Robustness in Sensor Networks: Difference Between Self-Organized Control and Centralized Control

Yuichi Kiri
Graduate School of Information Science and Technology
Osaka University
1-5, Yamadaoka, Suita-shi
565-0871 Osaka, Japan
y-kiri@ist.osaka-u.ac.jp

Masashi Sugano
School of Comprehensive Rehabilitation
Osaka Prefecture Univ.
3-7-30, Habikino, Habikino-shi
583-8555 Osaka, Japan
sugano@rehab.osakafu-u.ac.jp

Masayuki Murata
Graduate School of Information Science and Technology
Osaka University
1-5, Yamadaoka, Suita-shi
565-0871 Osaka, Japan
murata@ist.osaka-u.ac.jp

***Abstract*—Self-organized control has received significant attention in the area of networking, and one of the main factors for this attention is its robustness. However, it should be stressed that deciding whether self-organized control is robust or not is not a trivial task. Even if it is in fact robust, the factors underlying its robustness have not yet been explored in sufficient detail. In this paper, we provide the first quantitative demonstration of the superior robustness of self-organized control through comparison with centralized control in a sensor network scenario. Through simulation experiments, we show that self-organized control maintains the functionality of its data collection even in a variety of perturbations. In addition, we point out that the difference in the robustness of the abovementioned control schemes stems from the degree to which the comprehension of a given node about the state of the network depends on information obtained from other nodes.**

***Keywords*—sensor network; self-organized control; centralized control; robustness; simulation**

I. Introduction

As networks are becoming increasingly larger and more complex, a critical issue in today's dynamically changing and uncertain environments is to maintain the functionality of networks in a manner which allows them to adapt to environmental changes. A control scheme which maintains the performance even when the network state changes dramatically or unforeseeable circumstances occur is preferable for present and future networks, even if the basic network performance in such cases is inferior to that of networks operating with other control schemes. The property which allows a system to maintain its functionality despite external and internal perturbations is called "robustness" [15]. In this age when networks play an essential role in our everyday lives, the robustness of networks is becoming increasingly important.

Distributed control has been said to be superior to centralized control with respect to robustness. Currently, a type of distributed control scheme which is beginning to attract considerable attention is one of self-organized control [10][20]. The communication networks based on such a self-organized control are considered to be suitable as a network which consists of movable nodes like many persons or cars, and a network used in the situation where environmental variation is remarkable, like disaster sites. In this control scheme, each component autonomously decides the following action on the basis of local information, and the simple microscopic actions of the components collectively provide structure and functionality at macroscopic level without any centralized coordination [19]. Such behavior is distinct from plain distributed control, where individual components act autonomously but depend on global information. Although scalability, adaptability, and fault tolerance, which are included in the concept of robustness in a broad sense, are "known" as properties inherent to self-organized control, we stress that this knowledge is certainly not trivial. Even assuming that the notion of robustness is true, to the best of our knowledge the reasons why self-organized control is robust and the factors which determine the superiority of its robustness as compared to other control schemes have not been examined with sufficient rigor.

In our previous work [13][14], we provided quantitative evidence of the robustness of self-organized control with respect to transmission errors and node failures, and concluded that the robustness of the self-organized control scheme is superior to that of other control schemes. However, since sensor networks face a wider range of perturbations, the purpose of this paper is to demonstrate the advantages of self-organized control against perturbations different from those in our previous work. Furthermore, based on the results of the evaluation, we also pose interesting questions such as why and how self-organized control is robust. In [21], from the results of the comparison, we pointed out that the difference in the robustness is derived from the degree to which the comprehension of a given node about the state of the network depends on information from other nodes. This is the key to differentiating the degrees of robustness of those two control schemes. In this paper, we describe the details of each method which were not able to be described in [21]. Furthermore, we show the characteristic of self-organized control by distribution of the number of hop of routes, and present the difference in the robustness of each control method against bit error.

The remainder of the paper is organized as follows. In Section II, earlier approaches to self-organized control are reviewed. Sections III describes the mechanisms of centralized

control and self-organized control, respectively. Section IV presents the simulation results so as to compare the robustness of both control approaches. In Section V, we discusses what brings robustness to self-organized control on the basis of these results. The paper is concluded in Section VI and discusses the generalization of our conclusions.

II. Related work

The principle of self-organization is developed in nature [8], and we can find it everywhere. Each component autonomously decides its next action on the basis of local information, and the microscopic simple actions of the components collectively provide structure and functionality at the macroscopic level without any centralized coordination [19]. Such self-organized behavior is disparate from the distributed paradigm where individual components act autonomously while sharing global information, and many researchers have tried to derive the advantageous properties of the self-organizing system in efforts to solve scalability, reliability, availability, and robustness problems. For example, Directed Diffusion [12] is a well-known self-organization paradigm for certain novel features, including reinforcement-based adaptation of the gradient to the empirically best path. It is also known to be robust against node failures. [9] is proposed to achieve good adaptability and scalability by endowing mobile agents with simple intelligence. Some researchers further this approach and incorporate the behavior of social insects into the agents. BiSNET [4], which was shown to have strong self-healing capability for false positive data in data gathering, are examples that were inspired by the foraging principles of honey bees, while [16][5][25] are inspired by the Ant colony metaheuristic and said to be robust against node mobility. ACE [6] is an emergent algorithm that forms clusters through three rounds of feedback between nodes. Using local information alone, it efficiently covers the network with only a small amount of overhead. Ant-based clustering [11][24][22] is also a clustering method, drawing its inspiration from the behavior of ant colonies, but it is applied for data analysis. In addition, the task allocation method proposed in [17] uses the concepts of the "division of labor" of ants to achieve higher coverage in sensor network.

III. Centralized and Self-organized control schemes in sensor networks

We provide detailed explanation of our centralized and self-organized control schemes, which are the subjects of robustness evaluation in the present study. The operations of both control schemes are based on the premise that multiple sinks are deployed in their respective monitoring regions. Using this multi-sink configuration, both control schemes take a cluster-based approach, in which the same number of node clusters and sinks is formed, and individual sensor nodes transmit their sensed data to the sink located in their cluster (Figure 1).

A. Centralized control

Younis *et al.* [23] proposed a data-gathering scheme for sensor networks that assumes the existence of multiple

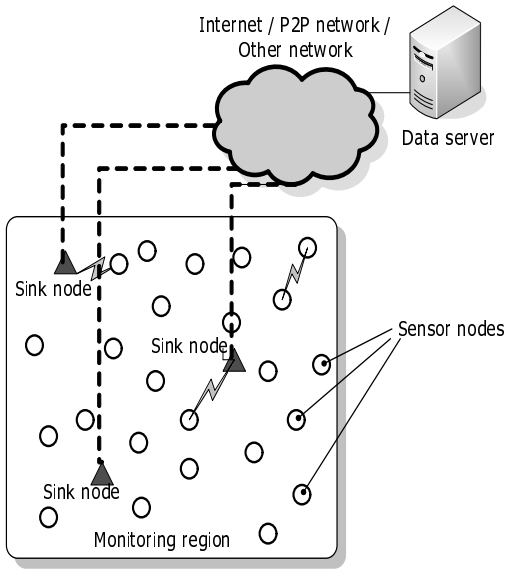


Fig. 1. Network model.

sinks (for consistency with the terminology used in our self-organized control [13], we use "sinks" here instead of the "gateway nodes" used in [23]). Sinks are significantly less energy-constrained than sensor nodes and the sensed data is gathered first in them. Sensor nodes are divided into the cluster which each sink manages, and the sinks calculate the route from each sensor node to themselves based on the residual power, state, etc. of a sensor node. They then tell their cluster members their previous- and next-hop nodes and the state they should stay in next (e.g., active or sleep state). In this data-gathering scheme the role of the clusters is almost same as that of the clusters in the scheme described in [13] — in both the cluster determines the eventual destination to which data packets are sent — so these two schemes are well-suited to be compared. Younis *et al.* [23], however, describe only the routing and node-state management and do not specify how the sensor nodes should be apportioned into clusters. In addition, some of its assumptions, for example, that each sink is located within the one-hop of all the sensor nodes in its cluster, are not appropriate for large-scale sensor networks. So we made some modifications to the proposed mechanism in order to make a convincing comparison.

We assume the existence of a control station, which is wired to all sinks. The station knows the initial power and locations of all nodes and sinks, and manages the overall network. Up-to-date residual power is reported periodically from sensor nodes, but the reporting packet is forwarded to the sink in a multi-hop fashion instead of direct communication. The station first divides the sensor nodes into as many clusters as there are sinks. The role of a cluster is to determine the destination sink for each sensor node, and we say that "sensor node n_i belongs to cluster S_j" if n_i transmits their sensing data to the destination sink S_j. The clustering method used is same as Voronoi tessellations using locations of sinks as basing points.

In other words, the central station splits the sensor nodes into clusters in such a way that each sensor node transmits packets to the nearest sink.

After clusters are determined, the station constructs routes for packets. As described in [23], the routes are determined by using Dijkstra's algorithm to minimize the total link cost. Link cost is assigned by the station beforehand to all the links between all node-and-node, node-and-sink pairs. Calculation of link cost is modified from [23] due to difference of assumptions, and the cost C_{ij} of the link between node n_i and n_j is defined by residual power of the node and the distance between them:

$$C_{ij} = \begin{cases} \frac{E_{I_j}(4\pi)^2 d(n_i,n_j)^2}{E_{R_j}\lambda} & \text{if } d(n_i,n_j) \le \delta \\ \frac{E_{I_j} d(n_i,n_j)^4}{E_{R_j} h^4} & \text{if } \delta < d(n_i,n_j) \le r_{\max} \\ \infty & \text{if } r_{\max} < d(n_i,n_j) \end{cases} \quad (1)$$

where E_{I_j} and E_{R_j} are respectively the initial and residual powers of node n_j, λ is the radio wavelength, h is the height of the antenna, and $d(n_i, n_j)$ is the distance between nodes n_i and n_j. The threshold value δ is a constant defined as $\delta = \frac{4\pi h^2}{\lambda}$, and $r_{\max}$ is the communication range of a sensor node.

After route construction is finished, the central station transmits the route information to sinks. For the sake of simplicity, packets which includes the route information are called "command packets" hereafter. The sink uses minimal transmission power when transmitting the command packet so that all the sensor nodes in its cluster can receive them. Command packet provides following information to sensor node n_i.

- Cluster to which n_i belongs.
- The previous-hop node from which n_i receives a packet and the next-hop node to which n_i should transmit a packet.

The detection of node failure is based on a soft state model. Each sensor node transmits a hello message at a regular interval t_{hello}. On receiving the hello message from a neighboring sensor node n_i, sensor node n_j registers entry of n_i to its neighboring node table and interprets the reception as a sign that n_i is working properly. Every time n_j receives a hello message from n_i, expiry-time field in the entry is updated to the sum of t_{expire} and the value of n_j's internal timer. Only if n_j's timer exceeds the value of expiry-time field, n_i is deemed to have failed, and n_j sends a failure-indication packet to its sink. This packet passes through the same route which the station calculated for data packets, and it reaches the sink. The sink passes the failure-indication packet to the station, which then recalculates new routes that circumvent the failed node. New routes are packed in a command packet and transmitted from the sink to sensor nodes.

Even when n_i works normally, hello packets from n_i might not arrive within the expiry time because of interference or transmission error. This possibility must be allowed for, because the accumulation of such false positives would cause a virtual connectivity problem limiting network performance. Preparing for such a false detection, node n_j memorizes an ID of the failed node when detecting the failure. And if n_j could receive a hello packet from n_i, it deems the detection of n_i's failure to have been false positive, and transmits a failure-recovery packet to inform the station about that. The sink relays it to the station, and the station recomputes new routes and distributes them to sensor nodes.

In this centralized control, sink-failure can be easily detected because of the assumption that sinks and the central station are linked with wire. By keeping track of sinks' status, the station can recompute clusters and routes just after the sink failure. It does not need to take explicit measures, and all it has to do is to transmit a command packet containing new cluster organization and route information as usual. Reliable communication can be readily provided in wired networks. So we ignore the possibility of false detection of sink failure.

B. Self-organized control

We have proposed a bio-inspired control which shows a self-organized property [13]. Our self-organized control approach is based on pheromone-mediated ant-swarm behaviors called ant colony optimization (ACO) [3] and ant-based clustering [11][24][22]. Sensor nodes are divided into as many clusters as there are sinks by using ant-based clustering with a virtual "cluster pheromone," and routing is performed in each cluster by using "routing pheromone." The detailed operation for our proposal is given in the following.

ACO is a probabilistic approach inspired by ants in their foraging activity to combinatorial optimization problems like the traveling salesman problem [7]. Ants follow efficient routes to their food by being attracted to higher concentrations of pheromones left by other ants. An ant will leave a volatile pheromone trail while carrying food back to the nest. If another ant finds the trail before it dissipates, that ant will follow it to the food and it too will leave pheromone on the way back, reinforcing the trail. If there is enough food that several workers can bring food back to the nest, a high pheromone concentration will be maintained and even more ants will be attracted. As the food supply becomes smaller, fewer ants will be attracted and the trail will gradually disappear as the pheromone evaporates. This positive-feedback trail building is the basic idea behind the ACO approach, and ACO has been applied to some of the routing problems.

We have also applied the principle of ACO to hop-by-hop routing in our proposed scheme. Each sensor node has a pheromone table, and the advantages of neighbors as a next-hop node are stored in the form of routing pheromones. When a sensor node transmits a packet to notify the sink of obtained data, it refers to its pheromone table, and stochastically selects the next-hop node leading to the sink based on the routing-pheromone value. Thus, each sensor nodes selects a next-hop node with greater probability of having more routing pheromones (sensor node with more routing pheromones means preferable next-hop node). Furthermore, if some neighboring nodes have almost the same routing-

pheromone value, they are selected as next-hop nodes with almost the same frequency, and the number of packets that must be relayed is distributed among them.

An important problem of applying ACO to routing is how to determine which route should have higher routing-pheromone value, in other words, how to define what are the "preferable routes" in a given network. We define good routes in sensor networks as follows:

- routes with a small hop count on the way to a sink.
- routes that go through sensor nodes with high residual power.

It is not necessary for each node to send packets (ants) in order to find good paths to the destination as some ant-based routing employed [5][25][2]. Such strategies could cause unnecessary power consumption and needlessly occupy wireless channels, because of ants traveling back and forth over the network. Thus, we chose sinks to flood the ants, which we call backward ants. Backward ants do not go back into the sink. As we previously pointed out, the required next-hop node is a sensor node located nearer to the sink, which has enough residual power. With that in mind, the role of backward ants is to establish a routing-pheromone distribution in which the required next-hop node has a higher routing-pheromone value. Let us introduce following terms to simplify our explanation of routing.

n_i:	ID of sensor node.
S_k:	ID of sink. At the same time, S_k also represents ID of cluster to which sink S_k is dedicated.
S_{n_i}:	ID of sink that n_i belongs to.
$Pb_{S_k}(n_i)$:	Routing-pheromone value that n_i assigns to backward ant, which is transmitted by S_k.
$P_{n_i}(n_i)$	Routing-pheromone value for n_i to declare as its own pheromone.
$P_{n_i}(n_j, S_k)$:	Routing-pheromone value stored in n_i's pheromone table that represents benefits of n_j as next-hop node to transmit packet to S_k.
$C_{n_i}(S_k)$:	Cluster pheromone of S_k estimated by n_i.

A sink S_a broadcasts backward ant B with maximum routing-pheromone value $Pb_{S_a}(S_a) = P_{\max}$. On receiving B, sensor node n_i stores routing pheromone carried by the backward ant ($Pb_{S_a}(S_a)$), its source node (S_a), and sensor node which relays B immediately before (S_a) as an entry, in its own pheromone table. Thus, n_i memorizes that the benefit of selecting S_a as a next-hop node for transmitting packets to S_a is $P_{\max}$. After that, n_i relays B, making B carry a new routing-pheromone value. This new routing-pheromone value $Pb_{n_i}(S_a)$ is calculated according to:

$$Pb_{n_i}(S_a) = \alpha\left(1 - \exp\left(-\beta\frac{E_{R_i}}{E_{I_i}}\right)\right) Pb_{S_a}(S_a) \qquad 0 < \alpha < 1, \beta > 0 \quad (2)$$

After receiving B, which is relayed by n_i, n_j creates a new entry $(n_i, S_a, Pb_{n_i}(S_a))$ as in the case of n_i. Then, n_j calculates a new routing-pheromone value according to Eq. (2), and forwards B with a new routing-pheromone again. A good pheromone distribution emerges through frequent repetitions of these behaviors.

Sensor nodes periodically communicate using a hello message like that in the centralized control described in Sect. III-A. But the purpose of this hello message is not only to provide a countermeasure to node failures but also to comprehend the situation of surrounding area. The hello message transmitted from n_i conveys routing-pheromone value of n_i itself (p_{n_i}), cluster ID to which n_i belongs to (S_{n_i}), and cluster pheromone of S_{n_i} evaluated by n_i ($C_{n_i}(S_{n_i})$), which is described in detail later in this section. p_{n_i} is the mean routing-pheromone value for all entries in n_i's routing table. After receiving the hello message, n_j updates the routing-pheromone value for the n_i's entry in n_j's routing table following Eq. (3) with $\gamma \in [0, 1]$.

$$P_{n_j}(n_i, S_{n_j}) = \gamma P_{n_j}(n_i, S_{n_i}) + (1 - \gamma) p_{n_i} \quad (3)$$

A sensor node chooses its next-hop node stochastically using the routing-pheromone distribution, and relays packets to it. Assuming N_{n_i} is a set of neighboring nodes for n_i, which is equivalent to candidate set of next-hop nodes, the probability of n_i selecting n_j as its next-hop node is represented as:

$$p_{n_i}(n_j) = \frac{P_{n_i}(n_j, S_{n_i})^2}{\sum_{k \in N_{n_i}} P_{n_i}(k, S_{n_i})^2} \quad (4)$$

This form of equation is used in some propositions using the ACO approach, e.g., [5]. Routing loop can be constructed due to the probabilistic approach, but discarding the looped packets made the data collection unreliable in our simulations. So now we avoid routing loops by appending node IDs the packet went through to the header. Sensor nodes listed in the header are excluded from the set of candidate for next-hop node. This requires only a small amount of communications overhead.

How to select a destination sink still remains a question in multi-sink sensor networks. Our clustering method, ant-based clustering, is also inspired by a swarm behavior of ants. Ant-based clustering was originally a method of swarm intelligence by ants grouping eggs or larvae according to their size. Ants repeatedly pick up and drop larvae based on their degree of similarity with neighbor eggs while wandering around. In such a behavior, larvae which differ substantially from their neighbors in size move toward similar-sized ones, and clusters of different-sized larvae emerge in a self-organized way. We substitute similarity with the advantage of belonging to a cluster, and do clustering to suit the network situation.

Each node calculates a cluster-pheromone value based on the routing-pheromone values, and uses them to determine which cluster it should belong to. Cluster S_{n_i}'s cluster pheromone evaluated by n_i is defined as:

$$C_{n_i}(S_{n_i}) = \frac{\sum_{k \in \mathrm{blng}_{n_i}(S_{n_i})} C_k(S_{n_i}) + avg\_ph_{n_i}(S_{n_i})}{|blng_{n_i}(S_{n_i})| + 1} \quad (5)$$

where $\text{blng}_{n_i}(S_{n_i})$ represents a set of neighboring nodes of n_i that participate in cluster S_{n_i}. This information can be recognized via hello messages, which has the cluster ID of the sender. The term $\text{avg\_ph}_{n_i}(S_{n_i})$ is the mean of routing-pheromone values in entries having destination sink S_{n_i}:

$$avg\_ph_{n_i}(S_{n_i}) = \frac{\sum_{k \in \text{blng}_{n_i}(S_{n_i})} P_{n_i}(k, S_{n_i})}{|\text{blng}_{n_i}(S_{n_i})|} \quad (6)$$

Cluster-pheromone value is conveyed in hello packets, so each sensor node can acquire the cluster-pheromone values of neighboring clusters. Sensor nodes regard a cluster with a higher cluster-pheromone value as a good cluster to join, and stochastically switch to it. The probability of n_i changing its cluster from S_j to S_k is

$$P_{n_i}(S_j \to S_k) = \left(\frac{f_{n_i}(S_j, S_k)}{k_{th} + f_{n_i}(S_j, S_k)} \right)^2 \quad (7)$$

where k_{th} is a constant value used to control the probability and where $f_{n_i}(S_j, S_k)$ is calculated as follows:

$$f_{n_i}(S_j, S_k) = \max\left(0, \frac{|\text{blng}_{n_i}(S_k)|}{N_{n_i}} \frac{C_{n_i}(S_k) - C_{n_i}(S_j)}{C_{n_i}(S_k)}\right) \quad (8)$$

The detection of node failures is exactly equivalent to that of centralized control described in Sect. III-A. After t_{expire} passes without receiving hello packets from sensor node n_j, neighboring node n_i detects that n_j has failed. By deleting the entry for n_j in its pheromone table, n_i selects appropriate next-hop nodes according to Eq. (4) without any special handling.

Detecting sink failure was also based on the same soft-state model. That is, the sink periodically broadcast hello message as well as sensor nodes. Sensor nodes around the sink determine that the sink has failed if they had not received hello message from the sink for $3 \times t_{\text{expire}}$. The cluster in sink failure is no longer preferable. Thus, sensor nodes set cluster-pheromone values of all the entries stored in their neighbors table to 0 and abandon their membership. As hello messages indicating the sink failure propagated over the network after that, sensor nodes participating in the failed sink's cluster also abandoned their membership, and joined other clusters.

IV. Evaluation and discussion

We try comparison of our self-organized and centralized controls by simulation experiments. First, we explain the simulation model which experimented, then we evaluate the robustness against various perturbations.

A. Simulation Environment

We implemented our self-organized and centralized controls on ns-2 network simulator [1]. In the following experiments, we randomly placed 300 sensor nodes over a region monitoring a square, 100 m per side, unless otherwise stated. We assumed there were four sinks at $(25, 25)$, $(75, 25)$, $(25, 75)$, $(75, 75)$, respectively. We tested other sink positions, and obtained almost the same results.

TABLE I
Sensor node parameters

Transmission power	0 dB
Communication range	10 m
Frequency	2,450 MHz
Bit rate	250 kbps
Height of antenna	20 cm
Initial power	25 J
Power consumption in transmission state	40.95 mW
Power consumption in receiving state	45.78 mW

TABLE II
Simulation parameters

t_{hello}	1 s
t_{expire}	5 s
P_{max}	10
α	0.7
β	7
γ	0.875
k_{th}	0.5
Size of a hello packet	10 bytes
Size of a failure detection packet	10 bytes
Size of a failure recovery packet	10 bytes
Size of a data packet	64 bytes

We used the two-ray ground reflection model [1] as the radio propagation model, and the MAC and PHY layers follow the IEEE 802.15.4 specification. In the simulation of the centralized control, the size of command packet can easily exceeded the value specified in IEEE 802.15.4. We therefore virtually set *aMaxPHYPacketSize*, which determines the maximum length of a packet, to infinity. The size of the command packet transmitted from sink S_j is calculated using the following equation:

$$\sum_i 6 \cdot e_{n_i} \cdot num_{S_j} + 7 \quad (9)$$

where e_{n_i} is the number of previous- and next-hop node pairs assigned to node n_i and num_{S_j} is the number of sensor nodes in cluster S_j. We assume that 6 bytes are enough for the pair, and that 7 bytes are enough for a header. We set the parameters of sensor nodes (listed in Table. I) by referring to [18]. The simulation parameters are also listed in Table. II. We do not consider FEC to take particular note of the effect of transmission error, therefore the packet is discarded even if one bit error occurs.

In the following data-collection model we used, sensor nodes send the information they obtain to their sinks in a multi-hop way at a predefined interval $t_{\text{intval}} = 10$ s. Sensor nodes do not synchronized with each other, and the transmission time of it is independent of that of the others. One of the most important metrics for sensor networks is the reliability of which information is brought to a sink. We therefore defined a metric we call the data-collection rate. When the number of sensor nodes that work properly is N_{act}, the number of data packets generated in t_{intval} is of course N_{act}. When the number of packets that reach one of the sinks is r, the data-collection rate is defined as r/N_{act}.

In the centralized control, the parameter with the greatest influence on the data-collection rate is command-packet trans-

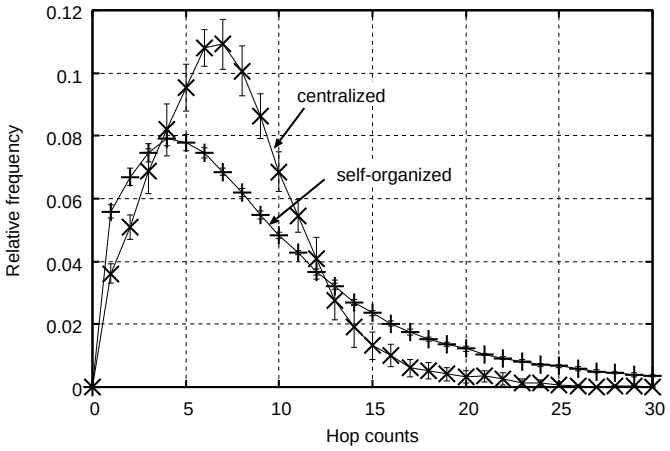


Fig. 2. Efficiency of routes generated by centralized control and of routes generated by self-organized control.

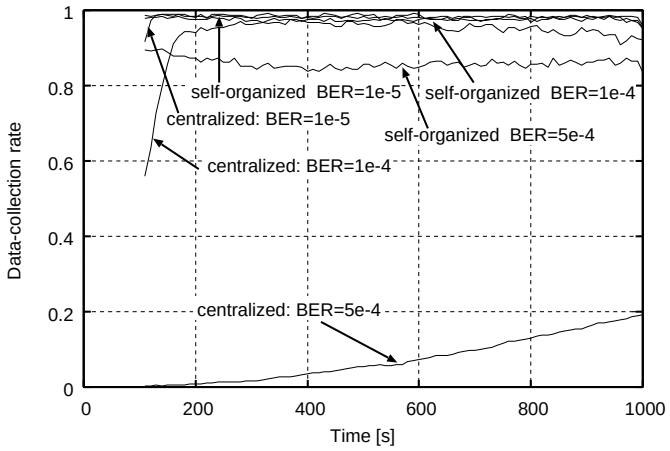


Fig. 3. Influence of BER on data-collection rate.

mission interval. If this interval is too long, sensor nodes will only slowly find out what it should do next, especially when the command packets are frequently lost. And if the interval is too short, command packets coming one after another result in severe interference problems. We conducted simulation experiments to find out whether 1 s, 10 s, 100 s, or 500 s would be the best interval and chose 10 s as the one yielding the best balance between data-collection rate and power consumption. Not only in centralized control, transmission interval of backward ants in our self-organized control also has great influence. Too short an interval causes repeated interference and too long an interval does not construct pheromone distribution enough for data gathering. We simulated transmission intervals of 10 s, 100 s, and 500 s and selected 100 s.

In the simulation experiments, each sink transmits command packets or backward ants until at least 100 s pass over. So we consider the network is in the transient state for 100 s from the start, and do not plot a graph in the time window.

B. Instability of Generated Routes

We first compared the efficiency, in terms of hop counts, of the routes generated using centralized control and self-organized control. The hop counts reported here are mean values of all routes between each sensor node and its sink. The distribution of hop count is shown in Figure 2 with 95% confidence intervals. This graph is for the idealized scenario in which no node failures occur, and bit error rate is set to 10^{-5}. Changing BER did not generate significant influence. Actually, there is only a little difference in their distribution as shown in Figure 2, and the same is true for their mean values as shown in Table. III, where statistics values of the routes are listed. However, variance of both control approaches differs substantially. These interesting results suggest that quality of generated routes can fluctuate widely, i.e., low predictability and controllability, in self-organized control. A sensor node in self-organized control decides its own action on the basis of limited, local information. Their lack of global viewpoint leads to difficulty in finding global optimum, and results in wide fluctuation.

C. Measures against Transmission Error

We conducted simulation experiments to study the robustness of both control approaches against transmission error under the assumption that no node failures occur. In Figure 3, both kinds of control show about the same data-collection rate with BER $= 10^{-5}$, but that of centralized control becomes slow to rise up along with the increase in BER.

In the centralized control, the tremendous amount of information is gathered to the central station to decide a course of actions for each sensor node, and the station issues the instructions to sensor nodes. Sensor nodes completely rely on the control information from the station, and the station believes sensor nodes follow the order. With this strong dependency, what will happen when the information is beyond some sensors' reach? This situation just arises due to transmission error in this simulation experiment. In the case where some sensors can receive the instruction and others cannot, inconsistent views of the routes can be introduced among them. Such inconsistency makes sensor nodes lose their next-hop node for a received packet, and the network gets stuck in the pathological state until their views get consistent. Actually, data-collection rate increases with time in Figure 3, but this is because frequently transmitted command packets (i.e., with an interval of only 10 s) compensate discarded ones. This slow ascent means that the network does not adapt well when the route changes for any reason.

In the self-organized control, sensor nodes are not able to know global information of the network, leading to easily have inconsistent information among them. But the adverse effects of their inconsistency are localized around them, because they have its own knowledge base based on their limited view, instead of sharing global information. That results in the good robustness against transmission error as shown in Figure 3.

Differences in the behaviors of the two kinds of control also appear in Figure 4, where mean of the data-collection rate

TABLE III
STATISTICS OF ROUTES GENERATED IN CENTRALIZED CONTROL AND SELF-ORGANIZED CONTROL. 95% CONFIDENCE INTERVALS ARE ALSO SHOWN.

	Centralized control	Self-organized control
Average hop count	7.47 ± 0.36	9.08 ± 0.34
Average delay	$0.156 \pm 8.62 \times 10^{-3}$	$0.226 \pm 1.56 \times 10^{-2}$

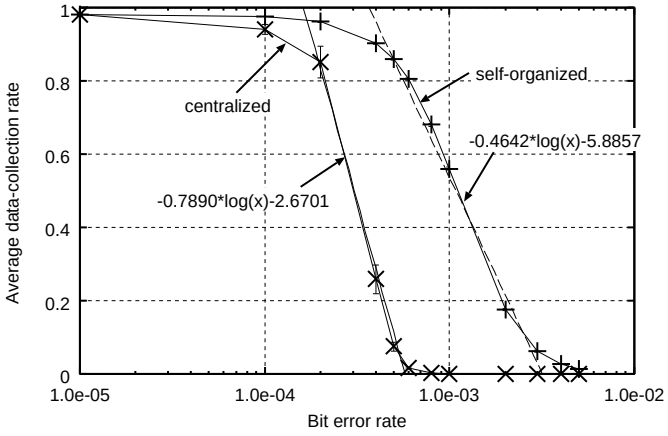


Fig. 4. Data-collection rate versus BER.

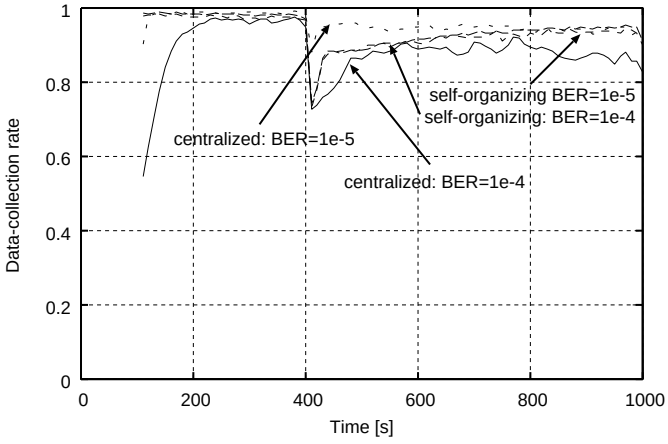


Fig. 5. Features of the process of recovery from sink failure.

are plotted against BER. Logarithmic approximation lines for their decays are also shown. Self-organized control keeps data-collection rate above 80% about 3 times longer. In addition, the gradient of the self-organized control is only 58% of the centralized control. When the gradient is steep, the network function might deteriorate markedly in response to even a small change of BER. When the gradient is gentle, however, data collection is not affected significantly if the BER changes. For that reason, self-organized control is more robust against transmission error. As mentioned above, such robustness of the self-organized control originates the fact that each node operates based on only its local information. On the other hand, the lack of the correspondence of routing information by the packet loss causes the performance deterioration at the centralized control.

D. Measures against sink failure

Figure 5 presents the results for the case in which a sink located at $(25, 25)$ fails at 400 s. After the sink failure, the data collection rate drops sharply to about 75%, except in the case of centralized control with 10^{-5} BER (Bit Error Rate), where the rate drops to only 90%. A rate of 75% means that one cluster suffered catastrophic damage (the ratio of data packets gathered within a cluster is about 25%). Not only is the drop in the data collection rate in the case of centralized control and low BER small, but also the recovery is almost immediate. The control station which is wired to the sinks becomes aware of the failure within a short amount of time (in our simulations, it is set to 0 s), after which the clusters are reconstructed and the routes are recomputed upon receiving the command packet, in order to adapt the whole network to the failure. Sensor nodes immediately modify their cluster membership and routing table according to the instructions contained in the command packet, and the data collection rate is restored soon after that. Indeed, in cases where the channel quality is poor, the data collection rate in the centralized control scheme is unable to recover within the simulation time shown in Figure 5, since centralized control is weak with respect to transmission errors, as indicated in [14].

In contrast to the centralized control scheme, the self-organized control scheme needs more time for the distant sensor nodes to adapt to the sink failure. In addition, since the network has no supervisor and no explicit instructions, some nodes might be prone to taking contradicting actions based on the possibility of receiving inaccurate information about the condition of the network. For these reasons, in low BER environments, the self-organized control scheme exhibits worse recovery than the centralized one. In high BER environments, however, the relationship between self-organized control and centralized control is reversed, since the self-organized control scheme inherently does not have critically important information whose loss can bring serious and adverse influence to the network.

E. Measures against node failure

We already demonstrated the robustness against node failure in our previous work [14]. Moreover, we showed that although most of the sensor nodes other than the failed ones exhibit data collection rates of about 100% in the self-organized control scheme, failures in the case of the centralized control scheme have considerable influence on the data collection rates at the cluster level, where many sensor nodes are unable to transmit packets to their sinks, and this influence is especially notable when concentrated and simultaneous failures occur. However, when we tested random failures in a 100 m $\times$100 m

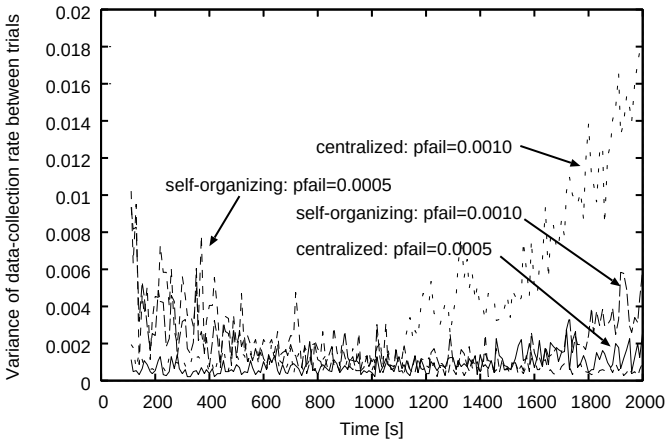
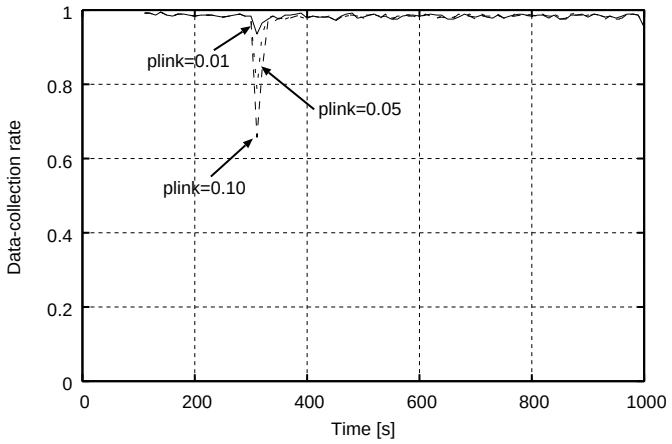
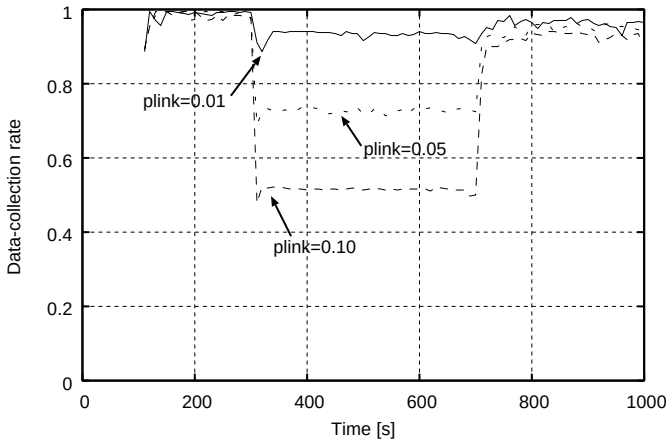


Fig. 6. Variances of the data collection rates among trials.



(a) Self-organized control.



(b) Centralized control.

Fig. 7. Influence of link disconnections on the data collection rate.

monitoring region containing 300 nodes, the difference in the robustness of the self-organized and the centralized control schemes was not clear due to the connectivity degradation caused by the continual node failures. Therefore, here we temporarily used a narrower monitoring region of 50 m $\times$ 50 m while keeping the number of nodes and sinks, and defined p_{fail} as the failure rate per second for each sensor node.

The variances of the data collection rates of both control schemes among trials are shown in Figure 6. The variance in the self-organized control scheme is small and not as sensitive to the failure rates. However, in the centralized control scheme, the data collection rates in some trials experience sudden drops, which lead to the higher variance of the data collection rates, as shown in Figure 6. The high variance in the case of centralized control indicates the difficulty of predicting the data gathering capability in harsh environments, although all of the plots are prepared using the same parameters.

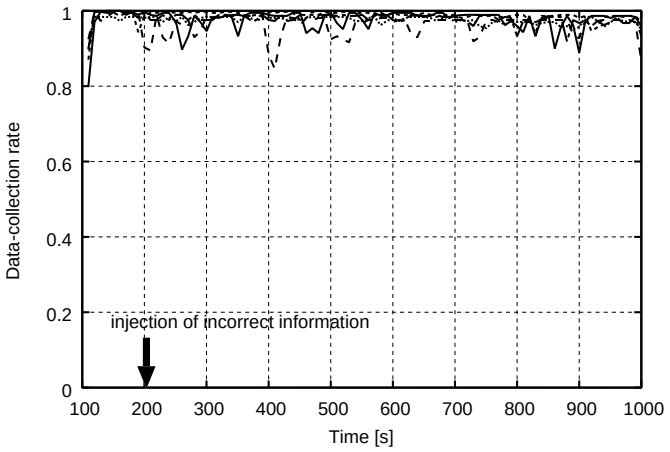
F. Measures against link disconnection

As links can become disconnected intermittently in wireless networks, in the case where the link between nodes n_i and n_j is disconnected but the link between n_i and n_k is still connected, there is a possibility that the status of n_i as seen from the perspective of n_j and n_k is inconsistent. Therefore, in order to study the differences in the robustness of the two schemes, we randomly disconnected a percentage of the links. We assume that each node is linked to an arbitrary neighboring node, and each link is disconnected with probability p_{link} in both directions. This disconnection process was conducted for all nodes, and the duration of the disconnection was 400 s, from t=300 s to t=700 s.

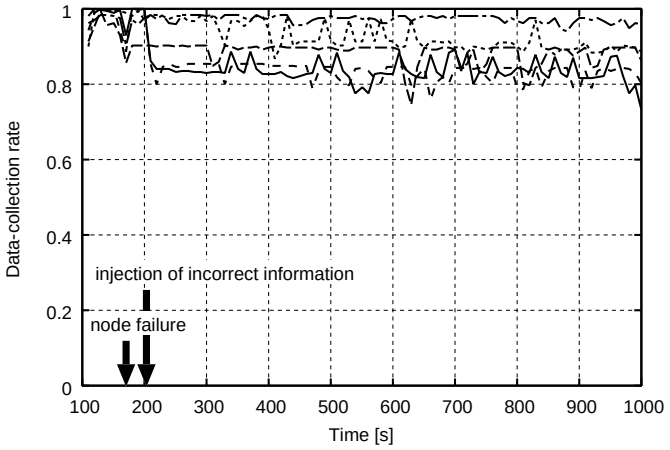
In the results shown in Figure 7, the data collection rate in the self-organized control scheme immediately recovers to the rate before the disconnection, although it experiences a declination for a short amount of time. The centralized control scheme, on the other hand, suffers greatly from the disconnections, where detection of massive node failures occurs since neighboring nodes regard disconnected nodes as failed due to their inability to transmit hello messages. In other words, sensor nodes cannot distinguish failures from link disconnections in our centralized control scheme. Furthermore, after the detection of a missing link, the neighboring nodes transmit failure-indication packets, which are in fact false-positive detection packets, to the control station. As a result, the control station does not provide routes to the node which is considered as failed, and the packets from the disconnected node are discarded, which is the main reason for the decay of the data detection rate in Figure 7(b).

V. Dependence on control information

Next, we consider the factor which affects the difference in robustness and perform the evaluation by additional simulations.

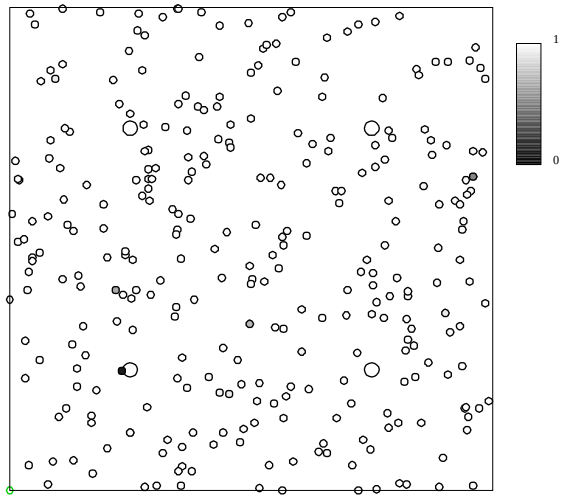


(a) False-positive failure detection.

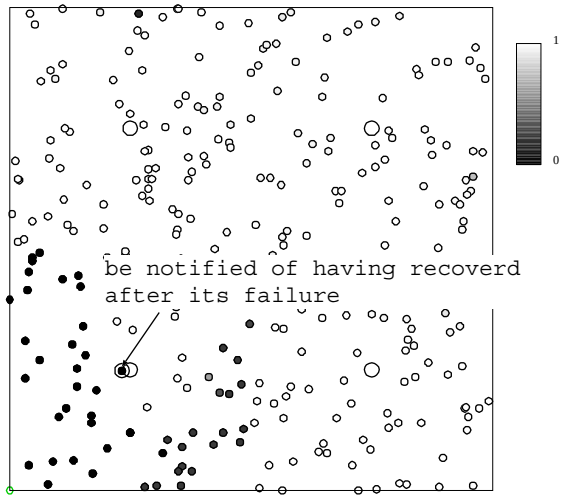


(b) False-recovery indication.

Fig. 8. Results of injecting incorrect information.



(a) From t=160 s to t=200 s



(b) From t=200 s to t=1000 s

Fig. 9. State of the network after injecting false-recovery information.

A. Factors influencing the difference in robustness

In the evaluation presented in Section IV and in previous works, there was a significant difference between the robustness of self-organized control and centralized control. We are inclined to explain this trend in terms of "dependence on control information". In this case, "dependence" has almost the same meaning as that used in fault management. The dependence is a relation in which an error or failure in an object may cause an error or failure in another object. We define control information as the information exchanged between entities of a given network which coordinates their joint operation.

In Sections IV-E and IV-F, even the control station itself did not comprehend the correct state of the network. This is caused by the fact that the control station also depends on control information received from the nodes in the network. The control station constructs a precise view of the whole network by integrating each piece of information about the state of the network. In other words, the problem of the dependence is that the control information from potentially unreliable nodes in environments where reliable communication is not guaranteed plays a critical role in generating the control scheme at the control station. In Section IV-E, failure indication packets, which notify the command node about the correct state of the network, did not reach the control station, resulting in a sudden drop of the data collection capability of the clusters. In Section IV-F, one node considers a neighboring node to be operating correctly, while another node considers the same neighboring node as faulty, resulting in the transmission of failure indication packets even though no nodes have failed.

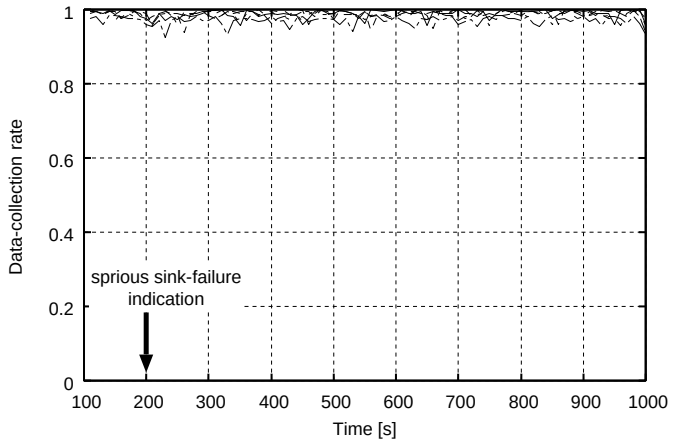


Fig. 10. Influence of erroneous sink failure indication.

In this way, information which does not reflect the correct state of the network brings vulnerability to the centralized control scheme. However, since optimization of the whole network cannot be performed if the dependency on the control information is reduced, general network performance like delay or a throughput degrades. For example, as shown in Figure 2, in self-organization control, distribution of the number of hop from a node to the sink is large, and there are nodes to whom delay becomes large very much.

Of course, at the node level, self-organized control is identical to centralized control, meaning that individual nodes potentially have an erroneous understanding about the state of the network. However, individual nodes affect only their surrounding environment or neighboring nodes since all nodes have only partial view of the network, and do not transmit or receive explicit control information. Due to this behavior, the influence of individual nodes on the global state of the network is much smaller than in the centralized control scheme. In this regard, since we have not yet clarified the influence of erroneous information received from individual nodes, in the next section we verify our idea by deliberately injecting incorrect information into the network.

B. Influence of incorrect information

The purpose of this demonstration is to determine how strong the influence of information received from individual nodes is, as well as how potentially unreliable nodes affect the behavior of the whole network. Therefore, in this section, we deliberately inject spurious information in order to show unambiguously the influence of information received from individual nodes on the functionality of the network. At first, in the centralized control scheme, we considered two scenarios: 1) we injected false-positive failure detection packets, which convey the misinformation that a properly working node is detected as failed, and 2) false-recovery packets, which inform the surrounding nodes that a node which has failed is detected as recovered.

Although we deliberately injected incorrect information at t=200 s that the node nearest to the coordinate $(25, 25)$ had failed, there was no fluctuation or drop in the data collection rate due to the injection, as seen from the results shown in Figure 8(a). In fact, the node which was wrongly detected as failed was not able to send its packets to the sink as the control station did not consider the failed node as a member of the data collection cluster. However, routing information was supplied to the other sensor nodes correctly, and thus the influence of the erroneous information was limited.

Next, we tested the scenario where incorrect information about the recovery of a node is injected into the network. At first, we made the node nearest to the coordinate $(25, 25)$ fail at t=160 s, followed by the injection of information that the node has recovered at t=200 s. Figure 8(b) shows the results of five trials, and it is clear that the behavior of the data collection rates are different among them, i.e., they are different depending on the node deployment. There is a clear drop in two of the plotted lines just after the injection of erroneous information at t=200 s. Given this factor, focusing on one of those lines, in Figure 9 we visualized the data collection rate of the individual nodes from the time when node fails (t=160 s) until the injection of misinformation (t=200 s), and from the injection (t=200 s) to the end of the simulation (t=1000 s), respectively. As shown in Figure 9(a), the influence of the node failure can be limited. However, after the injection, data collection in the larger part of the respective cluster becomes impossible.

Self-organized control does not have any means for explicit indication of failure or failure recovery. Therefore, it was impossible to compare it directly with the centralized control in terms of the influence of erroneous information. Instead, we used the indication of sink failure, which is a message which explicitly conveys information about the failure of a sink to the neighboring nodes by using a hello message. Furthermore, we made the sensor node nearest to the coordinate $(25, 25)$ transmit the information about the sink failure. This indication is spread over the respective cluster through forwarding by nodes which receive the indication.

As a result, although spurious sink failure indication was injected into the network at $t = 200$ s, there was no clear difference in the data collection rate before and after the injection, as seen from the data collection rates from five trials presented in Figure 10. In our self-organized control scheme, sensor nodes invalidate their membership to the respective cluster upon receiving the sink failure indication, and negative influence was expected due to the dynamic change of cluster membership. However, contrary to our expectation, the cluster memberships were restored to those before the injection. In other words, correct information from other nodes naturally adjusts the situation caused by erroneous information, and this fact contributes to the robustness of self-organized control.

VI. Conclusion

In spite of growing interest, there are many points regarding self-organization which remain insufficiently understood. In this paper, we studied the robustness of self-organized

control against a wide range of perturbations by comparing it with centralized control, and we attempted to answer some important questions. One such question is whether self-organized control is in fact robust, and we quantitatively demonstrated the affirmative answer by examining various scenarios. Although this result is not surprising, it was found that self-organized control has the obvious benefit of superior robustness, especially if applied to systems in dynamically changing environments, although at the cost of reduced system predictability. Furthermore, the questions about why self-organized control is robust and what factors determine the robustness of self-organized control were also addressed, and based on the results obtained from the simulation experiments, we arrived at the conclusion that the dependence on the control information in the system plays a critical role in determining whether or not the robustness is sufficient. In a network which is composed of potentially unreliable nodes and is located in a harsh environment, decreasing the dependence on the control information received from the nodes is critical to yielding sufficient robustness, and self-organized control inherently possesses such properties.

Acknowledgements

This research was partially supported by the "Global COE (Center of Excellence) Program" and the Grant-in Aid for Scientific Research (C) 19500060 of the Ministry of Education, Culture, Sports, Science and Technology, Japan.

References

[1] ns-2 – the network simulator. online available at http://www.isi.edu/nsnam/ns.

[2] B. Barń and R. Sosa. AntNet: Routing algorithm for data networks based on mobile agents. *Inteligencia Artificial, Revista Iberoamericana de Inteligencia Artificial*, 12:75–84, 2001.

[3] E. Bonabeau, G. Theraulaz, and M. Dorigo. *Swarm Intelligence: From Natural to Artificial Systems (Santa Fe Institute Studies in the Sciences of Complexity Proceedings)*. Oxford Univ Pr, Oct. 1999.

[4] P. Boonma, P. Champrasert, and J. Suzuki. BiSNET: A biologically-inspired architecture for wireless sensor networks. In *Proceedings of The Second IARIA International Conference on Autonomic and Autonomous Systems*, Published by the IEEE Computer Press, July 2006.

[5] G. D. Caro, F. Ducatelle, and L. M. Gambardella. AntHocNet: An ant-based hybrid routing algorithm for mobile ad hoc networks. *European Transactions on Telecommunications*, 16:443–455, Oct. 2005.

[6] H. Chan and A. Perrig. ACE: An emergent algorithm for highly uniform cluster formation. In *Proceedings of the First European Workshop on Wireless Sensor Networks*, pages 154–171, Jan. 2004.

[7] M. Dorigo, V. Maniezzo, and A. Colorni. The ant system: Optimization by a colony of cooperating agents. *IEEE Trans. Systems, Man, and Cybernetics*, 26(2):29–41, 1996.

[8] F. Dressler. Self-organization in ad hoc networks: Overview and classification. Technical report, University of Erlangen, Department of Computer Science 7, Mar. 2006.

[9] L. Gan, J. Liu, and X. Jin. Agent-based, energy efficient routing in sensor networks. In *Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems*, pages 472–479, Aug. 2004.

[10] C. Gershenson and F. Heylighen. When can we call a system self-organizing? In *Proc. 7th European Conference on Advances in Artificial Life*, pages 604–614, Sept. 2003.

[11] J. Handl, J. Knowles, and M. Dorigo. Strategies for the increased robustness of ant-based clustering. *Engineering Self-Organising Systems: Nature-Inspired Approaches to Software Engineering*, 2977:90–104, 2004.

[12] C. Intanagonwiwat, R. Govindan, and D. Estrin. Directed diffusion: A scalable and robust communication paradigm for sensor networks. In *Proceedings of the Sixth Annual International Conference on Mobile Computing and Networks*, pages 56–67, Aug. 2000.

[13] Y. Kiri, M. Sugano, and M. Murata. Self-organized data-gathering scheme for multi-sink sensor networks inspired by swarm intelligence. In *Proc. 1st IEEE Intl. Conf. on Self-Adaptive and Self-Organizing Systems*, July 2007.

[14] Y. Kiri, M. Sugano, and M. Murata. Robustness differences between bio-inspired control and centralized control. In *Proc. of Biological Approaches for Engineering Conference*, Mar. 2008.

[15] H. Kitano. Biological robustness. *Nature Review Genetics*, 5(11):826–837, Nov. 2004.

[16] Z. Liu, M. Z. Kwiatkowska, and C. Constantinou. A biologically inspired optimization to AODV routing protocol. In *Proceedings of the 3rd Workshop on the Internet, Telecommunications and Signal Processing*, pages 106–111, Dec. 2004.

[17] 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 19th National Conference on Artificial Intelligence*, pages 28–33, July 2004.

[18] Moteiv Corporation. *Telos (Rev B): PRELIMINARY Datasheet*, May 2004.

[19] C. Prehofer and C. Bettstetter. Self-organization in communication networks: Principles and design paradigms. *IEEE Communications Magazine, Feature Topic on Advances in Self-Organizing Networks*, 43(7):78–85, July 2005.

[20] T. D. Seeley. When is self-organization used in biological systems? *Biological Bulletin*, 202:314–318, June 2002.

[21] 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 dependence. In *Proceedings of The Third IARIA International Conference on Systems and Networks Communications (ICSNC 2008)*, Oct. 2008.

[22] A. L. Vizine, L. N. de Castro, E. R. Hruschka, and R. R. Gudwin. Towards improving clustering ants: An adaptive ant clustering algorithm. *Informatica Journal*, 29(2):143–154, July 2005.

[23] M. Younis, M. Youssef, and K. Arisha. Energy-aware routing in cluster-based sensor networks. In *Proc. 10th IEEE International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunications Systems*, Oct. 2002.

[24] D. Zaharie and F. Zamfirache. Dealing with noise in ant-based clustering. *IEEE Trans. Evolutionary Computation*, pages 2395–2402, Sept. 2005.

[25] Y. Zhang, L. D. Kuhn, and M. P. Fromherz. Improvements on ant routing for sensor networks. In *Proceedings of the Fourth International Workshop on Ant Colony Optimization and Swarm Intelligence*, pages 154–165, Sept. 2004.