[24] N. Wakamiya and M. Murata, “Toward overlay network symbiosis,” in *Proceedings of the Fifth International Conference on Peer-to-Peer Computing (P2P 2005)*, August-September 2005, pp. 154–155.

[25] V. Kawadia and P. Kumar, “A cautionary perspective on cross layer design,” *IEEE Wireless Communications Magazine*, vol. 12, no. 1, pp. 3–11, February 2005.

[26] S. Kondo and R. Asai, “A viable reaction-diffusion wave on the skin of pomacanthus, a marine angelfish,” *Nature*, vol. 376, pp. 765–768, August 1995.

[27] IBM Corporation, “An architectural blueprint for autonomic computing,” Autonomic Computing White Paper, 2004.

[28] “Autonomic network architecture,” available at http://www.ana-project.org/.

[29] AKARI project, “New generation network architecture AKARI conceptual design,” Report of National Institute of Information and Communications Technology, October 2007.

[30] M. Sugano, Y. Kiri, and M. Murata, “Differences in robustness of self-organized control and centralized control in sensor networks caused by differences in control dependense,” in *Proceedings of 3rd International Conference on Systems and Networks Communications (ICSNC 2008)*, October 2008.