Master's Thesis

Title

Robustness of Self-organizing Control in Sensor Networks

Supervisor

Professor Masayuki Murata

Author

Yuichi Kiri

February 13th, 2008

Department of Information Networking

Graduate School of Information Science and Technology

Osaka University

Master's Thesis

Robustness of Self-organizing Control in Sensor Networks

Yuichi Kiri

Abstract

The idea of advantage of distributed control over centralized control has been radicated. Now, a class of distributed control, self-organizing control, has received significant attention in the areas of networking. One of the main factors of their attention is its robustness, which is a property of maintaining function of the system despite perturbations. Assuming modern and future networks, which are becoming increasingly complex with growth in the number of nodes participating in the communications, frequent topological changes due to additions or departure of nodes, and unanticipated perturbations inside and outside of the network, the notion of robustness becomes also increasingly important. However, we stress that whether self-organizing control is robust or not is nontrivial. Even if it is actually robust, why self-organizing control is robust and what factors yield robustness has not been addressed to any great extent so far.

In this thesis, we first quantitatively demonstrate the advantage in robustness of self-organizing control through comparison with centralized control in a sensor network scenario as a special topic for the study. Through the simulation experiments, we show that although centralized control scheme shows superior predictability in routing metrics such as delay in ideal environments, self-organizing control scheme maintains its data collection functionality even in a variety of perturbations, such as transmission error, node failures, link disconnections, etc. In addition by the results of the evaluation, we point out that the difference of robustness between the two control schemes stems from the strength of dependence on other nodes' information to comprehend state of the network. A control station in centralized control depends on information about network state gathered from all the nodes in the network, which are often unreliable. This fact brings vulnerability to the centralized control especially when the control reflects the erroneous state of the network. On the contrary, network nodes in self-organizing control have not so strong dependency on other nodes' information, and thus, the influence of erroneous information is just localized.

Keywords

Sensor Network

Self-organizing Control

Centralized Control

Robustness

Fault Management

Contents

List of Figures

List of Tables

1 Introduction

Today's networks are becoming larger and more complex. A large number of diverse devices are being connected to them, and issues such as node failures and the addition or departure of nodes are continually changing the complexity and dynamics of peer-to-peer, ad-hoc, and sensor networks. Critical issue in such dynamically changing and uncertain environments is to maintain the functionality of networks in an adaptive manner to the environmental changes. Even if basic performance is inferior to other protocols or control schemes, keeping the performance even where the network state dramatically changes or unforeseeable circumstances occur is preferable for today's and future networks. In this age when our life and networks are closely related to each other, robustness of networks is becoming increasingly important.

Regarding to the robustness, distributed control has been said to be preferable to centralized control. In centralized control, a control station gathers each piece of network state from individual nodes in a network, integrates all the pieces to construct a full picture of the network, and issues explicit command generated based on the picture to all the nodes. In such cases, as the number of nodes increases, collecting, integrating, and distributing control information become increasingly difficult. In addition, network nodes under control of the station are unable to determine their actions without receiving control information from the station. Furthermore, the control station is inherently a single point of failure. In that sense, centralized control lacks scalability and fault tolerance. These weaknesses finally demand other control paradigm, and as a result, distributed control appears. Distributed control has been expected to solve these problems.

Now, a class of distributed control that is beginning to attract considerable attention is self-organizing control [1, 2]. Self-organizing control is a paradigm introduced from Nature [3]. In this control, 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 [4, 5]. Such behavior is distinct from merely distributed control where individual components act autonomously but depend on global information. Scalability, adaptability, and fault tolerance, which are included in robustness in a broad sense, are "known" as properties of self-organizing control, and a lot of researches incorporating the self-organizing control have been carried out in routing [6-10], clustering [11-13], task allocation [14], etc.

In spite of such significant attention, most of the researches have focused on improving network performance or optimizing the network efficiency. The notion of good robustness of self-organizing control is widely alleged, but we stress that it is certainly nontrivial. Even supposing that the notion is true, why self-organizing control is robust and what factors yield advantage of robustness compared to other control, to the best of our knowledge, have not been discussed to any great extent.

The purpose of this thesis is to quantitatively demonstrate the advantage of self-organizing control in robustness. By the results of evaluation, we also tackle interesting questions, why and how self-organizing control is robust.

In order to answer them, we have to define what "robustness" is. Robustness tends to be used in a vague way, however, its definition seems to gain basic consensus over broad fields. For example, in [15], robustness is defined as "dependability with respect to external faults, which characterizes a system reaction to a specific class of faults". This article also defines dependability as an "ability to deliver service that can justifiably be trusted", and the service is "system's behavior as it is perceived by its user." The definition of robustness in [16] is "a delivery of a correct service in implicitly-defined adverse situations arising due to an uncertain system environment,' and in [17, 18], "property that allows a system to maintain its functions despite external and internal perturbations." The definition in common is that robustness is a property to maintain its function, or service, to provide its user. The reason for a vague usage regardless of these common understandings on what robustness means assumed to be due to its enormous degree of dimensionality. So, we have to define what system is, what function of the system is, and what perturbations the system faces are to discuss robustness.

Our targeted "system" in this thesis is sensor network. Sensor network is a wireless network, which is composed of sensor nodes equipped with miniature sensing devices and wireless communication capability. To achieve its "function", i.e., data collection, individual nodes sense their ambient surroundings or events, and the sensor data are gathered in a multi-hop way to a base station, which is called sink. In sensor networks, a large number of nodes are assumed to be deployed, furthermore, a lot of other "perturbations" also must be taken into account such as poor quality of wireless links, addition or failures of nodes, and even incorrect control information. Thus, keeping its function regardless of these perturbations is an extremely important building block in the design of sensor network, and sensor network is appropriate to study "robustness".

In this thesis, we first show the better robustness of self-organizing control by comparing two typical protocols which belong to self-organizing control and centralized one, respectively. The system under consideration in this thesis is a sensor network. Its function to be maintained, or its purpose in a sense, is data collection. Thus, robustness of a sensor network in this thesis can be defined as how much its data-collection capability can be kept under perturbations compared against non-perturbed condition. In the followings, we compare and evaluate the robustness based on data-collection rate, whose figure represents data-collection capability of a sensor network. It is defined as the ratio of the total number of packets which have been successfully received by sinks in a predefined period of time, to the total number of transmitted packets in that period. Yet, we have not addressed an important problem: what are perturbations in sensor networks? To evaluate robustness, we need sufficiently broad perturbation space. We select perturbations in a sensor network as many as possible such as transmission error, node failures, and link disconnections, etc. and evaluate robustness under them. The results of the comparisons indicate that in the ideal environment where no perturbation occurs, centralized control exhibits good predictability in data-collection rate and routing metrics such as delay. However, especially when the network state changes, in harsh environment, centralized control cannot maintain its functionality. On the contrary, self-organizing control can keep its data-collection rate even when the state of the network dynamically changes, which indicate self-organizing control has good robustness, and predictability in a different sense from centralized control. Furthermore, from the results, we point out that the difference of robustness is derived from the strength of dependency for comprehending state of the network on other nodes' information, this is the key to differentiate the robustness of both control schemes.

For the second contribution of this thesis, we verified our indication of the dependency on information from unreliable nodes. We demonstrate the influence of the erroneous information with different strength of dependency, and explain why self-organizing control limit the effect of erroneous information locally based on extremely simple information-diffusion model.

This thesis is organized as follows. In Section 2, we explain the detail of self-organizing control scheme in a sensor network, as well as that of centralized control which is used for comparisons. Section 3 provides the simulation results so as to compare the robustness of both control schemes. Based on the obtained results, we discuss what factor yields good robustness for self-organizing control in Section 4, and we conclude this thesis in Section 5.

2 Self-organizing control scheme in sensor networks

In this section, we show the details of self-organizing control which we use for evaluations of robustness. In addition, as another control scheme to be compared with self-organizing control, the details of centralized control are followed. The operations of both control schemes are based on the premise that multiple sinks are deployed in their monitoring regions. Although much research has set its target on sensor networks with single-sink configuration, a not negligible amount of work has addressed multi-sink sensor networks [19-22]. Its advantages are improving scalability by alleviating load of nodes around a sink for forwarding packets and improving robustness by avoiding single point of failure. Especially, we have now most focused on the latter. Using this multi-sink configuration, both control schemes take a cluster-based approach, in which the same number of clusters of nodes as that of sinks is formed, and individual sensor nodes transmit their sensed data to the sink.

2.1 Self-organizing control

Our self-organizing control scheme is based on pheromone-mediated ant-swarm behaviors called ant colony optimization (ACO) [23] and ant clustering [12, 13, 24]. Sensor nodes are divided into as many clusters as there are sinks by using virtual "cluster pheromone" and routing is performed in each cluster by using "routing pheromone." The detailed operation for the self-organizing control is given in the following.

2.1.1 Intra-cluster routing

We have applied the principle of ACO to hop-by-hop routing in our proposed scheme. ACO is originally a probabilistic approach to combinatorial optimization problems like a traveling salesman problem [25], which inspired by ants in their foraging activity. Ants follow efficient routes to their food by being stochastically attracted to higher concentrations of pheromones left by other ants. After finding their food, ants 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 to the nest, reinforcing the trail. If there is enough food that several worker ants can bring food back to the nest, high pheromone concentration will be maintained and even more ants will be attracted. As the food supply becomes smaller, fewer

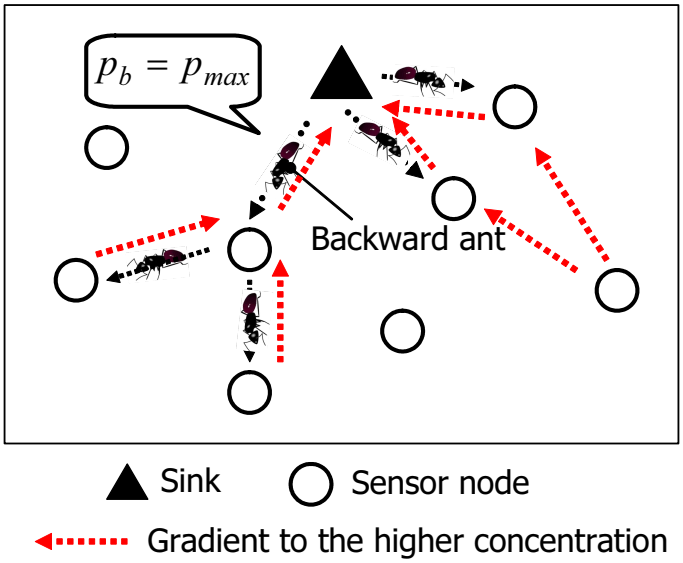


Figure 1: Construction of a gradient of pheromone concentration.

ants will be attracted and the trail will gradually disappear as the pheromone evaporates. Such a trail building is a basic idea behind the ACO approach.

In our self-organizing scheme, a measure of the goodness of going over a neighbor is abstracted to routing pheromone, which is repeatedly updated. An important problem here is the definition of good link to determine which route should have higher-pheromone value, in other words, how to define which node is the preferable next-hop node in a given network. We define a good next-hop node is the node 1) which is nearer to the sink, and 2) which has high residual power. In sensor network where the nodes rely only on their batteries as their power sources, residual power is invaluable and should be considered. All the packets, which are a counterpart of real ants, is stochastically select a next-hop node to arrive at a sink

To produce a gradient of pheromone concentration through which a packet reach a sink by going through efficient route, sinks periodically flood control packets, called backward ants. A sink s_l broadcasts a backward ant b which has maximum pheromone value p_{max} as in Fig 1. On receiving the ant b, node n_i stores b's pheromone value $p_b = p_{max}$, b's source node s_l, and b's previous-hop node s_l to its pheromone table as an entry. Here, node n_i memorizes that the advantage of s_l as its next-hop node is p_{max}. Then node n_i relays b, making b carry a new pheromone

value p_b, which is calculated according to:

$$p_b \Leftarrow \alpha \left(1 - \exp\left(-\beta \frac{E_{R_i}}{E_{I_i}}\right)\right) p_b \tag{1}$$

where $0 < \alpha < 1$ and $\beta > 0$. In Eq. (1), E_{R_i} and E_{I_i} are residual power and initial power of node n_i, respectively. After receiving b, which is forwarded by node n_i, n_i's neighbor node n_j registers a new entry for b in its pheromone table as in the case of node n_i. Node n_j updates b's pheromone value, and forwards b again. By repeatedly executing such a behavior, a pheromone distribution consistent with the above definition emerges.

Sensor nodes periodically communicate using a hello message once in the time t_{hello}. One of its purposes (another purpose is detection of node failures, which are explained in detail in Section 2.1.3) is exchanging pheromones among neighboring nodes. This communication enables sensor nodes to comprehend distribution of pheromone concentration of its surrounding area. The routing-pheromone value h_i conveyed by n_i's hello message is the average routing-pheromone value in its pheromone table. Neighboring node n_j, which receives the hello message, updates the routing-pheromone value for node n_i in its pheromone table according to:

$$p^s_{n_j}(n_i) \Leftarrow \gamma p^s_{n_j}(n_i) + (1 - \gamma) h_i \tag{2}$$

where γ is a constant within $[0, 1]$, and $p^{s_l}_{n_i}(n_j)$ is a pheromone value in n_i's pheromone table, which represents the benefit of using node n_j as a next-hop node bound for sink s_l. Thus, pheromone table reflects up-to-date pheromone distribution.

A sensor node stochastically chooses its next-hop node by using the routing-pheromone values in its pheromone table, and forwards a packet to it. An entries for the neighboring node n_j in n_i's pheromone table indicates the estimated goodness over n_j to reach a sink. Thus the neighbors registered in pheromone table are the candidate set $\mathbf{C_i}$ of next-hop nodes for node n_i.

If $\mathbf{C_i}$ is a candidate set of next-hop node for node n_i, the probability of n_i selecting its neighbor n_j as its next-hop node leading to sink s_l is represented as:

$$\mathit{Prob}^{s_l}_{n_i}(n_j) = \frac{p^{s_l}_{n_i}(n_j)^2}{\sum_{n \in \mathbf{C_i}} p^{s_l}_{n_i}(n)^2}. \tag{3}$$

2.1.2 Clustering

How to select a destination sink for each sensor node still remains a question in our multi-sink sensor network. Our clustering method, ant clustering, is also inspired by a swarm intelligence

of ants. Ant clustering is originally a method by ants grouping eggs of larvae according to their size. Ants repeatedly and also stochastically pick up and drop their eggs based on the degree of similarity in size with neighbor eggs while wandering around. In such a behavior, eggs which differ in size substantially from their neighbors are brought toward similar-sized ones. In such a behavior, clusters of different-sized eggs emerge in a self-organizing way.

We substitute similarity in size with the goodness of belonging to a cluster in order to adapt the ant clustering to sensor networks. Each node calculates a cluster-pheromone value based on the routing-pheromone values stored in their pheromone table, and uses them to determine which cluster it should belong to. Cluster pheromone of s_l evaluated by node n_i is defined as:

$$c_{n_i}(s_l) = \frac{\sum_{k \in \mathit{belong}_{n_i}(s_l)} c_k(s_l) + \mathit{avg\_ph}_{n_i}(s_l)}{|\mathit{belong}_{n_i}(s_l)| + 1} \tag{4}$$

where $\mathit{belong}_{n_i}(s_l)$ represents a set of neighboring nodes of n_i belonging to sink (or cluster) s_l. Cluster-pheromone value $c_{n_i}(s_l)$ and cluster membership of neighboring nodes, which are necessary for calculating Eq. (4), are actually conveyed by hello messages and stored in pheromone table as well as routing pheromone. The term $\mathit{avg\_ph}_{n_i}(s_l)$ represented in Eq. (4) is the average of routing-pheromone values in entries having s_l as a destination sink, calculated as follows.

$$\mathit{avg\_ph}_{n_i}(s_l) = \frac{\sum_{k \in \mathit{belong}_{n_i}(s_l)} p_{n_i}^{s_l}(k)}{|\mathit{belong}_{n_i}(s_l)|} \tag{5}$$

A cluster having higher cluster-pheromone value is regarded as a good cluster to join, and sensor nodes also stochastically switch their cluster membership to it. The probability of node n_i changing its cluster from s_l to s_m is given by:

$$\mathit{Prob}_{n_i}(s_l \rightarrow s_m) = \left(\frac{f_{n_i}(s_l, s_m)}{k_{th} + f_{n_i}(s_l, s_m)} \right)^2 \tag{6}$$

where k_{th} is a constant value used to control the probability, and $f_{n_i}(s_l, s_k)$ is given by:

$$f_{n_i}(s_l, s_m) = \max \left(0, \frac{|\mathit{belong}_{n_i}(s_m)|}{|\mathbf{C_i}|} \frac{c_{n_i}(s_m) - c_{n_i}(s_l)}{c_{n_i}(s_k)} \right) \tag{7}$$

2.1.3 Detection of failures

Sensor nodes are prone to fail due to their cheap production costs. Moreover, power is inevitably depleted during the long periods of operation of a sensor network. Sinks are no exception in that they can fail. Therefore, it is necessary to detect these failures and take appropriate countermeasures in order to be able to gather data over a long term.

We applied a soft-state model to detect failures by using the periodicity of transmitting a hello message. On receiving a hello message from a neighboring node n_i, node n_j interprets the reception as a sign of n_i working properly, and n_j starts a timer for node n_i. Every time node n_j receives a hello message from node n_i, the timer is initiated. If the timer reaches an expiry time t_{expire}, node n_i is deemed to have failed, and node n_j deletes its entry from its pheromone table. By only deleting it, node n_j selects an appropriate next-hop node according to Eq. (3) without any special handling.

Detecting a sink node failure is also based on the same soft-state model. The sink periodically broadcasts hello message as well as other sensor nodes. Sensor nodes around the sink node regard that the sink has failed if they had not received hello message from the sink for $3 \cdot t_{expire}$. After the failure, the cluster of that sink is no longer preferable because it cannot receive packets any more. Thus, sensor nodes which take notice of its failure set cluster-pheromone values of all entries having membership of that cluster to 0, also abandon its cluster membership itself, and transmit a hello message. As the hello message, which indicates the sink failure, propagates over the network, sensor nodes participating in the failed sink's cluster also abandon their membership, and join another cluster.

2.2 Centralized control

As another control scheme to be compared with self-organizing control, we use centralized control scheme.

Younis *et al.* [26] have proposed a data-gathering scheme for sensor networks that assumes the existence of multiple sinks (for consistency with the terminology used in our self-organizing control, we use "sink" here instead of the "gateway node" used in their proposal). According to their proposal, sinks are assumed to be significantly less energy-, performance-, and memory-constrained than sensor nodes. They have assumed that sensor nodes have already divided in some way into the clusters so that each cluster has a sink, and the sink calculates the route from each node in its cluster to itself based on the residual power, state, etc. of sensor nodes. Then the sink transmits their previous- and next-hop node pairs and the state they should stay in next step (e.g., active or sleep state). This data-gathering scheme is definitely centralized, and the role of clusters is almost same as that of clusters in our self-organizing control. So this scheme is well suited to be compared with the self-organizing control. However, it describes only the routing and node-state

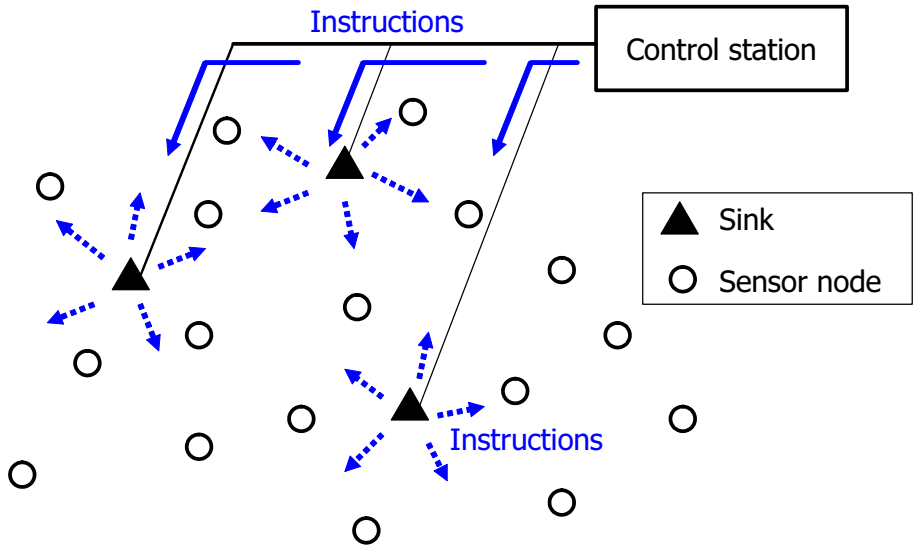


Figure 2: Overview of our centralized control.

management, does not provide how the sensor nodes should be apportioned into clusters, and some assumptions are not appropriate for large-scale networks, for example, a sink is located within the one-hop of an arbitrary node in that cluster. So we make some modifications to this scheme.

We assume the existence of multiple sinks and a control station. The station is connected to all sinks with wire, knows initial power and locations of all nodes and sinks, and manages the overall networks as shown in Fig. 2. The station first divides nodes into as many clusters as sinks, and a node belonging to cluster s_l must transmits their sensing data to the sink s_l. The command from the control station is transmitted from a sink to the sensor nodes in that cluster.

The control station first divides the node into clusters by Voronoi tessellations with sinks as basing points. In other words, the central station splits the sensor nodes into clusters such that each sensor node transmits a packet to the nearest sink. After clusters are formed, the station constructs routes from each node to its sink such that total link cost of the route is minimized. Link cost is assigned by the station to all the links between all node-and-node, node-and-sink pairs according

to:

$$C_{ij} = \begin{cases} \frac{E_{I_j}}{E_{R_j}} \frac{(4\pi)^2 d(n_i,n_j)^2}{\lambda} & \textit{if} \quad d(n_i,n_j) \leq \delta \\ \frac{E_{I_j}}{E_{R_j}} \frac{d(n_i,n_j)^4}{h^4} & \textit{if} \quad \delta < d(n_i,n_j) \leq r_{max} \\ \infty & \textit{otherwise} \end{cases} \tag{8}$$

where C_{ij} is the assigned link cost between node n_i and n_j. The term λ is a radio wavelength, h is the height of an antenna, and $d(n_i, n_j)$ represents the distance between nodes n_i and n_j. The threshold distance δ is a constant defined as $\delta = \frac{4\pi h^2}{\lambda}$, and r_{max} is the communication range of a node. Then the station transmits a command packet, which include the cluster memberships and route information, to sinks, and the sink relays the command packet with minimal transmission power so that all the sensor nodes in its cluster can receive the packet.

The detection of node failures is same as that of self-organizing control explained in Sect. 2.1.3. Each node transmits a hello message at a regular interval t_{hello}. If a node n_j cannot receive a hello message from node n_i for a fixed time t_{expire}, node n_i is deemed by n_i to have failed. In such a case, unlike with the self-organizing control, an explicit failure-indication packet must be transmitted to the control station, because the new routes must be provided such that packets circumvent the failed node. However, even when node n_i works normally, it is possible that hello packets from n_i does not arrive at a n_i's neighboring node within t_{expire} due to interference or transmission error. This possibility must be allowed for, because if such false positives occur repeatedly, data-collection rate will decrease and network connectivity will be lost. Preparing for such false detection, node n_j which detects n_i's failure memorizes that detection. If n_j could receive again a hello packet from node n_i, it deems the detection of n_i's failure must be false positive, and transmits a failure-recovery packet to inform the station about the false detection. The station then recomputes new routes and transmits them to the sensor nodes.

3 Evaluation and discussions

3.1 Simulation Conditions

In this section, we compare and evaluate the self-organizing control with centralized control in mainly robustness, through computer simulation experiments in a variety of perturbation scenarios. We implemented both two control schemes on ns-2 network simulator [27]. In a square monitoring region, whose side length is w, 300 sensor nodes are randomly distributed. Unless otherwise stated, w is 100 m. In addition to sensor nodes, four sinks are deployed, which are fixed at $(w/4, w/4)$, $(3w/4, w/4)$, $(w/4, 3w/4)$, and $(3w/4, 3w/4)$, respectively. An example snapshot of this configuration is shown in Fig. 3, where red nodes are sinks and blue nodes are sensor nodes. We used two-ray ground reflection model as the model of radio propagation, and the MAC and PHY layer follow IEEE 802.15.4 standard [28] targeted to sensor node radios, which is well known for its energy efficiency and low-rate communication. Forward error correction technology is not considered in our simulation in order to focus on the influence of transmission error on robustness, therefore a packet is discarded even if one bit error occurs. We set the parameters of sensor nodes as listed in Table. 1 by referring to [29]. Simulation parameters are also listed in Table. 2.

In the followings, sensor nodes send their sensed data to their sink in a multi-hop way at a predefined interval $t_{data} = 10$ s. They are not synchronized, and their transmission times are completely independent of that of the others. A key robustness metric, data-collection rate, is associated to this periodicity. When the number of deployed nodes in a monitoring region is N, the number of data packets generated in the time t_{data} is of course N. Thus, if in that time r packets successfully reach sinks, the data-collection rate is to be r/N.

As for the size of command packet in the centralized control, its size grows linearly with the number of nodes in a cluster according to:

$$\sum_i 6 \cdot e_i \cdot num_s + 7 \tag{9}$$

where e_i is the number of previous- and next-hop pairs assigned to node n_i belonging to a cluster s, and num_s is the number of nodes in the cluster s. We assume 6 bytes are for a pair, and 7 bytes are for a header. We observed this size of a command packet can easily exceed the value specified in IEEE 802.15.4. We therefore set *aMaxPHYPacketSize*, which determines the maximum length

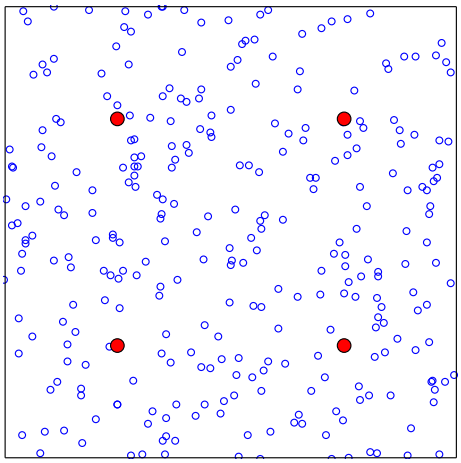


Figure 3: An example of a network. Blue circles represent sensor nodes. Red circles are sinks.

Table 1: Sensor node parameters.

Transmission power	0 dBm
Communication range	10 m
Frequency	2,450 MHz
Bit rate	250 kbps
Height of antenna	20 cm
Initial power	25 J
Buffer size	65 packets
Power consumption in transmission state	40.95 mW
Power consumption in reception state	45.78 mW

Table 2: 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-indication packet	10 bytes
Size of a failure-recovery packet	10 bytes
Size of a data packet	64 bytes

of a packet, to virtually infinity.

One of the most important parameter which has not addressed so far is transmission interval of, backward ant in the self-organizing control, and that of command packets in the centralized control, respectively. In the self-organizing control, too short the interval causes repeated interference through iterating forwarding the ants among sensor nodes, and too long an the interval does not construct pheromone distribution enough for data gathering. We tested transmission intervals of 10 s, 100 s, and 500 s, and selected 100 s because of its better balance between data-collection rate and power consumption. Transmission interval of command packets in a centralized control also has great influence on data-collection rate. In the interval is too short, command packets coming one after another result in a severe interference problem, and if the interval is too long, up-to-date control information which is necessary for adaptive behavior in changing environments does not sufficiently provided. We also conducted simulation experiments to study whether 1 s, 10 s, 100 s, or 500 s transmission interval yields best results and chose 10 s as the one yielding the best balance.

Table 3: Statistics of routes generated in the self-organizing control and the centralized control. 95% confidence intervals are also shown.

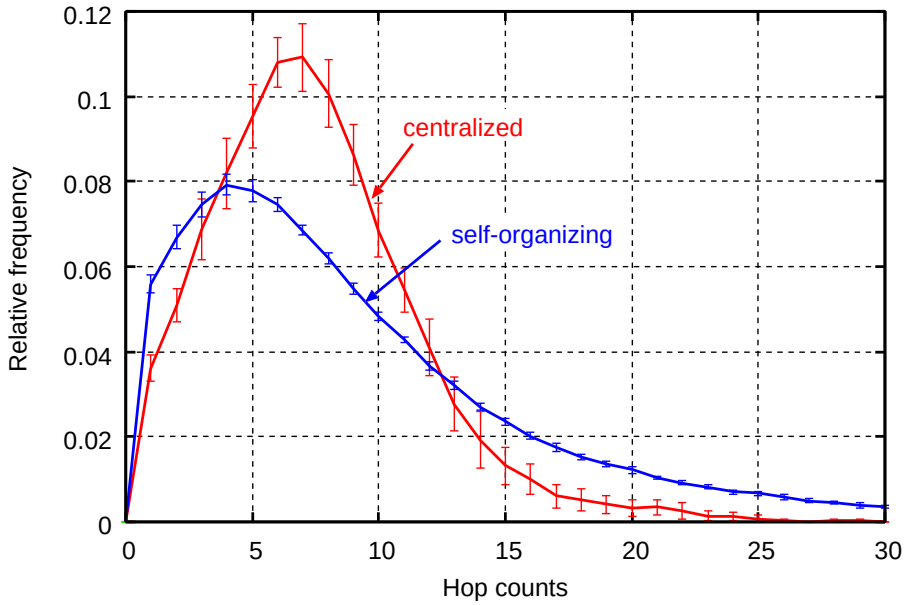
	Hop-Avg (hop)	Hop-Var (hop^2)	Delay-Avg (s)	Delay-Var (s^2)
Self-organizing	$9.1 \pm 3.4 \times 10^{-1}$	1.2×10^{2}	$2.3 \times 10^{-1} \pm 1.6 \times 10^{-2}$	7.6×10^{-1}
Centralized	$7.5 \pm 3.6 \times 10^{-1}$	1.6×10^{1}	$1.5 \times 10^{-1} \pm 8.6 \times 10^{-3}$	1.0×10^{-2}

3.2 Efficiency of data collection in ideal conditions

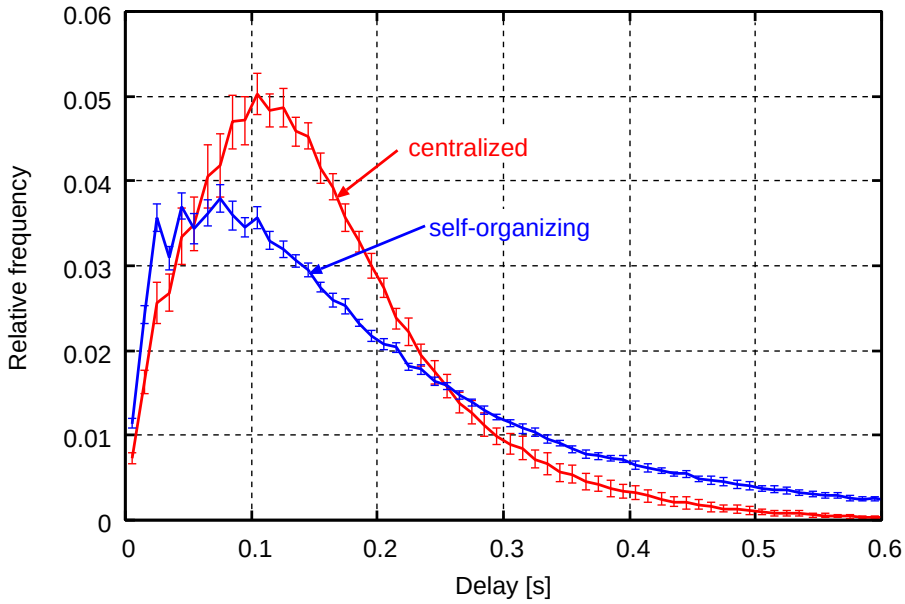
We first compared basic efficiency of the two schemes in terms of hop counts and delay in an ideal environment where no failures occur. Bit error rate (BER) is set to 10^{-5}. Hop counts reported here are of all routes between each sensor node and its sink generated over the simulation time over 15 trials. Delay is also of all the packets over the simulation time from transmission time to arrival time to sink. Their relative frequencies of distributions are shown in Fig. 4 with 95% confidence intervals. Actually, there are not remarkable differences in their distributions, and for average values from Table. 3 we can say the same thing. However, variances, which are also shown in Table. 3, differ substantially between the self-organizing control and the centralized control. Concretely, all the hop counts observed in the centralized control are not over 30 hops. However, for example, some packets produced in the self-organizing control experienced more than 300 hops. These interesting results suggest that in self-organizing control, quality of generated routes can fluctuate widely, i.e., low predictability and controllability. The lack of global viewpoint leads to difficulty in finding global optimum, and results in wide fluctuation.

3.3 Robustness evaluation

In this subsection, we focus on the main metric in this thesis, robustness. As already mentioned, robustness can be seen in the behavior of data-collection rate before and after perturbations in a sensor network. In the following, we evaluate the robustness against a wide range of perturbations, and what self-organizing control yield at the expense of predictability shown in Section 3.2.



(a) Hop counts.



(b) Delay.

Figure 4: Efficiency of routes generated by the self-organizing control and the centralized control.

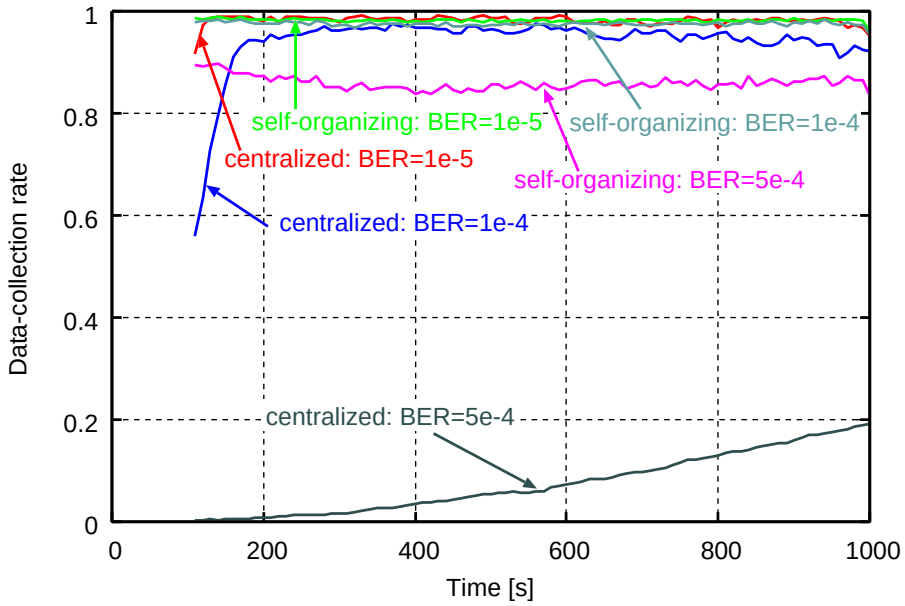


Figure 5: Changes of data-collection rate in various BERs.

3.3.1 Against transmission error

We conducted simulation experiments to study the robustness of both control schemes against transmission error, under the assumption that no node failure occurs. In Fig 5 where changes of data-collection rate over the simulation time are shown, a clear difference in robustness can be seen. In low BER environments, both the self-organizing control and the centralized control show high data-collection rate from the beginning. However, as the channel quality deteriorates, i.e., as BER becomes higher, only the centralized control takes more time for yielding high data-collection rate. This clear difference is mainly due to, as is well known for a weak point of centralized control, the loss of important control information, in our case, command packets. Even with the best network management, unless a sensor node receives that information, they do not adapt well to the network change going on, because the all actions necessary for adaptation are included only in the control information. That situation leads to the inconsistency occurred between the nodes which successfully received a command packet and the node which did not, and it brings routing error in our simulation, in which a forwarded packet was discarded by failing to find next-hop node for it in node's routing table. Actually the date-collection rate of the centralized control increases slowly with time, but this is attributed to only the fact that frequently transmitted command packets

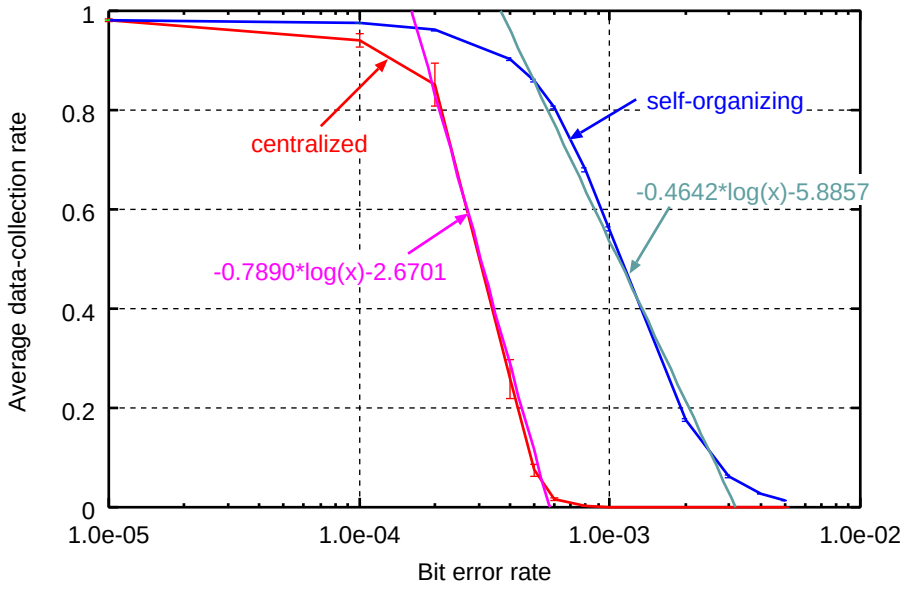


Figure 6: Data-collection rate versus BER.

compensate the packet loss.

The self-organizing control, on the other hand, kept comparatively high data-collection rate from an early stage of simulations. The main reasons of these results are that the nodes operates based only on information from neighboring nodes, and the exchange of information among them is repeated several times. In this case, if a packet is lost for some reason, it can be compensated by information from other neighboring nodes. In this manner, self-organizing control inherently has redundancy of control information.

This advantage of redundancy in self-organizing control is also obvious in Fig. 6, where average values of data-collection rate from 100 s to 1000 s are plotted against BER, and logarithmic approximation lines for their decays are also shown. The self-organizing control keeps data-collection rate above 80% about 3 times longer. In addition, the gradient in its decay is only 58% of that of centralized control. When the gradient is gentle, data-collection capability is not affected significantly in response to a small change of BER. For that reason, self-organizing control is more robust against transmission error.

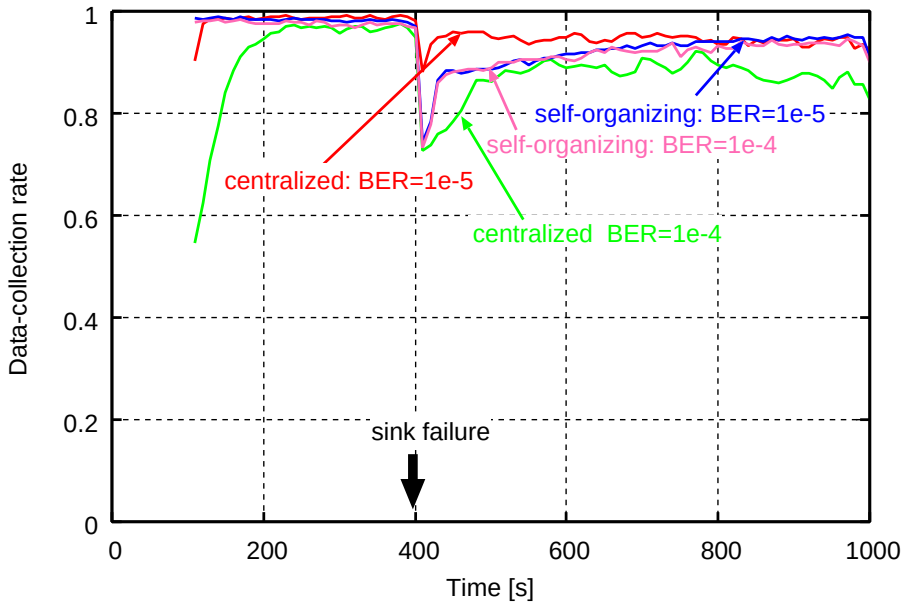
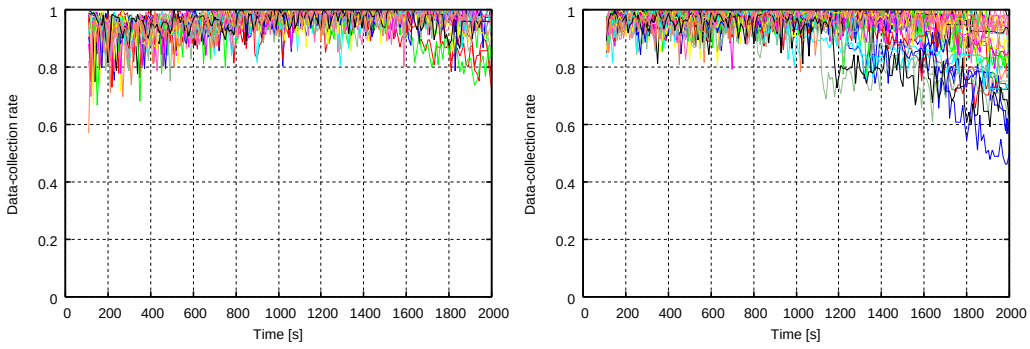


Figure 7: Recovery feature when sink fails.

3.3.2 Against sink failure

We next presented the result in Fig. 7 for the case in which a sink located at $(25, 25)$ fails at 400 s. Failure in this thesis is defined as a state in which neither transmission nor reception is possible. After the sink failure, data-collection rate once sharply fall to about 75% with the exception that the rate of centralized control with 10^{-5} BER falls to only 90%. The rate 75% means that one cluster suffered catastrophic damage (recall we have four clusters now, and the ratio of data packets gathered in a cluster is about 25%). The rate of centralized control in low BER situation not only fall just a little but also immediately recovers. The control station connected with sinks with wire become aware of the failure within a short amount of time (in our simulations, it is set to 0 s), then clusters are reconstructed and routes are recomputed to adapt whole network to the failure upon receiving the command packet. Sensor nodes immediately modify its cluster membership and routing table according to the instruction contained in the command packet upon receiving it. Then the data-collection rate recovers soon. Of course in the case channel quality is poor, the data-collection rate of the centralized control is unable to recover within the simulation time shown in Fig. 7, its reason has already stated in Section 3.3.1.

Different from centralized control, self-organizing control takes more time for the distant sen-



(a) Self-organizing control.

(b) Centralized control.

Figure 8: Results of each trial in both control schemes against random node failures.

sor nodes to adapt to the sink failure. In addition, because the network has no supervisor and no explicit instructions, some nodes prone to opposing actions based on its possibly inaccurate network conditions. For these reasons, in low BER situation, self-organizing control exhibits worse recovery feature than centralized one. In high BER environments, however, this relationship between self-organizing control and centralized control gets reversed, because self-organizing control inherently does not have important information which can bring serious and adverse influence if it dropped.

3.3.3 Against node failures

We simulated sensor node failures as follows to study the robustness of both control schemes. Sensor nodes operate autonomously for periods of time, and are not accessible for battery replacement because of physical deployment locations. For these reasons, failures are likely to occur frequently in sensor networks. Defining p_{fail} as a failure rate for each sensor node every second, we first generated random failures from $t = 100$ s according to p_{fail}. When we tested random failures in a 100×100 m^2 monitoring region with 300 nodes, the difference of robustness between the self-organizing control and the centralized control was not clear well due to the connectivity degradation through continued node failures. So we temporarily used a narrower monitoring region with 50×50 m^2 while keeping the number of nodes and sinks.

The results of each trial shown in Fig. 8 indicate distinct aspect for robustness. In Fig. 8(a), there is a little difference in the data-collection rate among trials in the early state of simulation.

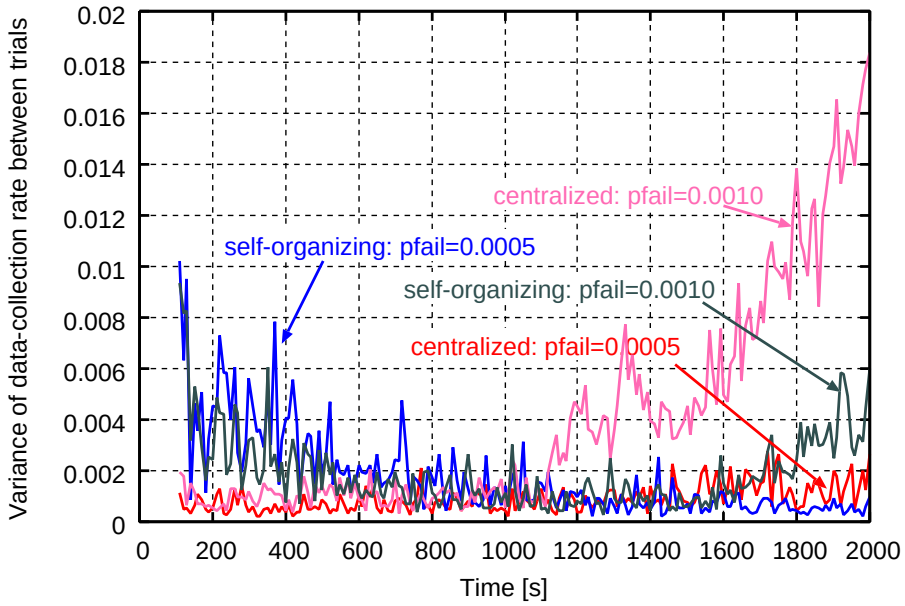


Figure 9: Variances of data-collection rate among trials.

But this difference becomes small with time, although the rate again starts increasing in the end of simulation. The valiance of the self-organizing control shown in Fig. 9 is small and not so sensitive to the failure rates. However, in the centralized control shown in 8(b), some lines in centralized control diverge greatly from others. These lines lead higher variance of data-collection rate as shown in Fig. 9. This high variance in centralized control means difficulty of predicting data-gathering capability in harsh environments, although all the plots use same parameters.

We also investigated the influence of concentrated and simultaneous node failures. We selected particular coordinate and made all the sensor nodes within 10 m of the coordinate fail simultaneously at $t = 400$ s. The resultant changes in data-collection rate are shown in Fig. 10, from which we can see that the failures have only a little influence on the self-organizing control when the coordinate is $(15, 15)$ or $(20, 20)$. Even when the coordinate is $(25, 25)$, which just corresponds to the lower left sink, data-collection rate shows basically a tendency to recover, although the rate in some trials does not recover depending on the node deployment. On the other hand, once data-collection rate in centralized control drops, it never recovers at all, because failure-indication packets did not reach the control station. The station considers nodes that had failed to be working properly, and therefore, distributes routes that included the failed nodes. We observed only one

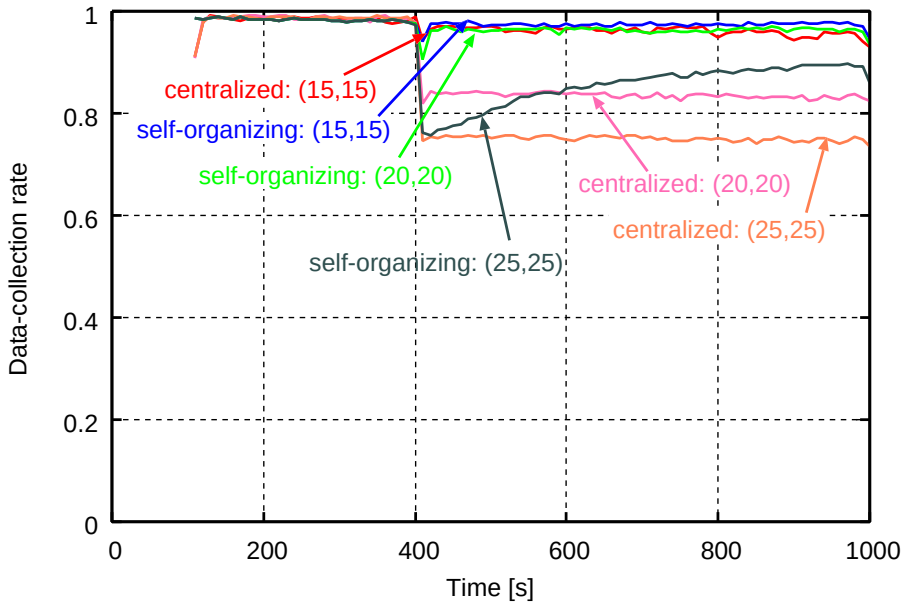


Figure 10: Robustness against concentrated and simultaneous failures.

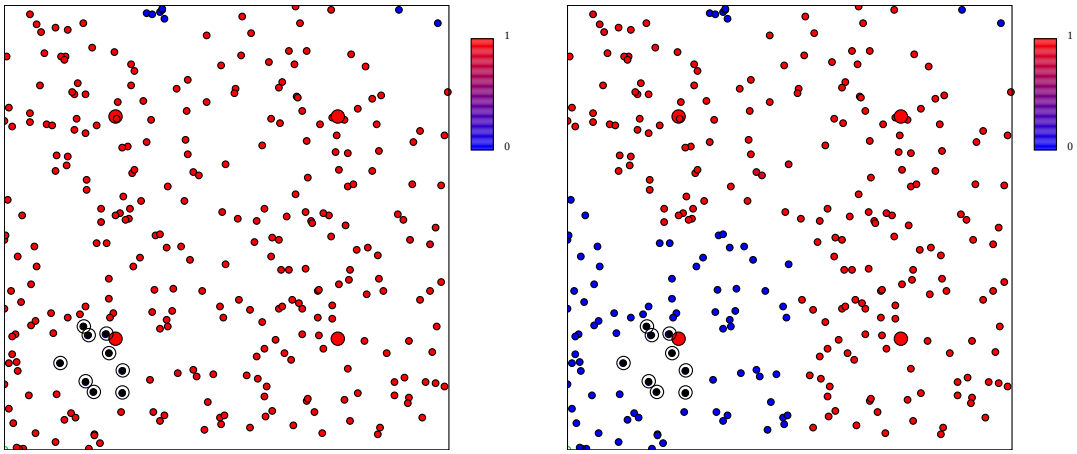
node failure can create cluster level influence in centralized control.

Figures 11(a) and 11(b) show the states of the network when the center of a failure circle was $(20, 20)$. The small circles there are sensor nodes, and larger circles are sinks. Double circles represent failed nodes, and the color of the other nodes shows the data-collection rates of themselves. Red indicates that all the packets transmitted from the node can reach a sink, and blue indicates no packets reach the sink. Most of the sensor nodes in Fig. 11(a) other than the failed ones show data-collection rates of about 100% in self-organizing control. In centralized control, however, the failures have a considerable influence at the cluster level and many sensor nodes are unable to transmit packets to their sinks. It is undesirable for sensor networks to be unable to gather information from particular areas.

3.3.4 Against node addition

Node addition is sometimes necessary during operating time of sensor networks to keep monitoring capability for the case where the some nodes deplete their batteries or break down. The addition brings a change into the network, which can be evaluated in terms of robustness.

The control station in the centralized control must have geographical information of newly



(a) Self-organizing control. (b) Centralized control.

Figure 11: Influence of concentrated and simultaneous node failures on monitoring capability. Sensor nodes surrounded by a ring are failed nodes. The color of the sensor nodes represents how much these nodes brought packets to the sink after $t = 400$ s. Red means the all the packets transmitted from the node can reach a sink, and blue means packets did not reach it.

added sensor nodes to calculate cluster memberships and routing information. Therefore, added nodes must inform the station about their locations, possibly in a multi-hop way. However, other nodes do not have entries for the added nodes in their routing table. Thus, some mechanisms are required so that their geographical information is delivered to the control station. We assume the following mechanism, and the mechanism is illustrated in Fig. 12.

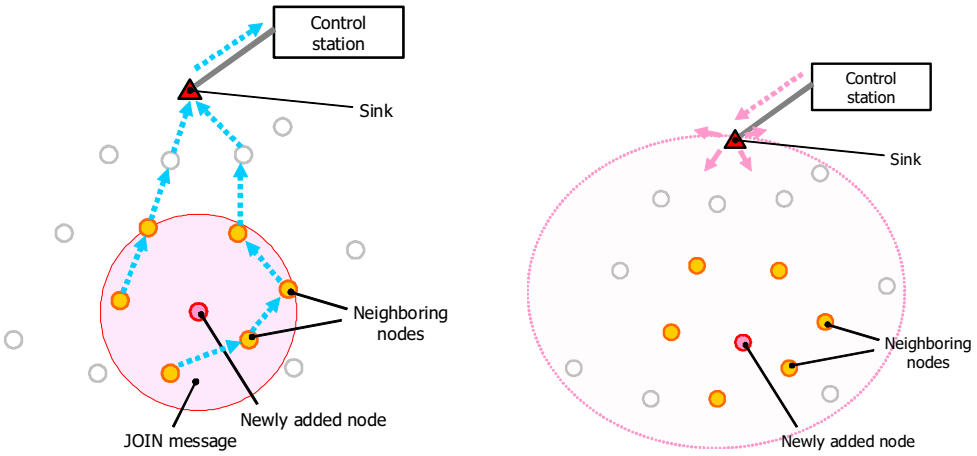
(1) A newly added node broadcasts JOIN message, which include its geographical information.

(2) The message is forwarded by its neighbors through the routes to the control station that they have already had.

(3) The control station newly calculates cluster memberships and routing information based on the received JOIN message.

(4) The control station transmits that information to the nodes.

Upon receipt of the its membership and routing information from the control station via a command packet, the added nodes indeed join the data collection. In the case where an added node



(a) A newly added node transmits JOIN message, which is forwarded to a control station.

(b) The control station transmits new control information to the nodes.

Figure 12: Mechanisms for added nodes to join the data collection.

cannot receive the command packet, after the time $t_{\text{pos}} = 1$ s from its previous broadcast, it rebroadcasts a JOIN message again, until it gets to join the data collection.

Fig. 13 portraits a scenarios when 30, 60, 90 nodes are added randomly into the network at $t = 400$ s. In the centralized control, each added node starts transmitting a JOIN messages after waiting for random time within 10 s to avoid collisions among other JOIN messages. In self-organizing control, even after $t = 400$, the influence of the addition cannot be seen clearly, and the results are impervious to the number of added nodes because added nodes do not need to take special handling. On the other hand, in centralized control, data-collection rate falls dramatically, and its drop gets bigger with the number of added nodes.

The paramount cause for the centralized control being less robust to node addition is interference caused mainly by command packets. The control station, upon receiving a JOIN message, immediately recomputes and transmits new cluster memberships and routing information. Thus the frequency of transmission of command packets markedly increases with the increase in the number of added nodes. Then, reception rate of command packet decreases due to the interference, and not receiving the command packet at that time leads to the situation where the node cannot adapt the possibly substantial changes due to node addition. As a result, data-collection rate sharply decreases just after the addition.

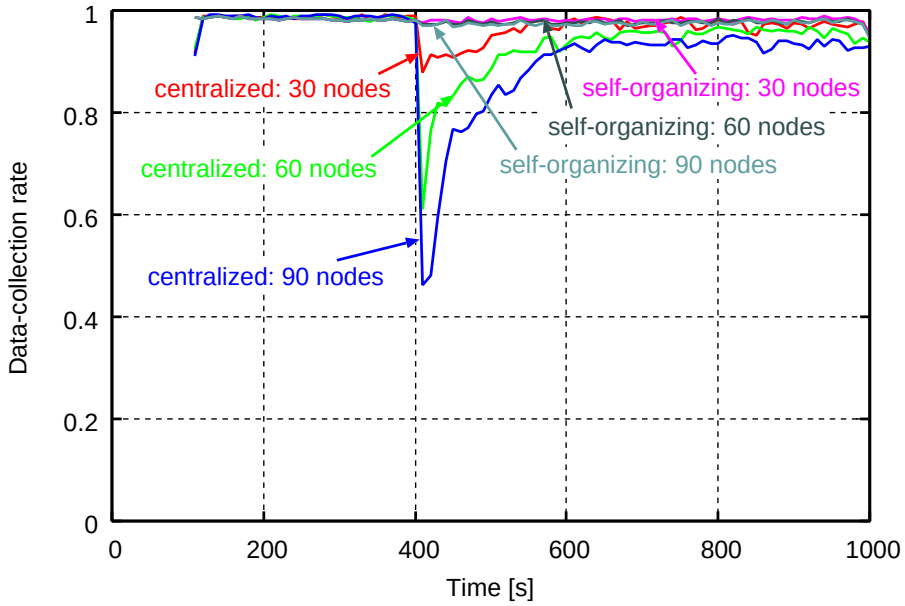


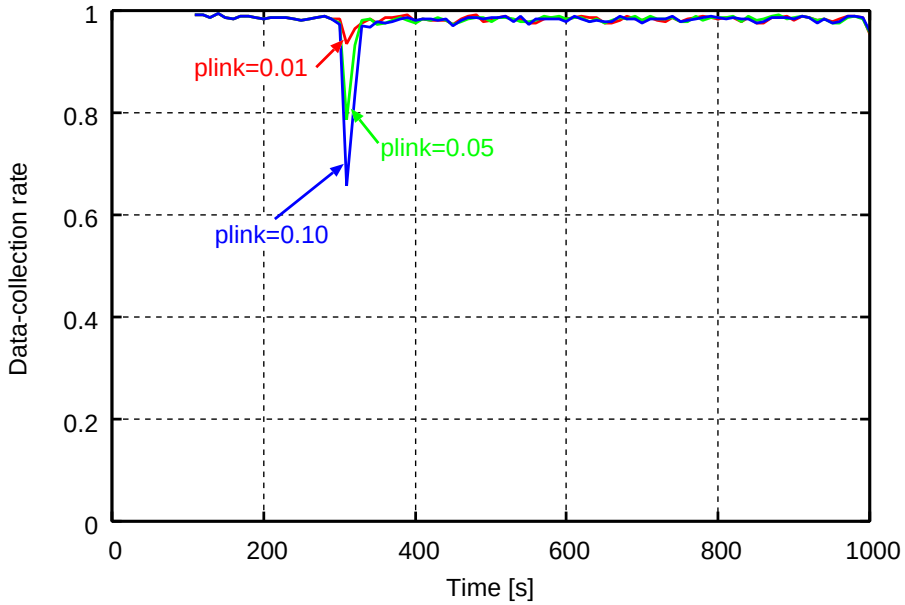
Figure 13: Data-collection rate when new nodes are added to the network.

3.3.5 Against link disconnection

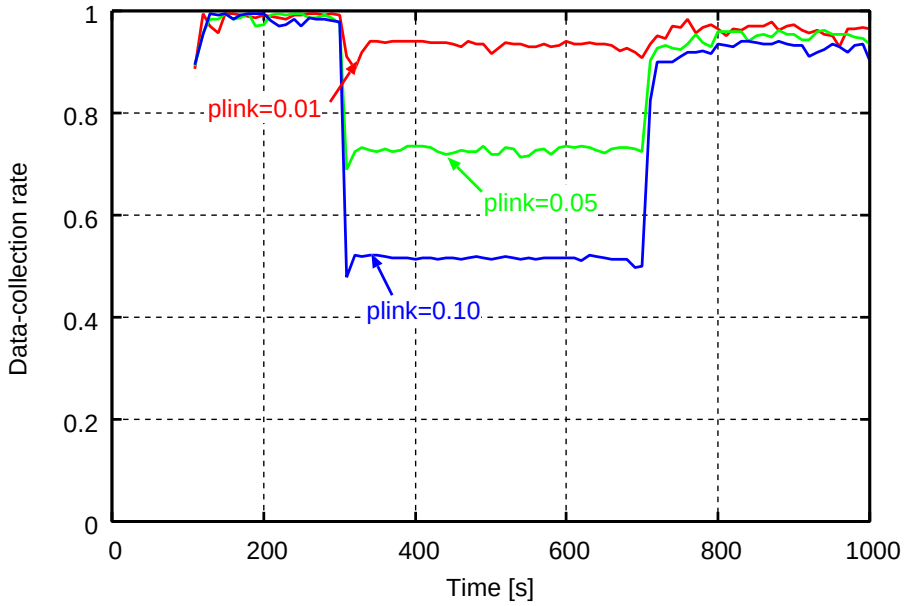
In wireless networks, intermittent link disconnections can occur due to interference, obstacles, etc. In the case where the link between node n_i and n_j is disconnected but the link between n_i and n_k is connected, n_i's statuses which n_j considers and which n_j considers are possibly inconsistent. To study the difference in robustness between the two schemes, we disconnected the percentage of links. We assume that each node has links to an arbitrary neighboring node, and disconnect each link with probability p_{link} in both directions. We conducted this disconnection for all nodes. The duration of the disconnection is 400 s, from $t = 300$ s to $t = 700$ s.

In the results shown in Fig. 14, data-collection rate of the self-organizing control immediately recovers to the rate before the disconnection, experiencing declination in a short time though. The centralized control, on the other hand, experienced great influence of the disconnection. What occurred in the centralized control are massive detections of node failures. Hello packets do not reach a neighboring node under the link disconnection, the neighboring node detects this situation as a failure. That is, sensor node cannot identify failure and link disconnection in our centralized control. Furthermore, after the detection, the neighboring node transmits a failure-indication packet, which is actually false-positive detection, to the control station. As a result, the control

station does not provide routes for the node which is considered as failed, and the packet from the node will be discarded. This is the main reason for the decay in Fig. 14(b).



(a) Self-organizing control.



(b) Centralized control.

Figure 14: Influence of link disconnection on data-collection rate.

4 Dependency of control information

The evaluation conducted in Section 3 is just a cast study, but it allows to us to gain important insights about why self-organizing controls yield better robustness. The commonly observed nature through the evaluation is that when the state of network changes, although the centralized control suffers great and bad influence on data-collection rate, i.e., robustness, the self-organizing control is less affected by the changes. More precisely, the nature is observed when the entities, i.e., control station or nodes in a network, do not see the correct state of the network, or even they recognize incorrect state as correct. We discuss these issues in Section 4.1. In Section 4.2, we evaluate the influence of the incorrect view of individual nodes on the network robustness, and in Section 4.3, we address why self-organizing control enables the influence to be localized. In Section 4.4, we discuss why self-organizing control is robust.

4.1 Factor in differentiating robustness

In the evaluation represented in Section 3, there was a significant difference between the self-organizing control and the centralized control. We would like to explain this trend in terms of "dependence on control information". In this case, "dependence" is almost the same meaning of that used in fault management [30] in which dependency is a relation of that an error or fault in an object may cause an error in another object. We define the control information as the information exchanged between entities of a given network to coordinate their joint operation.

For example, in the evaluation of robustness against transmission error shown in Section 3.3.1, sensor nodes in the centralized control could not receive control information of the network, command packets, in higher BER situation. Sensor nodes in the centralized control completely depend for comprehending network state on the control information from a control station. Control information explicitly prescribes the action of sensor nodes based on the current network state, and the node participates in data collection based on the order, believing that other nodes follow the order. In the case where some sensors can receive the command packet and others cannot, the formers acts based on the old order, and inconsistent views of the routes can be introduced among them. Such inconsistency makes sensor nodes lose their next-hop node for a received packet, resulting in packet loss, and the network gets stuck in the pathological state until their views get consistent. However, dependence on the control information in self-organizing control is weak. What deter-

mines their action is themselves and the node which acts based on old or incorrect information influence only their neighbors.

In Section 3.3.1 and 3.3.2, we saw the case where the control station saw the correct picture of the entire network. But from Section 3.3.3 through 3.3.5, even the control station did not comprehend the correct state of the network actually. This is because the control station also depends on the control information from the nodes in the network. The control station constructs precise view of the whole network by integrating each piece of network state information. That is, the problem of the dependence is that the control information from possibly unreliable nodes in environments where reliable communication is not guaranteed plays a critical role in generating control at the control station. In Section 3.3.3, failure-indication packets, which notify the correct state of the network to the command node, did not reach the control station, resulting in catastrophic failure of cluster's data-collection capability. In Section 3.3.5, one node has perception that a neighboring node is operating correctly, while another node considers the neighboring node is faulty, resulting in transmission of failure-indication packets although no nodes have failed. In this way, information which does not reflect correct state of the network brings vulnerability to the centralized control.

Of course, in the node level, the self-organizing control is just like the centralized control, meaning that individual nodes possibly have erroneous understanding about the state of the network. However, the individual nodes affect just a surrounded environments or neighboring nodes, because the nodes have only a partial view of the network, and do not transmit and receive explicit control information. Due to such behaviors, the influence brought by individual node is much smaller than the centralized control. Unfortunately, we have not clarified the influence of the erroneous information from individual nodes. So the next section, we verify our idea by deliberately injecting incorrect information to the network.

4.2 Influence of incorrect information

What we would like to demonstrate clearly is that how much influence brought by information from individuals, and possibly unreliable nodes affect the behavior of whole network. Influence of the information which does not correctly reflect actual network state is still ambiguous. So in this section, we deliberately inject the spurious information to definitely show the influence of the information from individual nodes on the function of the network. At first, in centralized

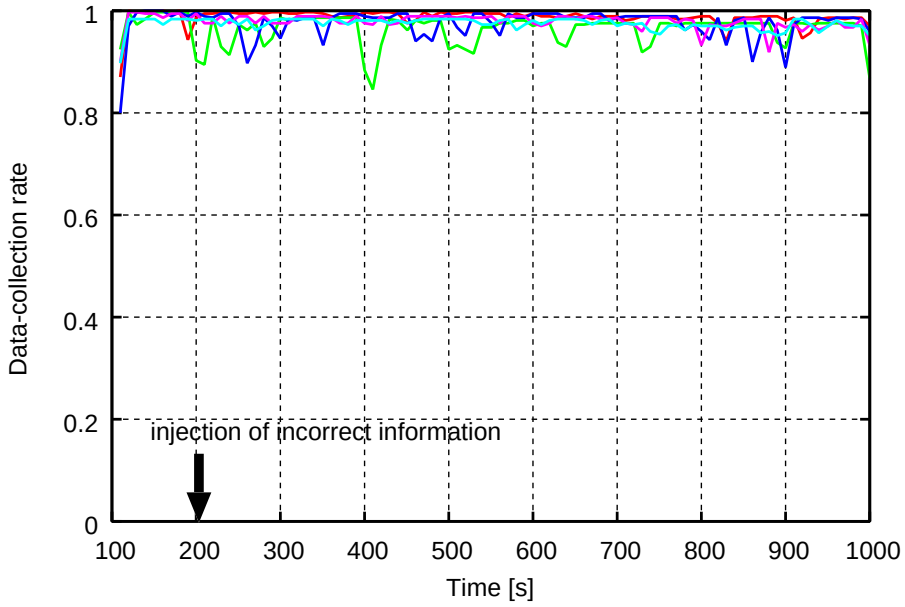


Figure 15: Results of injecting information of false-positive failure detection.

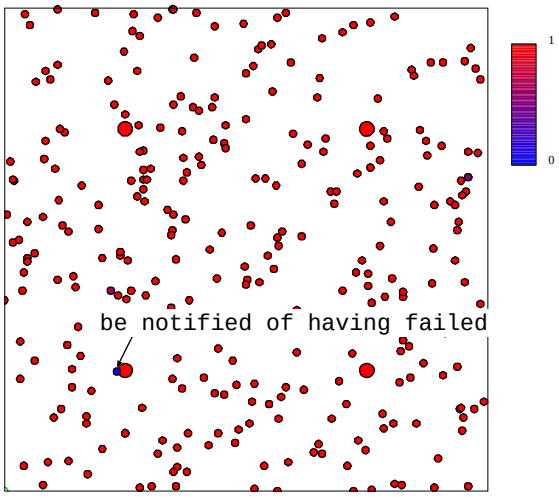


Figure 16: State of the network when injecting information of false-positive failure detection.

control, we considered two scenarios, 1) we inject information of false-positive failure detection, which is the information that the node properly working is detected as failed, and 2) false-recovery information, which is the information that the node which has failed is detected as recovered.

We deliberately inject incorrect information at $t = 200$ s that the node nearest to the coordinate $(25, 25)$ has failed. We cannot see the fluctuation or drop of data-collection rate due to the injection from the result shown in Fig. 15. Actually, from Fig. 16 where the data-collection rate of each node can be seen, the node which is wrongly detected as failed cannot send its packet to the sink, because the control station did not consider the failed node to join data collection. However, routing information is supplied to the other sensor nodes correctly, thus the influence of the erroneous information is limited.

Next, we tested the scenario where node recovery information is injected which is actually wrong. We at first 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 17 shows the results of five trials, and the behavior of data-collection rate are different among them, i.e., are different according to the node deployment. A blue line and a red line clearly fall just after the injection of erroneous information at $t = 200$ s. Given this factor, focusing on the trial of the red line, we visualized the data-collection rate of the individual nodes from the time when node fails ($t = 160$ s) until the injection ($t = 200$ s), and from the injection ($t = 200$ s) to the end of simulation ($t = 1000$ s), respectively in Fig. 18. As shown in Fig. 18(a), the influence of the node failure can be limited. However, after the injection, data collection in most part of a cluster becomes impossible.

Self-organizing control does not have the explicit failure indication, or failure recovery indication. So we could not compare directly with the centralized control in terms of the influence of erroneous information. Instead, we used the indication of the sink failure, which is a message of explicitly indicating sink failure to the neighboring nodes by using a hello message. We made a sensor node nearest to the coordinate $(25, 25)$ transmit the sink-failure indication. This indication is spread over a cluster through forwarding by nodes which receive the indication.

A spurious sink-failure indication is injected to the network at $t = 200$ s, but we cannot see clear difference of data-collection rate before and after the injection of false sink-failure indication in Fig. 19, where data-collection rates of five trials are depicted. In our self-organizing control, sensor nodes invalidate their membership to its cluster as described in Section 2.1, and negative

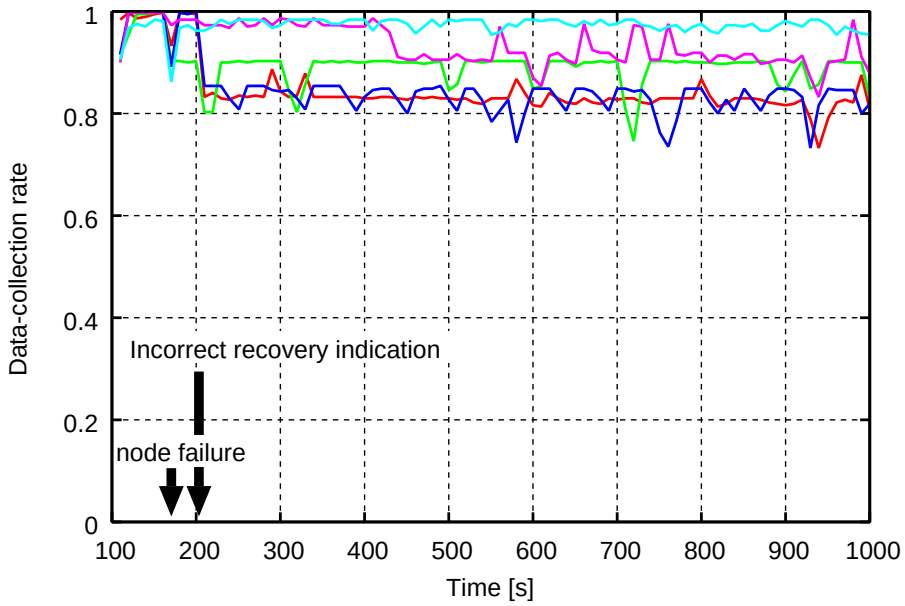
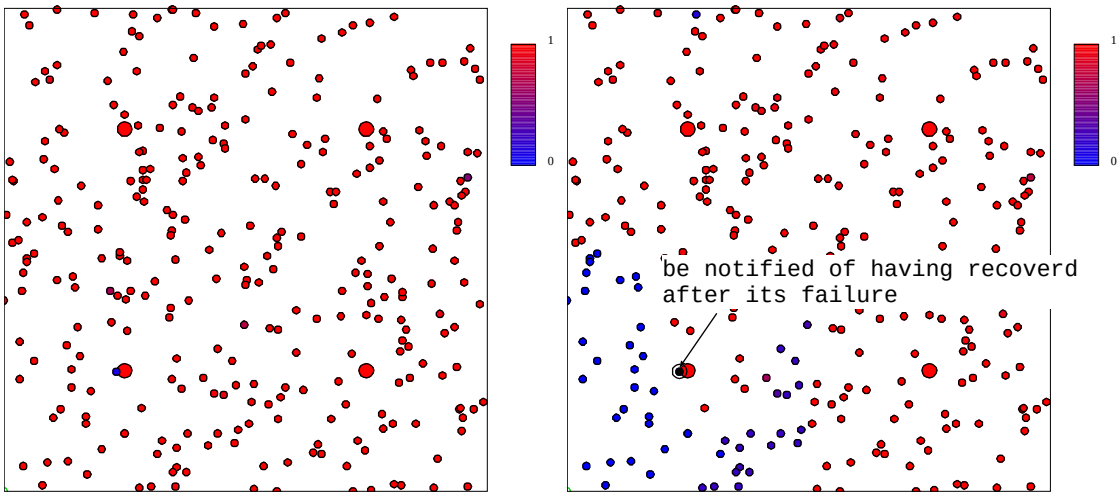


Figure 17: Results of injecting false-recovery information.



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

(b) From $t = 200$ s to $t = 1,000$ s

Figure 18: State of the network when injecting false-recovery information.

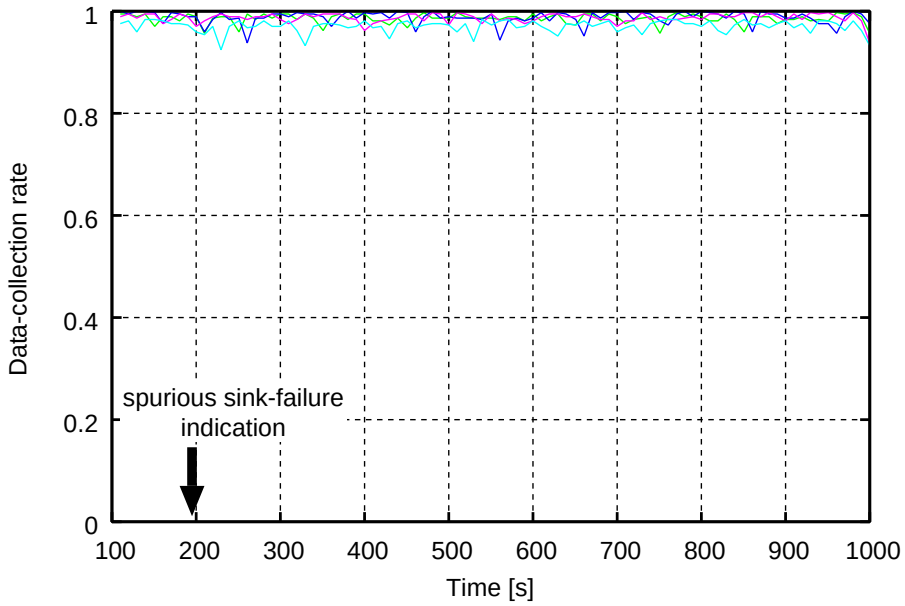


Figure 19: Influence of erroneous sink failure indication.

influence was expected due to the dynamic change of cluster membership. However, contrary to our expectation, their cluster memberships are returned to that before the injection. In other word, correct information from other nodes adjusts the situation caused by erroneous information naturally, and this fact contributes to the robustness of self-organizing control.

4.3 Localization of incorrect information

Individual nodes in the centralized control depend on control information from a control station, and the control station also depends on control information from the individuals. As a result, information from a node can bring considerable influence on the control for a whole network. In the self-organizing control, we cannot be denied that individual nodes have dependence on control information from neighboring nodes. If so, we guess that the information in self-organizing control is also propagates to the entire network by iterating communication among neighboring nodes. That is, is control information in self-organizing control localized? or propagated throughout the network?

Some research on information diffusion has been already studied so far [31-33]. However, these research determine whether information is propagated or not only by binary metric, which

is whether the information reach a node or not, or assume that propagation stops when the node forwarding the information communicates with the node already having the information. So they are not suitable for our aim. Thus, we modeled the extremely simple information diffusion as the followings.

(1) Each node has information value which scales running from 0 to 1.

(2) Nodes are arranged regularly on all cells on 2-dimensional lattice.

(3) Nodes can communicate only with its eight nearest neighbors.

(4) Information value at next time step depends on the current information value of neighbors.

(5) Nodes continue communication forever.

First through fourth assumptions above reflects that nodes in self-organizing control only affects its surrounding environments without explicit control messages. Fifth assumption considers nodes keep communicating with their neighbors to recognize up-to-date (possibly partial) network state.

Under these assumptions, we make judgment whether the information is localized or not observing information value of individual nodes. In our information-diffusion model, node n_i has a square matrix F with size s (we call this matrix as a filter inspired by image processing), which represents which node affects the information value of n_i. We assume the size s is an odd number because the filter is considered to have isotopic communication capability. Each element in a filter F means that how much information value is diffused from a node to itself. For example, the element at the center of the matrix $\left(\frac{s-1}{2}, \frac{s-1}{2}\right)$ represents that how much current information value of itself is affect its information value at the next time step, and its right element $\left(\frac{s-1}{2}+1, \frac{s-1}{2}\right)$ represents that how much information value is given from the right-handed node to itself in the next time step.

We define $I_t(x_i, y_i)$ as information value the node n_i located at (x_i, y_i) has at time t. We also define that the information value node n_i has at next time $t+1$ is calculated as:

$$I_{t+1}(x_i, y_i) = \phi \frac{\sum_{m=-\frac{s-1}{2}}^{\frac{s-1}{2}} \sum_{n=-\frac{s-1}{2}}^{\frac{s-1}{2}} I_t(x_i+m, y_i+n) in(x_i+m, y_i+n) F(\frac{s-1+2m}{2}, \frac{s-1+2n}{2})}{\sum_{m=-\frac{s-1}{2}}^{\frac{s-1}{2}} \sum_{n=-\frac{s-1}{2}}^{\frac{s-1}{2}} in(x_i+m, y_i+n) F(\frac{s-1+2m}{2}, \frac{s-1+2n}{2})}. \tag{10}$$

where

$$in(x, y) = \begin{cases} 1 & \textit{if} \quad 0 \leq x \leq D \textit{ and } 0 \leq y \leq D \\ 0 & \textit{otherwise.} \end{cases} \tag{11}$$

The term ϕ is an attenuation factor, representing the attenuation of information value over time.

In self-organizing control, each node determines its action by itself based on the information from its neighboring nodes and from itself. Typically, information from one node and that from other nodes can be considered to be treated equally. In consideration for that, we use a filter F such that:

$$F = \begin{bmatrix} 1 & \dots & 1 \\ \vdots & \ddots & \vdots \\ 1 & \dots & 1 \end{bmatrix}. \tag{12}$$

With this filter, information value of a node is updated to the average information value of its eight nearest neighbors and itself.

We observed the information diffusion of self-organizing control by using the above model. From time step $t = 0$ to $t = 10,000$, the node at $(0, 0)$ transmits information, which value is constantly one. Figs. 20 through 22 depict the influence of matrix size s on information diffusion without attenuation of information value over time ($\phi = 1$). The x-axis shows the time step, and the y-axis shows the ratio of cells which have information value more than threshold given as a parameter in this figure. As expected, the bigger the size of matrix, the faster the information diffuses over the network.

With all the sizes in Figs. 20 through 22, the information transmitted from only one node located at $(0, 0)$ finally diffused over the network, that is, all the node finally have the information value of one. So we introduced attenuation of information value over time. We set $s = 3$, which is the minimum size for the matrix F, and ϕ is selected among $\{0.99, 0.999, 0.9999\}$ as a parameter. The term $\phi < 1$ reflects the fact that old information becomes meaningless as time goes by. The node located at $(0, 0)$ continues transmitting information by 10000 time step, and it stops the transmission. The results of information diffusion are shown in Figs. 23 and 24, where the x-axis is the time steps, and the y-axis is the ratio of number of nodes which have more than threshold value 0.1 and 0.5, respectively. We can see that the information diffusion stops, and localized.

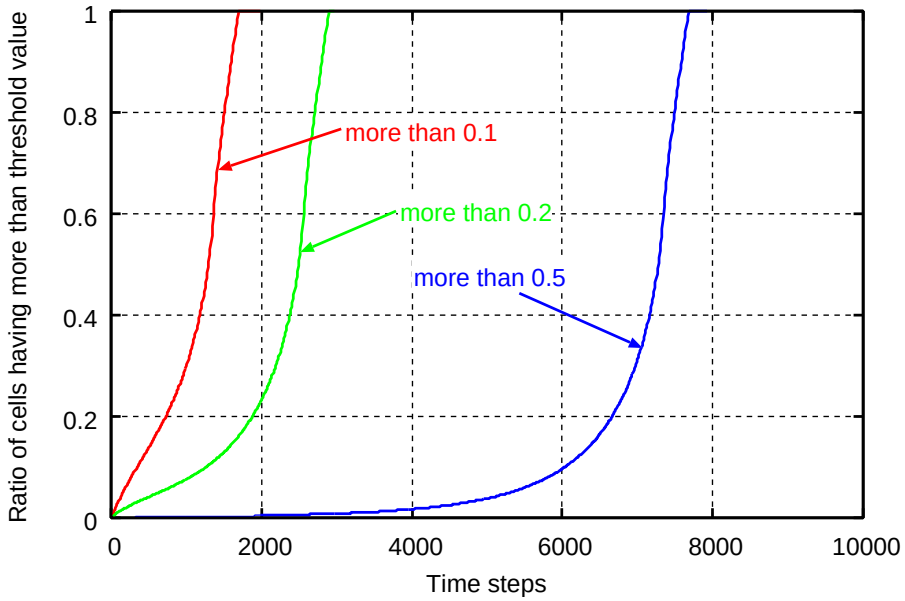


Figure 20: Influence of matrix size ($s = 3$)

Through this simple models described in this section, we saw the localization of information is attributed to two points. First, the size of matrix, which means each node only has partial view of the whole. Second, information attenuates over time. This is inherent nature of the information, that is, old information is meaningless.

4.4 Discussion

Centralized controls are generally composed of a control station and its controlled object, i.e., nodes. The control station governs the behavior of them, and sends directions to them to achieve the behavior. However, although the station must have a whole view of the network in order to determine the behavior of the nodes, the station generally does not have such capability especially in large-scale network. Thus the station depends on individual nodes for the collection of each piece of state of the network, and the station integrates that information to draw a precise picture of whole network. Generally, individual nodes cannot guarantee sufficient reliability to comprehend, memorize, and transmit such important information. Even so, they can contribute to normal operation of the network in the static or not harsh environment, but in the variable environment, they frequently send unreliable, incorrect, misguided control information to the control station.

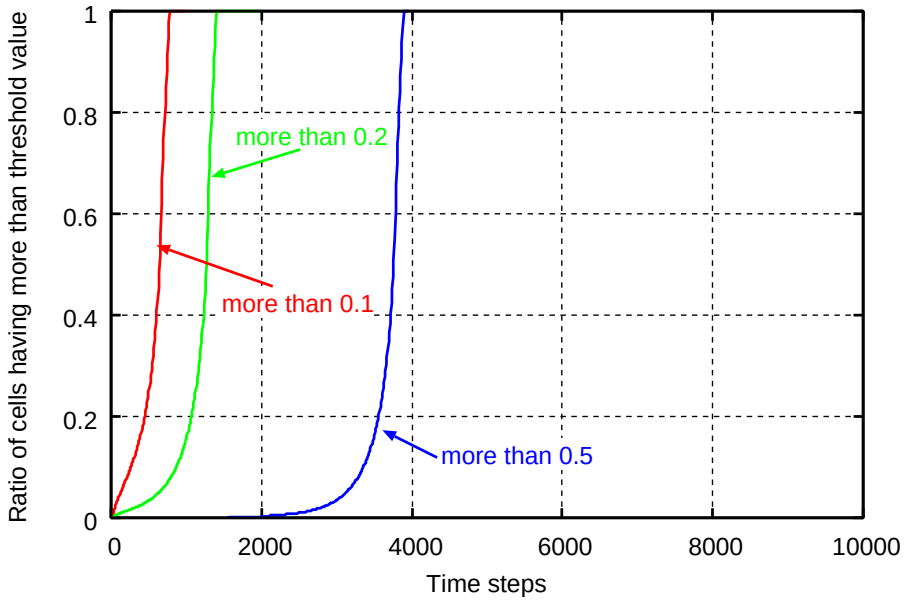


Figure 21: Influence of matrix size ($s = 5$)

In such cases, the control station may not be sure which received control information should be taken into account to adapt the environment. The strong dependence of a control station on control information from individual unreliable nodes is actually a big problem as well as the well-known dependence of individual nodes on control information from the control station, for robustness.

Self-organizing control, on the other hand, inherently does not possess strong dependence like centralized control. Control information from nodes in self-organizing control does not reflect all the nodes' states or information. Each node acts only based on its local environment and on information from its neighbors. So if a neighboring node generates incorrect information, resultant influence is only localized as discussed in Section 4.3. Self-organizing control which based only on nodes' partial view actually sacrifice, for example, efficiency of data collection, immediate recovery from some failures due to the lack of entire view and of explicit control information. However, by being satisfied with good enough performance, and by discarding aim for optimality, self-organizing control can lessen the strength of dependence on control information. In harsh environments where ambiguity can stems from the inconsistent control information from individual nodes, reliable communication cannot be expected, and composed of unreliable nodes, the network should focus on generating preferable behavior as a whole while eliminating dependence on

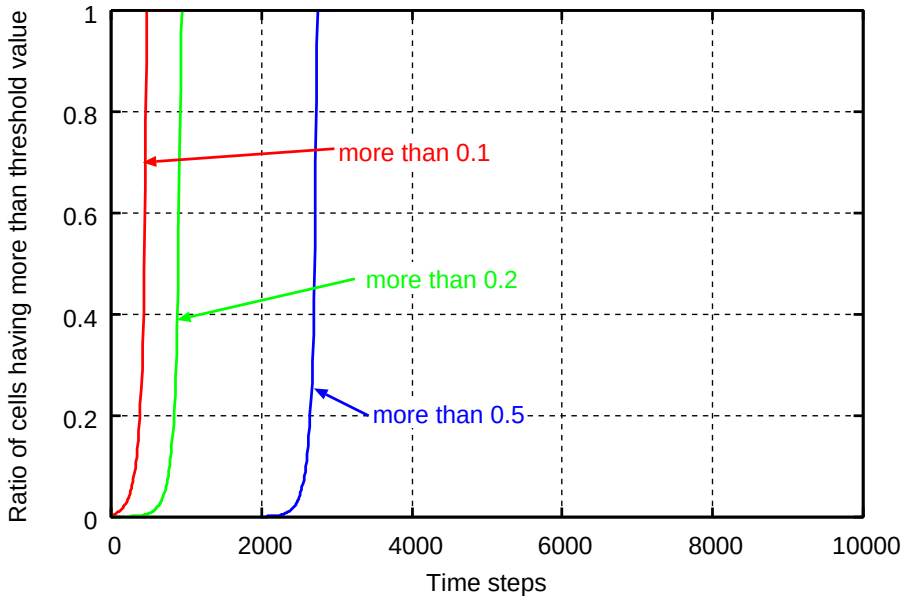


Figure 22: Influence of matrix size ($s = 9$)

control information from the nodes in the network. This is done just by self-organizing control.

We note that it is obvious to enhance robustness of centralized control, as you are probably aware. For example, in our centralized control, optimal interval of transmitting command packet and strong FEC functionality can help centralized control lessen the adverse effect of BER. Multi-path routing or flooding improves the reliability for communication between sensor nodes and sinks. But it is reported that such countermeasures to designed-for perturbations introduce hypersensitivity to unanticipated perturbations [18, 34]. That is, such countermeasure does not necessarily improve its intrinsic robustness, and easily invite another type of vulnerability, which designers are difficult to recognize, as a cat-and-mouse game.

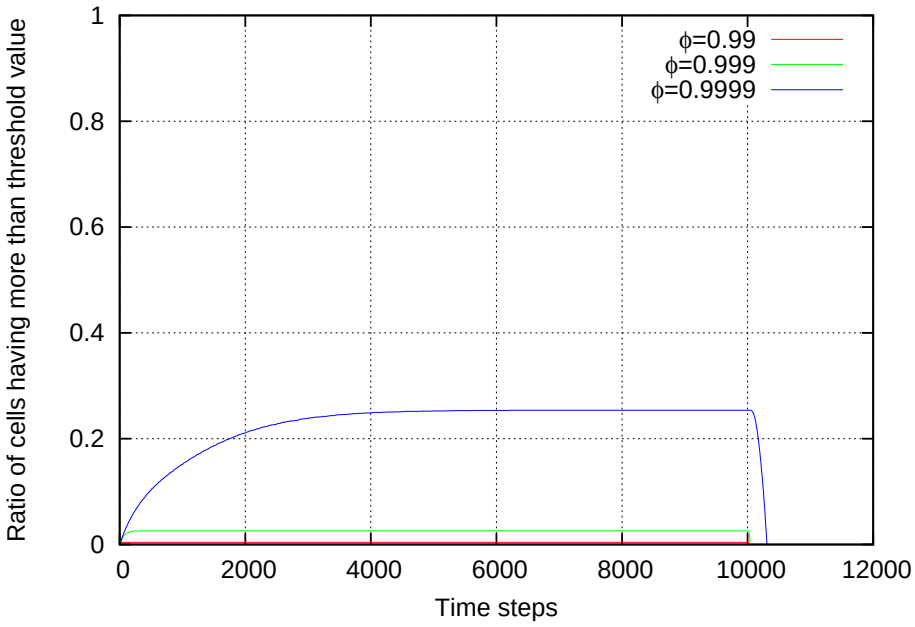


Figure 23: Ratio of cells which has more than information value of 0.1

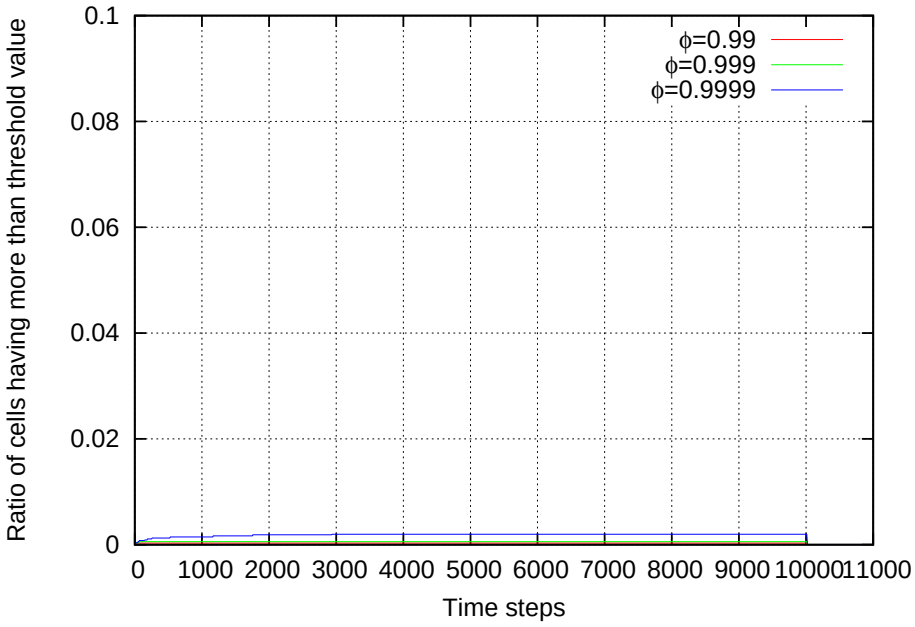


Figure 24: Ratio of cells which has more than information value of 0.5

5 Conclusion

In spite of its growing interest, there are a lot of things which are remained incompletely understood regarding self-organization. In this thesis, we studied robustness of self-organizing control against a wide range of perturbations with comparing to the centralized control, and we tried to tackle some of the important problems. The first one is "is it true that self-organizing control robust?" We quantitatively demonstrated its answer by using extensive scenarios. Although this result is not surprising, self-organizing control has obvious benefit on robustness especially in the system with highly changing environment, while reducing predictability of the system. "Why self-organizing control is robust" and "what factors yield robustness to self-organizing control" are the next problems we addressed. Based on the results obtained from the simulation experiments, we draw the conclusions that dependency on control information in the system plays a critical role in whether to provide good robustness or not. In the network composed of possibly unreliable nodes and located in harsh environment, decreasing dependency on control information from the nodes is critical to yield good robustness. Self-organizing control inherently has such property.

We compared only two control mechanisms and are aware that some readers will doubt the generality of our conclusions. So we have to mention about that. Actually we cannot be denied that our evaluation of robustness has protocol dependence. However, our explanation to the results, vulnerability of centralized control and the good robustness of self-organizing control, are derived from essence of each control. Centralized control has no other choice but strongly depending of the control information, and self-organizing control inherently does not have such strong dependence. The foundation of our generality is based on that.

Acknowledgements

I would like to greatly appreciate to my supervisor, Professor Masayuki Murata of Osaka University, for his valuable comments, insights, and continuous encouragement, not just for my research but my future. My gratitude for his support is not expressible in word. I am proud of being under his supervision.

I greatly acknowledge Associate Professor Masashi Sugano of Osaka Prefecture University with deepest appreciation. He has always and hospitably given me a number of appropriate advices and feedback all the time until now. I was helped number of times by his support. Without his support, all of my works are not achieved.

I would like to express my sincere appreciation to Professors Koso Murakami, Makoto Imase, Teruo Higashino, and Hirotaka Nakano of Osaka University, for their technical guidance and cooperation.

I would like to particularly express my gratitude to Associate Professor Naoki Wakamiya, Specially Appointed Associate Professor Kenji Leibnitz, and Research Associate Shin'ichi Arakawa. Their helpful and invaluable comments and suggestions improve this thesis.

I also wish to thank Associate Professor Go Hasegawa, Assistant Professor Masahiro Sasabe, and Assistant Professor Yuichi Oshita. Their help greatly contributed to my research.

I owe a lot to my friends, colleagues, and Advanced Network Architecture Research Group of Osaka University. Our conversations and discussions greatly helped me to advance my research.

Finally, I express my thanks to my family for their constant encouragement during my undergraduate studies.

References

[1] C. Gershenson and F. Heylighen, "When can we call a system self-organizing?," in *Proceedings of the 7th European Conference on Advances in Artificial Life (ECAL 2003)*, pp. 604–614, Sept. 2003.

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

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

[4] 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*, vol. 43, pp. 78–85, July 2005.

[5] F. Dressler, *Self-Organization in Sensor and Actor Networks*. John Wiley & Sons, Nov. 2007.

[6] L. Gan, J. Liu, and X. Jin, "Agent-based, energy efficient routing in sensor networks," in *Proceedings of the 3rd International Joint Conference on Autonomous Agents and Multiagent Systems*, pp. 472–479, Aug. 2004.

[7] P. Boonma, P. Champrasert, and J. Suzuki, "BiSNET: A biologically-inspired architecture for wireless sensor networks," in *Proceedings of the 2nd IEEE International Conference on Autonomic and Autonomous Systems*, July 2006.

[8] 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*, vol. 16, pp. 443–455, Oct. 2005.

[9] Y. Zhang, L. D. Kuhn, and M. P. Fromherz, "Improvements on ant routing for sensor networks," in *Proceedings of the 4th International Workshop on Ant Colony Optimization and Swarm Intelligence*, pp. 154–165, Sept. 2004.

[10] A. Kamik and A. Kumar, "Distributed optimal self-organization in ad hoc wireless sensor networks," *IEEE/ACM Transactions on Networking*, vol. 15, pp. 1035–1045, Oct. 2007.

[11] H. Chan and A. Perrig, "ACE: An emergent algorithm for highly uniform cluster formation," in *Proceedings of the 1st European Workshop on Wireless Sensor Networks*, pp. 154–171, Jan. 2004.

[12] 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*, vol. 29, pp. 143–154, July 2005.

[13] D. Zaharie and F. Zamfirache, "Dealing with noise in ant-based clustering," in *Proceedings of the IEEE Congress of Evolutionary Computation*, pp. 2395–2401, Sept. 2005.

[14] 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 the 19th National Conference on Artificial Intelligence*, pp. 28–33, July 2004.

[15] A. Avižienis, J.-C. Laprie, B. Randell, and C. Landwehr, "Basic concepts and taxonomy of dependable and secure computing," *IEEE Transactions on Dependable and Secure Computing*, vol. 1, pp. 11–33, January–March 2004.

[16] B. Lussier, R. Chatila, F. Ingrand, M.-O. Killijian, and D. Powell, "On fault tolerance and robustness in autonomous systems," in *Proceedings of the 3rd IARP/IEEE-RAS Joint Workshop on Technical Challenge for Dependable Robots in Human Environments*, Sept. 2004.

[17] H. Kitano, "Biological robustness," *Nature Review Genetics*, vol. 5, pp. 826–837, Nov. 2004.

[18] H. Kitano, "Towards a theory of biological robustness," *Molecular Systems Biology*, vol. 3, Sept. 2007.

[19] E. I. Oyman and C. Ersoy, "Multiple sink network design problem in large scale wireless sensor networks," in *Proceedings of the International Conference on Communications*, pp. 3663–3667, June 2004.

[20] Y. Chen, E. Chan, and S. Han, "Energy efficient multipath routing in large scale sensor networks with multiple sink nodes," in *Proceedings of the 6th International Workshop on Advanced Parallel Processing Technologies*, pp. 390–399, Oct. 2005.

[21] M. Kalantari and M. Shayman, "Design optimization of multi-sink sensor networks by analogy to electrostatic theory," in *Proceedings of the Wireless Communication and Networking Conference*, pp. 431–438, Apr. 2006.

[22] H. Lee, A. Klappenecker, K. Lee, and L. Lin, "Energy efficient data management for wireless sensor networks with data sink failure," in *Proceedings of the Workshop on Resource Provisioning and Management in Sensor Networks*, Nov. 2005.

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

[24] 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*, vol. 2977, pp. 90–104, May 2004.

[25] M. Dorigo, V. Maniezzo, and A. Colorni, "The Ant System: Optimization by a colony of cooperating agents," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 26, pp. 29–41, Feb. 1996.

[26] M. Younis, M. Youssef, and K. Arisha, "Energy-aware routing in cluster-based sensor networks," in *Proceedings of the 10th IEEE International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunications Systems*, pp. 129–136, Oct. 2002.

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

[28] IEEE, *802.15.4–2003 IEEE Standard for Information Technology–Part 15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low–Rate Wireless Personal Area Networks (LR-WPANs)*, Oct. 2003.

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

[30] M. Steinder and A. S. Sethi, "A survey of fault localization techniques in computer networks," *Science of Computer Programming, Special Edition on Topics in System Administration*, vol. 53, pp. 165–194, Nov. 2004.

[31] Y. Moreno, M. Nekovee, and A. F. Pacheco, "Dynamics of rumor spreading in complex networks," *Physical Review E*, vol. 69, June 2004.

[32] D. H. Zanette, “Dynamics of rumor propagation on small-world networks,” *Physical Review E*, vol. 65, Mar. 2002.

[33] D. H. Zanette, “Critical behavior of propagation on small-world networks,” *Physical Review E*, vol. 64, pp. 901–905, Oct. 2001.

[34] J. M. Carlson and H. Doyle, “Highly optimized tolerance: A mechanism for power laws in designed systems,” *Physical Review E*, vol. 60, pp. 1412–1427, Aug. 1999.