Extension and Evaluation of Biologically-inspired Routing Protocol for MANETs

Narun ASVARUJANON†, Kenji LEIBNITZ†, Naoki WAKAMIYA†, and Masayuki MURATA†

† Advanced Network Architecture Laboratory, Department of Information Networking,
Graduate School of Information Science and Technology, Osaka University,
1-5 Yamadaoka, Suita, Osaka 565-0871, Japan
E-mail: †{narun-a,leibnitz,wakamiya,murata}@ist.osaka-u.ac.jp

Abstract Biologically-inspired systems are known for their robustness and self-adaptability to a changing environment. Therefore, some approaches for applying biologically-inspired mechanisms to routing in Mobile Ad Hoc Networks (MANETs), e.g. MARAS, have been proposed in the past. MARAS is a noise-driven routing algorithm based on the dynamics of gene expression. Parameters which reflect the path condition (e.g. path length) are used to update the routing information. This information is then used to determine the next hop where a data packet will be forwarded to. Unlike the original protocol, which focused only on the basic routing mechanism using some simplified assumptions on the packet level, we extend MARAS to fully operate within the IEEE 802.11 protocol stack in this paper. Furthermore, we investigate the performance of MARAS by simulation studies and compare its robustness to failures to that of AODV.

Key words Biologically-inspired networking, Routing protocol, Ad hoc networks

1 Introduction

These days, the access technology of information networks is shifting more and more from conventional wired networks to mobile networks. Compared to conventional networks, Mobile Ad Hoc Networks (MANETs) are more flexible as they do not rely on a fixed infrastructure and the topology is set up in an ad hoc manner. Therefore, MANETs are becoming more and more popular and are expected to play a fundamental role in a future ubiquitous network infrastructure. However, the characteristics of MANETs make routing faces many inherent difficulties, e.g. limited wireless channel capacities, multi-hop transmission with frequent topology changes, and sensitivity to interference from multiple channel access. Therefore, a robust, adaptive, and self-organizing routing protocol is required to effectively organize communication in MANETs.

As biological systems are found capable of exhibiting self-organizing behavior, there has been much attention given by researchers to utilize biological mechanisms in routing functions, which has resulted in many biologically-inspired MANET routing protocols, e.g. AntHocNet [1], BeeAdHoc [2], ANSI [3]. In this paper, we present the extension of the previously proposed MARAS [4, 5] mechanism, a biologically-inspired robust routing protocol for MANETs based on *adaptive response by attractor selection* (ARAS) [6], which is a method found in gene expression in cell biology. Our *mobile ad hoc routing with attractor selection* (MARAS) mechanism is a reactive MANET routing protocol, which probabilistically routes data packets to the destination. The goal of MARAS is to provide a seamless route recovery when problems occur during transmission. In contrast to the existing state-of-the-art MANET routing protocol—*ad hoc on-demand distance vector* (AODV) [7, 8], which performs route recovery by broadcasting messages that introduces a high level of packet collisions, MARAS has a simple self-organizing ability to find a new path without broadcasting additional control messages. However, in order for this feature to function efficiently, another unicast feedback control is used by propagating the efficiency indicator, called *activity*, of the current path selection to the source node.

Even though the objective of most routing protocols is to increase the performance in terms of throughput, we focus more on adaptability and stability in dynamic environments. Based on the simulation results, it can be observed that MARAS with delayed feedback mechanism has a higher successful delivery count than AODV in a dynamic environment.

This paper is organized as follows. In Section 2, we describe the model, components, and algorithm of our proposed protocol, and show simulation results in Section 3. Finally, we conclude the paper in Section 4 with future extension plans of this protocol.

2 Biologically-inspired Routing Protocol for MANETs

We have made a few modifications on the previously proposed MARAS in [4, 5] to achieve better self-recovery performance. As an extended version, we have adopted the same model which was used in MARAS—*the attractor selection model*, but we use a different activity definition (see Section 2. 3). A brief explanation of the attractor selection model is provided in the Section 2. 1 and further details can be found in [6]. Additionally, we define the feedback packet routing mechanism, which has not been proposed in the previous version of MARAS, and give suggestions on how to limit the next hop's candidate list as future work.

2. 1 Attractor Selection Model

The attractor selection model is inspired from the behavior in cell biology where gene networks adapt to new environment conditions and finally reach a stable state, called an attractor, even in an unknown and dynamic environment. The dynamics of gene expression are formulated by the following differential equations:

$$\frac{dm_i}{dt} = f(m_1, \ldots, m_M) \times \alpha + \eta_i \qquad i = 1, \ldots, M, \quad (1)$$

which leads the M-dimentional system state to converge to attractors. If the system state is moved away from attractors, the cell's state vector of mRNA concentrations $\vec{m} = (m_1, m_2, \ldots, m_M)^T$ will shift to a new attractor by the effect of the *noise* $\vec{\eta} = (\eta_1, \ldots, \eta_M)^T$ until it once again reaches one of the attractors.

Another important parameter in the attractor selection model is *activity* α. The activity reflects how well the current system state is performing in the environment conditions and adjusts the influence of randomness accordingly. When the current state is far from the suitable attractor, the activity will be low and there will be a larger effect from internal and external noise, making it easier for the system to switch from one attractor to another. Once the system reaches a suitable attractor, the activity will be high and the effect of noise will be suppressed, which then allows the system to become stable again.

In Fig. 1, we show the general principle of the attractor selection concept. The x-axis shows the state m, the y-axis is the activity α, and the z-axis indicates the energy potential defined by $f(\vec{m})$. The current system state is illustrated as a circle which is constantly in motion due to the effect of the noise. It can be observed that when the activity is high, changing the system's state would be difficult because of the steepness of the potential landscape. On the other hand, when the activity is low, the landscape becomes smoother and changing the state can be achieved by the effect of noise.

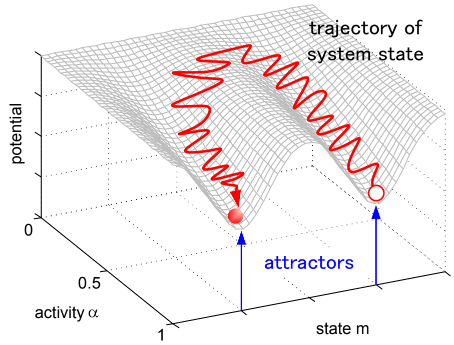

Fig. 1 General principle of the attractor selection concept

Based on the attractor selection model, we map the mRNA concentration vector to the *neighbors' selection probability vector* which reflects the approximate effectiveness of how each neighbor node delivers the data packet to the destination. We also map the attractor to the state where the path between source and destination is established which includes a full recovery from path failure. We therefore have to design a good method to effectively utilize noise and at the same time define an appropriate activity which will lead the system nearer to an attractor and suppress noise once the attractor is reached.

The concept of having noise in the system may look undesirable. However, adding noise into the system makes it in general more robust to external noise. Moreover, noise gives the system a chance to leave local minima while searching with the random-walk strategy as explained in [9].

2. 2 Mathematical Model

For the mathematical model, we use the following differential equation from [4] for each neighbor i,

$$\frac{dm_i}{dt} = \frac{s(\alpha)}{1 + m_{max}^2 - m_i^2} - d(\alpha) m_i + \eta_i, \qquad (2)$$

where $m_{max} = \max_{j=1,\ldots,M}(m_j)$, $s(\alpha) = \alpha[\beta\alpha^\gamma + \varphi^*]$, $d(\alpha) = \alpha$, and $\varphi^* = 1/\sqrt{2}$.

Using Eqn. (2) independently at every node, the neighbor's selection probability will keep changing until the system reaches an equilibrium at $dm_i/dt = 0, \forall i$. While the value is changing and the next hop is selected nondeterministically, the system is in a *random-walk* state. The degree of randomness changes over time and depends on the current activity. If the activity is high, then the effect of noise is suppressed and less randomness is introduced to the routing function. On the other hand, if the activity is low, m_i will decrease and a relatively higher effect from noise will increase the degree of randomness. At the equilibrium, the vector consists of one maximum high value and $M - 1$ low values which is the desired condition for next hop selection in deterministic routing.

2. 3 Activity Definition

In the previously proposed MARAS [4], the activity is defined by the delivery ratio at destination. However, the delivery ratio relies too much on information from the past which causes inaccurate perception of the current network conditions. Therefore, we propose a more dynamic activity as follows.

Let W be the window containing the travelled hop count information of n arrived packets at the destination in the last T seconds, sorted by arrival order where w_n is the newest arrived packet's travelled hop count. At the destination, the activity $\alpha \in [0, 1]$ at time $t + \Delta t$ is updated as follows:

$$\alpha(t+\Delta t) = \begin{cases} \alpha_{new} & \text{if } \alpha(t) \leqq \alpha_{new} \\ \alpha(t) + c\,(\alpha_{new} - \alpha(t)) & \text{otherwise,} \end{cases} \tag{3}$$

where

$$\alpha_{new} = \frac{\min_{\forall w_i \in W} w_i}{w_n}. \tag{4}$$

The α at time $t+\Delta t$ is updated according to the difference between the newly calculated α_{new} and α at time t. Partly similar to [5], α_{new} is calculated from the ratio between the minimum travelled hop count of all packets in the queue and the travelled hop count of the latest arrived packet. When the α_{new} is decreased, it can be assumed that it is caused by a problem occurring in the network, faulty information, or outdated information. Therefore, to prevent the activity from decreasing in a sudden manner caused by faulty or outdated information, parameter c is used to gradually decrease the activity. On the other hand, when the α_{new} is increased, it is safe to assume that the path which the latest packet has travelled on is the better path and it can therefore be used immediately for the purpose of fast path recovery.

With the new activity definition, the extended MARAS operates with only the last T seconds information which makes it more dynamic than the previously proposed MARAS. However, the parameter T has to be chosen carefully because it has to be large enough to sense a change in the network (i.e. a link failure or a link recovery), but also small enough to avoid using outdated information. Currently, we use an empirical value of T, but we also wish to investigate the system behavior according to T in future work.

2. 4 Routing Table and Routing Vector

In MANET, each node can perform both terminal and router roles. Therefore, each node has to maintain its own routing table. In our protocol, each node sets up a route entry in the routing table for each source and destination pair only when the node becomes a part of the session. In short, the route entry is set up reactively.

Each entry of our routing table consists of:

(1) A destination address

(2) A source address

(3) A neighbors' selection probability vector, i.e. a routing vector $\vec{m} = (m_1, m_2, \ldots, m_M)^T$

(4) An activity α for this source and destination pair

(5) The address of the last node which sent the data packet to the destination via this node

A routing vector contains *MARAS state values* which reflect each neighbor's probability of delivering data packets to the destination. The MARAS state values are changed over time (periodically every τ seconds) by the effects of noise and function of activity in Eqn. (2), and are mainly used for routing the data packets. The address in element (5) is used in the feedback mechanism which will be explained later in Section 2. 7. 1.

2. 5 Data Packet Forwarding

In MARAS, a probabilistic routing function is used to forward the data packets to the destination. The next hop of the data packet is selected based on the MARAS state value. The MARAS state value of each neighbor in the routing vector is normalized and used as the probability for that neighbor to be selected as the next hop. Similar to the previously proposed MARAS, the candidate list concept is also used in this new MARAS where only the node which is relatively closer to the destination than the current node will be selected. Currently, we assume that such information is provided to MARAS. However, we plan to include an approximation method as a further extension to MARAS.

The key idea of probabilistic routing is that the highest probability node will not always be selected which gives MARAS the self-recovery ability. In the random-walk process, MARAS will continuously attempt to select next hop nodes until a sufficiently short path to the destination is found which then makes the activity becomes higher and the better next hop is selected more frequently compared to the others. As explained in Section 2. 2, the MARAS state values of neighbors will be eventually separated into one high value and $M - 1$ low values.

2. 6 Route Establishment

A similar approach to AODV [7] is used for the route establishment phase. We adopt the broadcasting route discovery mechanism from AODV and make a few modifications. In our protocol, the *route-request packet* (RREQ) is broadcasted from the source node and flooded until it reaches the destination. Every RREQ packet has a unique ID in order to avoid forwarding any duplicated RREQ packet. The previous hop of a valid RREQ packet is remembered for sending the reply back to the source if the reply packet is forwarded via the current node.

When the RREQ packet reaches the destination, a *route-reply packet* (RREP) is generated. As the reverse path for the RREP packet is remembered, it is forwarded in unicast manner to the source. On reception of the RREP packet at any intermediate node, that particular node sets up or updates the activity and the route entry for this session. The activity value will be set to highest and the routing vector's state values will be changed to one high value for the previous hop of the RREP packet and 0 for the rest. Subsequently, the RREP will be forwarded again via the remembered neighbor. At the source node, after updating the activity and the routing vector in the same manner to an intermediate node, the data packet forwarding begins.

On reception of a data packet, if the current node has no route entry for that session, then it will set up a new random vector which contains equal MARAS state values λ for every neighbor and starts the random-walk mechanism.

2.7 Route Maintenance

After the route connecting source to destination has been successfully established, a route maintenance is performed to detect undesirable conditions and recover the route from failures.

2.7.1 Route Updating by Feedback Activity

As updating the routing vector requires the current activity of the session, a *feedback packet* with the current activity $\alpha(t+\Delta t)$ embedded is sent from the destination to the source every time a new activity is calculated. In other words, the feedback packet is sent back every time a data packet arrives at the destination.

On reception of a feedback packet at any node, the activity is updated by using the Eqn. (3). According to Eqn. (2), the routing vector should be updated in realtime over a continuous time. However, due to the practical limitations, it is difficult to perform a realtime simulation. Therefore, an event-based (timer-based) simulator is used and the routing vector is updated every specific time interval τ. In short, for simplicity, the update mechanism is performed regardless of the feedback packet arrival.

Unfortunately, as the routing vector is used only for forwarding data packets to the destination, the feedback packet which is sent from the destination to the source is not able to utilize the routing vector. Therefore, an alternative mechanism is required for the feedback packet forwarding. In the previously proposed MARAS, the feedback packet is assumed to be sent back via the same path the data packet has travelled on. To achieve such behavior, each node remembers the last neighbor which forwarded the data packet via it. When the data packet arrives at the destination, the feedback packet will be sent via that remembered neighbor. Considering the short delay, this remembered path is considerably the same path to the data packet's path.

2.7.2 Activity Decay

The reasons why it is necessary to decay activity on each node can be explained as follows:

(1) When the route is not used for a long time we can assume that the route is no longer a suitable route for the current session. Therefore, the previously learned MARAS state values need to be changed to another attractor. In order to switch to the other attractor, the activity must be decreased to allow random-walk mechanism to perform.

(2) As the feedback packet is sent only when the data packet arrives at the destination, it can be concluded that if no data packet arrives at the destination then the activity will never be updated. In order to recover from such situations, the activity on each node must be decayed over time.

In our protocol, we use the simple activity decay equation on the stored activity: $\alpha_{decay} = \delta\alpha_{stored}$ where δ is the decay rate over interval τ which is the same interval we used to update the routing vector. Similar to the update mechanism, the activity decay mechanism is performed regardless of the feedback packet arrival. Therefore, when there is no incoming feedback packet, the activity will continuously be decayed and the routing vector will be updated by using the decayed activity.

2.7.3 Local Connectivity Maintenance

In MARAS, the routing vector consists of the local neighbor list and the corresponding MARAS state values. When the connectivity to a neighbor node is lost, the related MARAS state value is also lost. As the list of neighbors plays a significant role in MARAS, we need to maintain the connectivity with the neighbors as long as the neighbor node is in range and remains active.

In our protocol, we adopt the HELLO packet mechanism from AODV [7] where every node broadcasts the HELLO packet periodically to notify its neighbor of its existence. When a node does not receive a HELLO packet from one of its neighbors for a certain period of time, that neighbor is considered lost and then removed from the neighbor list. With this mechanism, we can maintain the neighbor list and tolerate to some transmission failures of HELLO packets. However, an explicit local route repair mechanism of AODV is not adopted in MARAS.

3 Evaluation

We evaluate MARAS by performing simulation in a commercial network simulator—*QualNet*. In QualNet version 4.0, AODV draft 8 [7] with extensions from draft 9 [8] is used. We compare MARAS to AODV with some constraints: (1) only the destination can issue a RREP, and (2) a local route

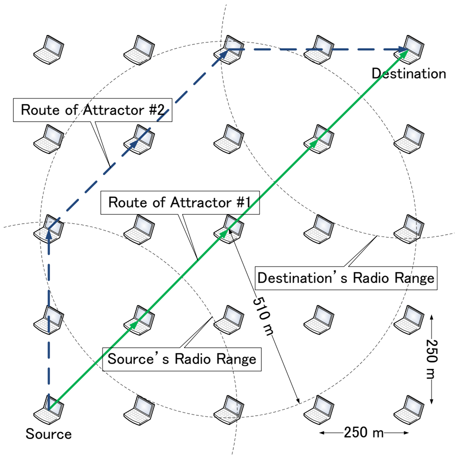

Fig. 2 Attractor and neighbor selection concept is shown over an evaluation scenario

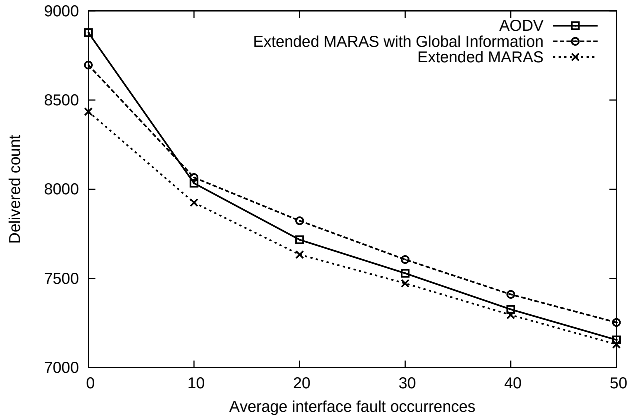

Fig. 3 Delivery count vs. average interface fault occurrences between extended MARAS and AODV

Table 1 MARAS parameters in simulation

Parameter	**Value**
High value in $s(\alpha)$ calculation β	1000
The exponent of α in $s(\alpha)$ calculation γ	3
Window interval T	5.0 s
Activity update coefficient c	0.1
Initial random-walk vector's state value λ	0.5
Decay rate δ	0.9
Update interval τ	1.0 s

repair is not used.

3. 1 Simulation Settings

The scenario for evaluation is set as in Fig. 2. There are 25 nodes placed in a 250m×250m grid. Each node uses an IEEE 802.11b wireless module with an estimated radio range of 510 meters in a free-space model without fading. The data rate used in IEEE 802.11b is 2Mbps. With regard to the traffic, we use the constant bit rate (CBR) application with UDP as a transport-layer protocol. The bottom-left source node initiates CBR traffic to the top-right destination node. We use CBR bitrate of 8 kbps which sends out 10 packets per second. The simulation time is 1000 seconds where the transmission of traffic starts at 0 s and ends at second 900 s. The last 100-second interval is spared for delayed packets.

The MARAS parameters are set as described in Table 1. Other related parameters of MARAS and AODV are the default values of QualNet 4.0 implementation. Moreover, to simulate a dynamic environment in QualNet, we use *interface fault* configuration of QualNet which turns off the MAC layer interface. We perform simulations with various frequencies of interface faults at each node from 0 to 50 occurrences in the whole simulation duration. The results in the next section are the average values from the results of 500 simulation runs.

3. 2 Delivery Efficiency Result

Using the above mentioned scenario setting, we compare MARAS to AODV in term of delivery efficiency. First, we perform simulations of our proposed version of MARAS and AODV. The result is shown in Fig. 3. In Figs. 3 and 4, we show number of successfully delivered packets at the destination (shown as delivery count in y-axis) over the change of the degree of dynamics in the system (shown as average interface fault occurrences in x-axis).

It can be observed that our extended MARAS, indicated as “Extended MARAS” in the figure, has lower delivery efficiency than AODV. However, MARAS is less sensitive to the dynamic environment as the tendency of the MARAS curve is decreased slower than that of AODV.

We believe that MARAS performs worse because of the effect of feedback packets. Therefore, to confirm our assumption, we perform another simulation of MARAS, called *global information* scenario, where no feedback packet is sent and all nodes can learn the current activity immediately via a separate channel. According to the simulation result, MARAS in ideal case (“Extended MARAS with Global Information”) has higher performance and better tolerance to the dynamic environment than AODV. Moreover, our assumption that the lower efficiency of MARAS is the result of feedback packets' effect is confirmed.

3. 3 Delayed Feedback and Delivery Efficiency

After it is confirmed that MARAS's efficiency and feedback mechanism have a direct relation, we reduce the number of feedback packets to shift the efficiency curve closer to the ideal scenario. The number of feedback packets is reduced by sending the feedback packet after the arrival of every n data packets. This mechanism is called *delayed feedback mechanism.* The results of this scenario with parameter $n = 10, 50, 100$ are shown in Fig. 4, where y-axis shows the normalized delivered count based on AODV.

Based on the simulation results, the efficiency of MARAS

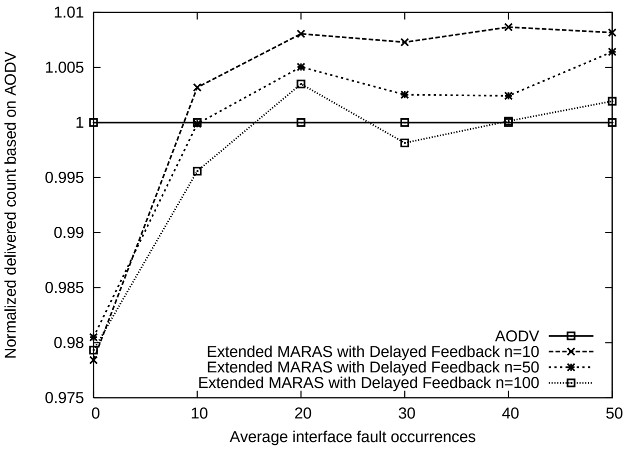

Fig. 4 Normalized delivery count based on AODV vs. average interface fault occurrences between extended MARAS with delayed feedback mechanism and AODV

is improved by using the delayed feedback mechanism as expected. With delayed feedback parameter $n = 10$, the efficiency of MARAS is improved the most among the 3 values of parameter n and the efficiency is degraded as n is increased. This behavior can be explained by the trade-off between the validity of feedback information and the level of network load. When $n = 1$ (as shown in Fig. 3), the feedback is immediately sent to the source which provides the most valid information, but also causes too much load to the network. On the other hand, when $n = 100$, fewer feedback packets are sent which reduces the network load. However, when too much feedback is skipped, the activity at each node becomes outdated and the protocol's efficiency deteriorates.

Based on the simulation results, the good trade-off parameter is $n = 10$ which still maintains the activity's validity and does not consume a too large portion of the network capacity. With delayed feedback parameter $n = 10$, MARAS shows its ability to achieve higher delivery count than AODV in the considered dynamic scenarios.

4 Conclusion and Future Work

In this paper, we present a robust, adaptive, and self-organizing biologically-inspired mobile ad hoc network routing protocol. The protocol is based on a cell biology attractor selection mechanism and inherits its ability to react to a dynamic environment. Simulation results show that the proposed protocol with appropriate parameter settings can achieve higher delivery efficiency than AODV in the considered dynamic scenarios.

In the current study, we have performed simulations only in a static grid topology with active/inactive state changes. In the future, we would like to perform more simulations in various scenarios with concurrent traffic sessions. In addition, a study of appropriate parameters is necessary. Besides the delayed feedback parameter, we still have update interval τ, queue length T, high-value β, etc. which may play important roles relating to the performance of MARAS. Also, the evaluation metrics should not be limited to the delivery efficiency. We definitely would like to study more about the overhead comparison between MARAS and AODV.

As previously mentioned, we are planning to include a candidate limiting mechanism into MARAS. The principle of this mechanism is to utilize the HELLO packet for updating the estimated hop count to destination [10]. With this method, MARAS does not require external information and is likely to result in a further improvement in performance.

Acknowledgment

This research was partly supported by the "Global COE (Centers of Excellence) Program" of the Ministry of Education, Culture, Sports, Science and Technology, Japan.

References

[1] G. Di Caro, F. Ducatelle and L. Gambardella: "AntHocNet: An adaptive nature-inspired algorithm for routing in mobile ad hoc networks", European Transactions on Telecommunications, **16**, 5, pp. 443–455 (2005).

[2] H. Wedde, M. Farooq, T. Pannenbaecker, B. Vogel, C. Mueller, J. Meth and R. Jeruschkat: "BeeAdHoc: an energy efficient routing algorithm for mobile ad hoc networks inspired by bee behavior", Proceedings of the Genetic and Evolutionary Computation Conference (GECCO), Washington DC, USA, pp. 153–160, June (2005).

[3] S. Rajagopalan and C. Shen: "ANSI: a swarm intelligence-based unicast routing protocol for hybrid ad hoc networks", Journal of Systems Architecture, **52**, 8-9, pp. 485–504 (2006).

[4] K. Leibnitz, N. Wakamiya and M. Murata: "A bio-inspired robust routing protocol for mobile ad hoc networks", Proceedings of 16th International Conference on Computer Communications and Networks (ICCCN), Honolulu, HI, USA, pp. 321–326, August (2007).

[5] K. Leibnitz, N. Wakamiya and M. Murata: "Self-adaptive ad-hoc/sensor network routing with attractor-selection", Proceedings of IEEE Global Telecommunications Conference (GLOBECOM), San Francisco, CA, USA, pp. 1–5, November (2006).

[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, **1**, 1 (2006).

[7] C. Perkins, E. Royer and S. Das: "Ad hoc on-demand distance vector routing, draft-ietf-manet-aodv-08.txt", IETF MANET Working Group (2001).

[8] C. Perkins, E. Royer and S. Das: "Ad hoc on-demand distance vector routing, draft-ietf-manet-aodv-09.txt", IETF MANET Working Group (2001).

[9] B. Selman, H. Kautz and B. Cohen: "Noise strategies for improving local search", Proceedings of the national conference on artificial intelligence, Seattle, WA, USA, pp. 337–343, July (1994).

[10] E. Sakhaee, N. Wakamiya and M. Murata: "An energy-efficient self-organizing global extremity reporting scheme for sensor networks", Wireless Communications and Mobile Computing, **9**, 4 (2009).