Self-Organized Data-Gathering Scheme for Multi-Sink Sensor Networks Inspired by Swarm Intelligence

Yuichi Kiri†, Masashi Sugano‡ and Masayuki Murata†

† *Graduate School of Information Science and Technology, Osaka University*
1–5 Yamadaoka, Suita, Osaka 565–0871, Japan
{y-kiri,murata}@ist.osaka-u.ac.jp

‡ *School of Comprehensive Rehabilitation, Osaka Prefecture University,*
3–7–30 Habikino, Habikino, Osaka 583–8555, Japan
sugano@rehab.osakafu-u.ac.jp

Abstract

We propose a new system of gathering data from sensor networks with multi-sink configurations inspired by the swarm intelligence of ants. It is a novel solution developed to achieve reliable data gathering over extended periods of time. Each sensor node determines its next action through repeated interaction and feedback from its neighbors. Beneficial clustering and routing emerge in a self-organized way from these actions, and this leads to a robust and dependable data-gathering system. Our simulation results revealed that the proposed system can reliably deliver event information to the sink nodes, is robust over very-poor-quality wireless channels, and has self-recovery capabilities to deal with failures of sensor nodes including the sink nodes.

1. Introduction

Sensor networks have become one of the hottest areas of research and a lot of research has focused on this topic. The nodes in these networks sense their ambient surroundings or events, and transmit the information they obtain via wireless channels to sink nodes, which gather the data from all sensor nodes. Because sensor networks can cover large areas and extract localized features, they have been attracting increased attention from a wide variety of areas such as medical fields, welfare, security, and disaster prevention.

It is difficult to achieve longer and more reliable monitoring of regions. The sensor nodes' memory capacity, processing power, and countermeasures against error in wireless channels inevitably need to be sacrificed for economic efficiency where their production cost should be lower than 1$ to attain affordable sensor networks [2]. Sensor nodes in some applications being considered today are outdoors and in environments that vary a great deal according to the location. Therefore, we need to take poor quality in communication channels and even sink node failure into account according to the circumstances. Although a great deal of conventional research has emphasized that battery power is one of the most important resources in sensor networks, robustness and dependability should be valued more for the environments.

Sensor networks also suffer from scalability problems. Although the number of sensor nodes deployed within the region being monitored depends on a network's applications, the total number can reach hundreds or thousands. The common method of communication used by sensor networks is converge-cast (many-to-one) in which all packets transmitted from sensor nodes concentrate on a sink node, because the purpose of the network is usually to collect information. Thus, as the number of sensor nodes increases, those nodes located near the sink node consume large amounts of power. The sink node is then isolated once they deplete their batteries, and the sensor network fails to function.

Considering that an enormous number of sensor nodes can be deployed, it is widely said that concentrated controlling systems do not fit well to sensor networks. The difficulty to obtain information over the entire network is caused by the number of sensor nodes. From that standpoint, autonomous distributed systems hold great promise. In such systems, sensor nodes must determine their next actions based on limited information. Each of their actions must lead to the most beneficial action for the network as a whole without external management. Therefore, self-organized behavior is strongly desired for sensor networks.

We propose a novel solution to these problems with sensor networks detecting events in this paper. Our scheme does not require geographical information or synchronization between sensor nodes. The most significant characteristic of our scheme is the application of the swarm intelligence of ants to sensor networks with multi-sink configurations. Each ant determines what it should do through estimating its surroundings, interacting with neighboring ants, and obeying the local rules it has as a species. Although an individual ant is simple and unintelligent, their collective action creates biological order and results in them heading toward a beneficial direction on the system level. This feature of ant swarming is highly concurrent, adaptable to changes in the environment, and has been attracting a great deal of attention. The model of swarming is well suited to sensor networks composed of nodes with only simple functional capabilities.

Our scheme, which adopts swarm intelligence, achieves long-term data gathering in a different way to much of the conventional research, i.e., it is robust and adaptable to environmental changes. Most of the past research has only focused on reducing the imbalances in power consumption over the region being monitored or power consumption itself. Of course as we previously mentioned, although power is one of the most valuable resources of battery-operated sensor nodes, we urgently require robustness against poor quality in communication channels, sensor node failure, and even sink node failure. We resolved such critical demands using swarm intelligence such as *ant-based clustering* and *Ant Colony Optimization* (ACO). We show that our new scheme enables high degrees of data gathering and attains self-recovery capabilities and robustness through the simple actions of individual sensor nodes.

The remainder of the paper is organized as follows. Section 2 presents related work that has taken multi-sink sensor networks and swarm intelligence into consideration. The details on our proposed scheme and its operation are described in Section 3 and the simulation results are provided in Section 4. The paper concludes with an outlook toward our future research.

2. Related work

Our proposed scheme includes many concepts within its design. Multi-sink sensor networks, which are the target of this paper, have been actively discussed lately. Swarm intelligence has also drawn a great deal of interest from a variety of areas, other than networking, due to its self-organized behavior.

2.1. Sensor networks with multiple sink nodes

Although much research has set its target on sensor networks with a single-sink configuration, a not negligible amount of work has addressed multi-sink sensor networks. The motivation for studying multiple-sink nodes has been to improve scalability. For example, Oyman and Ersoy [12] stated that in a network with thousands of sensor nodes, they should be grouped into several clusters and sink nodes should be individually dedicated to a single cluster in view of scalability. They solved problems by finding the best sink locations on that basis and determining the minimum number of sinks for a predefined number of operation periods.

There has been some research on the routing problem in multi-sink sensor networks. Chen et al. [5] deployed multi-sink nodes taking into account that sensor nodes around a sink node consume a large amount of power in relaying the packets destined for that sink node. Multi-path Routing in large-scale sensor networks with Multiple Sink nodes (MRMS) has been proposed on that basis. MRMS takes a multi-path routing approach, in which sensor nodes shift a primary path to an alternative one when the residual power of the sensor nodes on the primary path falls below a predefined threshold. Kalantari and Shayman [8] analytically derived the optimal-routing direction for an arbitrary point over a region being monitored for both single-sink and multi-sink configurations based on electrostatic theory. Using their results, we can equalize the power consumption of sensor nodes as much as possible. The results are represented in the form of partial differential equations; however, whether the equations can be solved depends on the sensor-network model.

The proposal by Lee *et al.* [10] is close to our idea in terms of applying the multi-sink configuration to improve robustness against sink-node failure as well as scalability. They established the same number of routing trees as sink nodes, with the sink nodes acting as roots using the concept of *De Bruijn digraphs*. When there is sink-node failure, the sensor nodes in the failed sink-node's tree can move into another tree, so their scheme was robust against sink failure. However, they did not evaluate how robust it was or clarify how this mechanism would affect sensor networks.

2.2. Swarm intelligence

One of the most frequently used forms of swarm intelligence in the networking area is probably ACO, which is observed in ants foraging for food. Although the ACO approach will be described in detail in Section 3.3, we also utilized ACO's idea in our routing along with some trailblazers. For example, AntNet [3] applied ACO to packet-switching networks such as the Internet, and AntHocNet [4] applied it to Mobile Ad-hoc Network (MANET). Although differing in the details, BeeAdHoc [14] was also proposed for MANET, which also provided positive feedback inspired by the swarm intelligence of honey bees. Zhang *et al.* [16] found that AntNet does not perform sufficiently well in sensor networks and, therefore, improved it. However, their improved version only considered single-sink sensor networks as the target and they did not discuss robustness.

Some species in the world of ants divide their eggs and larvae based on their size. A clustering algorithm called ant-based clustering was studied [15, 13] inspired by such behavior. Ant-based clustering is characterized by a probabilistic approach, where clustering is repeatedly realized by ants and stochastically selected eggs are picked up or dropped. We introduced such a probabilistic approach to the clustering of sensor nodes in our scheme. Dynamic clustering adapting to changes in the environment can accommodate sink-node failures.

3. Design of proposed scheme

3.1. Design consideration

Scalability is critical in sensor networks incorporating a number of sensor nodes. However, converge-cast constrains scalability in sensor networks.

The transmission range by sensor nodes needs to be short to reduce the amount of power consumed by their limited batteries. Therefore, multi-hop communication is necessary for sensor networks. All the packets transmitted by sensor nodes, on the other hand, are destined for sink nodes to enable information to be gathered. The sensor nodes near a sink node must relay all packets and, therefore, run out of power earlier than the others. As a result, the network is partitioned and packets cannot reach the sink node. The area around the sink node is a bottleneck for sensor networks. We previously pointed out this problem in previous work [9]. A wide variety of energy-aware routing protocols has been proposed thus far to resolve these imbalances in power consumption. However, these approaches are not essential solutions as long as converge-cast is used in a multi-hop way. We believe that we must adopt a multi-sink configuration to improve the scalability of sensor networks.

The benefits that can be gained from a multi-sink configuration are not limited to scalability. Conventional research has actively discussed sensor node failure, but only a few have discussed sink node failure. It is necessary to continue data collection if a sink node fails, especially in sensor networks deployed in areas people cannot access. If we could prevent a sink node from becoming a single point of failure by applying a multi-sink configuration to a sensor network, and if we could also maintain performance the same as it was before the sink node failed, we can say it is a self-recovering sensor network. Such a system would greatly contribute to dependability and robustness.

Our sensor network model is outlined in Fig. 1. A number of sensor nodes are placed over the region being monitored as well as multiple sink nodes. These sink nodes behave as gateways to other networks such as the Internet and dedicated networks. Packets received at the sink nodes are processed and transmitted to data servers. We only focus on

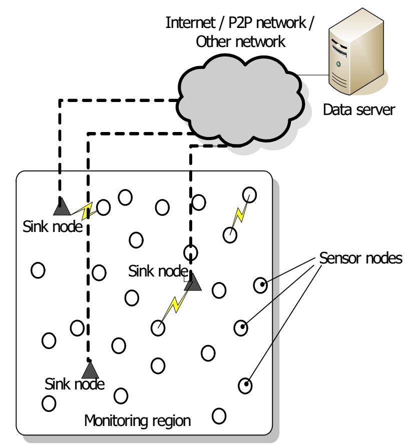


Figure 1. Network architecture model

communications from sensor nodes to sink nodes, and other communications and data processing are beyond the scope of this paper.

3.2. Overview of proposed scheme

We propose dividing a sensor network into as many clusters as there are sink nodes, and sensor nodes that transmit packets to sink nodes are dedicated to that cluster. In terms of clustering, one option is to optimally split sensor nodes into clusters and retain them for the lifetime of the sensor networks. Data collection over an area may be impossible due to cluster-level power depletion where a number of events, which the sink node should be notified of, only occurs in the area of a particular cluster. If a sink node fails, its cluster cannot collect information, either. Clusters in our proposed scheme dynamically expand and shrink by applying ant-based clustering depending on the situation. Such behaviors by clusters adapt to the varieties of patterns for occurring events and contribute to longer operating times. Even when a sink node fails, dynamic clustering enables sensor networks to work well with the failure having only a slight influence.

We apply the principle of ACO to hop-by-hop routing in our proposed scheme. Each sensor node has a *pheromone table*, and the benefit of selecting a neighbor as next-hop node are stored in the form of pheromones. When a sensor node transmits a packet to notify the sink node of event, it refers to its pheromone table, and stochastically selects the next-hop node based on the pheromone value. Thus, each sensor node selects a next-hop node with greater probability of having a higher pheromone level. A sensor node with more pheromone value means preferable next-hop node. Furthermore, if some neighboring nodes have almost the same 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. This leads to load sharing.

3.3. Details on ant routing in our scheme

The basic concept underlying ACO in the routing algorithm is outlined in Fig. 2. A source node transmits small control packets called "ants" to the destination node to construct paths. In sensor networks, destination node corresponds to sink node. Each ant wanders randomly over the network to find its destination, collecting information on the quality of the trail it walked through (e.g., hop count, delay, and the residual power of nodes on the trail). Once the ant reaches its destination, it follows the trail back to the source node, leaving some pheromone along the trail. The amount of pheromone it leaves depends on the quality of the path. Ants that are transmitted later from the source node stochastically select a path on which more pheromones remain, and more effectively arrive at the destination following better paths. These ants also leave pheromones on their return journey. Application packets are likely to follow a beneficial path due to such a positive-feedback approach.

Unlike the Internet or MANET that mainly use unicast communication, a "good path" in the sensor network is:

- A path leading to a sink node without any unnecessary circumvention and
- One whose sensor nodes have sufficient residual power.

It is not necessary for each source node to find good paths to a sink node by flooding them with ants. Such flooding beginning from the source node could cause unnecessary power consumption and needlessly occupy wireless channels, because of ants traveling back and forth over the network. Thus, we choose sink nodes to flood the ants, which we called *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 node, which has enough residual power. With that in mind, the role of backward ants is to establish a pheromone distribution in which the required next-hop node has a higher pheromone value.

Let us introduce the following terms to simplify our explanation of routing.

n_i: ID of sensor node.
S_k: ID of sink node. At the same time, S_k also represents the ID of a cluster to which sink node S_k is dedicated.
$Pb_{n_i}(S_k)$: Pheromone value that n_i assigns to a backward ant, which is transmitted by S_k.
S_{n_i}: ID of sink node that n_i belongs to.

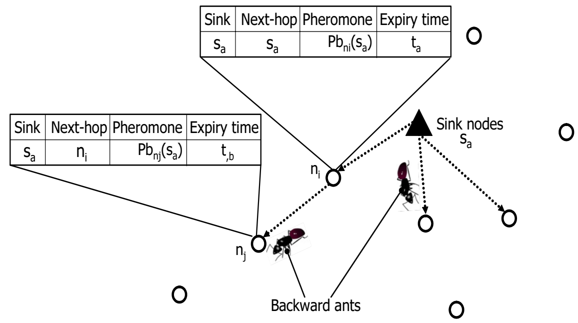


Figure 3. Updating pheromone table on receiving backward ants.

$P_{n_i}(n_i)$ Pheromone value for n_i which is declared by n_i.
$P_{n_i}(n_j, S_k)$: Pheromone value that represents benefits of using n_j as next-hop node for n_i to transmit packet whose destination is S_k.
$C_{n_i}(S_k)$: Cluster pheromone of S_k, which means benefits of belonging to the cluster S_k, estimated by n_i.
E_{r_i}: Residual power of n_i.
E_{m_i}: Initial power of n_i.

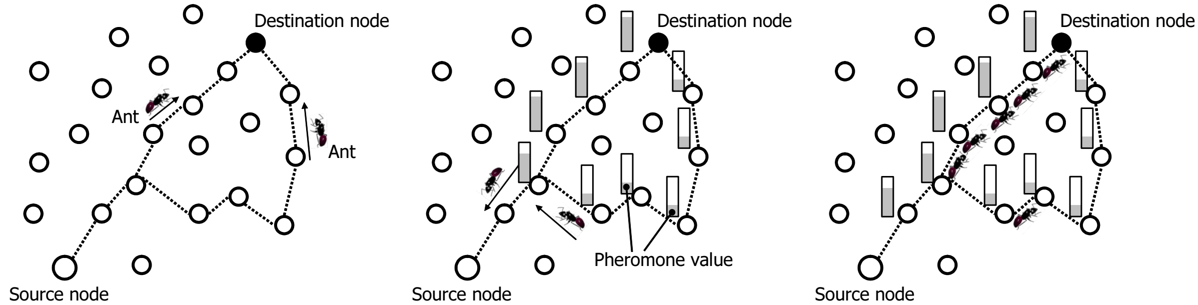
S_a broadcasts backward ant B with maximum pheromone value $Pb_{S_a}(S_a) = P_{\max}$. On receiving B, sensor node n_i stores its pheromone value ($Pb_{S_a}(S_a)$), its source node (S_a), and sensor node which relays B immediately before (S_a) in its own pheromone table as an entry. Thus, n_i memorizes 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 pheromone value. This new pheromone value $Pb_{n_i}(S_a)$ is calculated according to:

$$Pb_{n_i}(S_a) = \alpha \left(1 - \exp\left(-\beta \frac{E_{r_i}}{E_{m_i}}\right)\right) Pb_{S_a}(S_a) \quad (1)$$

$$0 < \alpha < 1, \beta > 0$$

After receiving B, which is relayed by n_i, n_j creates a new entry for it, as in the case of n_i. Then, n_j calculates a new pheromone value according to Eq. (1), and forwards B with a new pheromone again. An ideal pheromone distribution emerges through frequent repetitions of these behaviors.

Although the entities of backward ants are merely small control packets, flooding them consumes vast amounts of power in relaying and occupies wireless channels. Thus, the frequency with which backward ants are transmitted should be as low as possible. However, the pheromone table must reflect current changes to the network to construct an efficient path to sinks where changes to the network topology occur during the intervals backward ants are



(a) Source node transmits ants to its destination.

(b) Once an ant arrives at destination, it returns to the source node, leaving pheromones on its return path.

(c) More ants and application packets are attracted to a path with more pheromones.

Figure 2. Principle underlying the routing algorithm using ACO.

transmitted. We decided to assign the role of updating the table to *hello ants*, which are broadcast packets transmitted more frequently at fixed intervals than backward ants. They do not cause flooding with a TTL of 1. Hello ants carry pheromone value P_{n_i} of source node n_i, the ID of sink node S_{n_i} to which n_i belongs, and the pheromones of cluster $C_{n_i}(S_{n_i})$ to which S_{n_i} is dedicated. We will describe cluster pheromones in detail in Section 3.4. The exchange of hello ants by neighboring nodes reflects an up-to-date state in the pheromone table of each sensor node. On receiving a hello ant from n_i, and only if it has an entry for n_i stored in its pheromone table, n_j updates the pheromone value of its entry as follows

$$P_{n_j}(n_i, S_{n_i}) = \gamma P_{n_j}(n_i, S_{n_i}) + (1-\gamma)P_{n_i}(n_i) \tag{2}$$

with $\gamma \in [0, 1]$. When n_j broadcasts a hello ant, n_j makes it carry its pheromone value, P_{n_j}, which is defined as follows.

$$P_{n_j} = \frac{\sum_l \sum_m P_{n_j}(l, m)}{l \cdot m} \tag{3}$$

This is the average pheromone value for entries in its pheromone table.

Sending event information to sink nodes is not a difficult task if pheromone distribution is properly constructed. When sensor node n_i transmits a packet, it sets the destination of the packet to the sink node S_{n_i} to which it belongs, and stochastically selects the next-hop node by referring to pheromones in its pheromone table. The probability of n_i selecting n_j as a next-hop node is represented as:

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

where N_{n_i} means the candidate set of next-hop nodes for n_i.

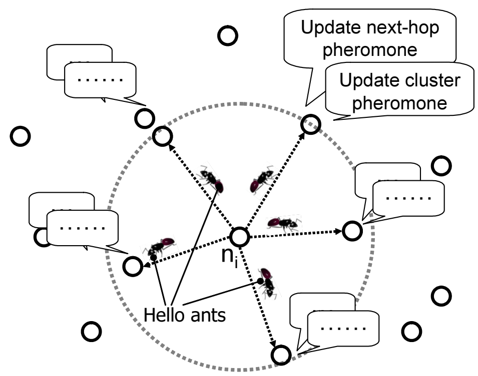


Figure 4. Sensor nodes receiving hello-ant update the pheromone values in their pheromone table.

3.4. Ant-based clustering in our system

How to select a sink node for each sensor node still remains a question in multi-sink sensor networks. One option is setting a sink for all sensor nodes when deploying a sensor network and keeping them for as long as the sensor network continues to operate. However, sensor nodes near the sink node, which is responsible for that area, may deplete their power when relaying numerous packets when events frequently occur in a specific area of the region being monitored. As a result, collecting information from that area becomes impossible. Otherwise, sensor nodes belonging to that sink node may blindly send packets to it when failure occurs. We introduced an ant-based method of clustering to solve these problems.

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 while wandering around a certain neighborhood. Clusters of different-sized larvae emerge in a self-organized way based on their behavior. We substitute similarity with the advantage of belonging to a cluster, and perform clustering to suit the network situation.

A cluster is a set of sensor nodes. Therefore, we should consider the state of sensor nodes in a cluster, such as residual power and distance from the sink node to define the advantages a cluster gains by behaving in the proposed way. These situations have already been reflected in the pheromone value introduced in Section 3.3, but this pheromone value mostly depends on neighboring sensor nodes and does not necessarily indicate the situations on a cluster level. To overcome this limitation, we introduce a *cluster pheromone*. The value of a cluster pheromone is calculated based on the pheromone of sensor node, and reflects the attractiveness of a cluster.

When sensor node n_i belongs to cluster S_j, n_i generates hello ants to carry the cluster pheromones of $C_{n_i}(S_j)$, which n_i considers, as well as cluster ID S_j. Sensor node n_k, which receives a hello ant transmitted by n_i, memorizes the n_i that says its cluster is S_j and its cluster pheromone is $C_{n_i}(S_j)$ in its *neighbor table* (This is currently a separate table from the pheromone table, but these are easily integrated into one). $C_{n_i}(S_j)$ is defined as:

$$C_{n_i}(S_j) = \frac{\sum_{k \in belong_{n_i}(S_j)} C_k(S_j) + avg\_ph_{n_i}(S_j)}{|belong_{n_i}(S_j)| + 1} \quad (5)$$

where $belong_{n_i}(S_j)$ represents a set of nodes which are neighbors of n_i that participate in cluster S_j. $avg\_ph_{n_i}(S_j)$ is represented in Eq. (6) and is the average amount of pheromones in the entries which has the destination sink S_j.

$$avg\_ph_{n_i}(S_j) = \frac{\sum_{k \in belong_{n_i}(S_j)} P_{n_i}(k, S_j)}{|belong_{n_i}(S_j)|} \quad (6)$$

Eq. (5) merely calculates the average value for the cluster pheromones of S_j, which is given by n_i and neighbors of n_i belonging to S_j.

Because hello ants also convey the cluster ID of the source node, sensor nodes can learn which cluster the neighboring nodes belong to, and whether they are on the boundaries of clusters. Sensor nodes on boundaries stochastically select whether they will transfer their membership from their cluster to another, just like real ants that pick up larvae to carry them elsewhere. Actual ants determine whether to pick up larvae based on similarities between these and neighboring larvae. We uses this feature of belonging to a cluster instead of similarity. The probability of n_i changing membership from cluster S_j to cluster S_k is defined in Eq. (7)

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

where k is a threshold parameter which controls probability of changing cluster membership. $f_{n_i}(S_j, S_k)$ is calculated as

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

In this equation, we consider the ratio between the number of neighbors participating in a certain cluster and the number of all neighbors. This term reflects our idea that sensor nodes on cluster boundaries should have a higher tendency to change their membership than others in view of connectivity among sensor nodes inside clusters. The cluster gradually changes according to the probability $P_{n_i}(S_j \to S_k)$.

3.5. Detection of sensor-node and sink-node 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. Sink nodes 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 long term.

We applied a soft-state model to detect failures using hello ants transmitted by all sensor nodes at fixed intervals. As seen in Fig. 3, each entry in the pheromone table has a field for expiry time. At each time a sensor node receives a hello ant from n_i, it updates the expiry time field in the entry whose next-hop node field is n_i in its pheromone table. If n_i fails or experiences depleted power, neighboring sensor nodes cannot receive hello ants from n_i. After expiry time δ_n, neighboring sensor nodes delete the entry for n_i. Thus, as we use a soft-state model, we do not need to take explicit measures to detect failure. Sensor nodes select appropriate next-hop nodes according to Eq. (4) without any special handling.

Detecting sink node failure is also based on the same soft-state model. That is, the sink node periodically broadcast hello ants as well as sensor nodes. Sensor nodes around the sink node determine that the sink has failed if they have not received hello ants from it for a predefined time, δ_S. The cluster is no longer preferable in sink node failure. Thus, sensor nodes detecting sink node failures only stored a small value for the cluster pheromones in their hello ants. As these hello ants propagate over the network, sensor nodes participating in the failed sink's cluster abandon their membership, and join other clusters.

Detecting failure in sink nodes differs vastly from that in sensor nodes in that false detection is unacceptable. If hello ants cannot reach a neighboring node due to packet loss, the neighboring node mistakenly detects failure. Detecting failure incorrectly at a sensor node has an insignificant local effect; however, detecting failure in a sink node causes a transition in membership on the cluster level. We therefore decided to set δ_S for a longer duration than δ_n. The longer the expiry interval is, the lower the probability of incorrect detection is. A shorter expiry interval is preferable to enable adaptation to rapidly changing situations. Balancing this tradeoff is an important challenge that remains for the future.

4. Evaluating our proposed scheme

4.1. Simulation environment

We implemented our proposed scheme for the ns–2 [1] network simulator. In all the following experiments, we randomly place 200 sensor nodes over a region monitoring a square, 100 m per side. We assume that there are four sink nodes, and prepare three scenarios for a series of coordinates for the sinks as follows.

- *Scenario 1*: $(25, 25)$, $(75, 25)$, $(25, 75)$, $(75, 75)$
- *Scenario 2*: $(15, 75)$, $(38, 20)$, $(60, 85)$, $(85, 50)$
- *Scenario 3*: $(10, 10)$, $(90, 10)$, $(10, 90)$, $(90, 90)$

Each plot below was made from the average of 15 simulations. We used IEEE 802.15.4 [7] in the simulation experiments, which is well known for its energy efficiency and low-rate communication. We applied a two-ray ground reflection the radio propagation model to simulate the signal power received for each packet. The parameter settings of the sensor nodes from [11, 6] were used in Table 1. Further simulation parameters are also shown in Table 2. We are currently investigating the individual influences of these parameters, but the values in the table were chosen because they yielded a good performance.

Our target was detecting event in sensor networks. Sensor nodes detect various kinds of events such as increase in temperature, and each node generates a packet to notify its sink of these events. These packets are relayed by other sensor nodes and arrive at the sinks. Events in our simulation experiments occurred at random points per second.

4.2. Event notification rate

Reliability of communication is one of the most important metrics in sensor networks whose purpose is to collect event data. We, therefore, define a metric to reflect that idea, the event-notification rate. The event-notification rate is defined as r_e/s_e, where s_e is the number of packets generated by sensor nodes which detect event e, and r_e is the number of packets carrying information on e and arriving at sink nodes. The plots in Fig. 5 are the results obtained with 200 sensor nodes in all three scenarios. PER is the packet error rate. The packet is randomly dropped depending on value of the PER. As we can see from these figures, the proposed scheme achieves a high event-notification rate of about 90%. The sensor network can carry event information to the sink nodes without the harmful effects of notification errors even when the wireless channels are of very poor quality. This is because the control packets are broadcast packets and interaction between neighboring sensor nodes compensates for the dropped packets. The pheromone distribution over the region being monitored using *Scenario 1* at $t = 400$ s with PER $= 0$ and PER $= 0.3$ is plotted in Fig. 6. We can see that the pheromone distribution is almost the same regardless of PER or the quality of the wireless channels. This is the advantage of the self-organized scheme, which solves difficult problems through iterating the simple actions of agents.

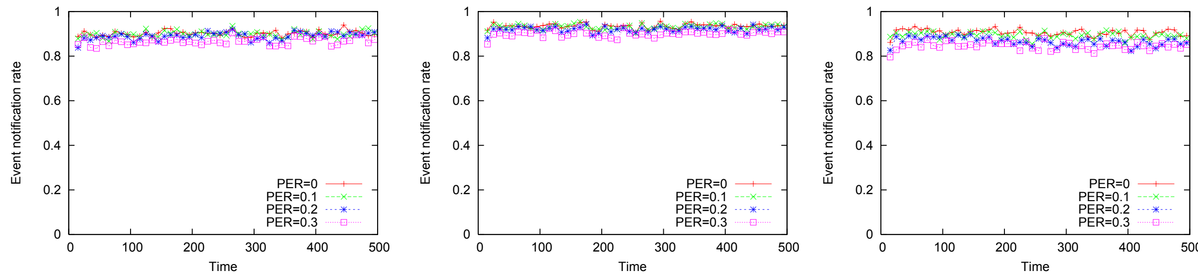
Fig. 7 plots the event-notification rate for Scenario 1 with 400 sensor nodes and looks about the same as in Fig. 5(a). The reason the event-notification rate is slightly reduced is due to the interference between packets carrying event information and backward ants, which increase in number while propagating in the network. Controlling the number of backward ants remains a serious challenge for future work.

Table 1. Parameters for sensor nodes

Power consumption for transmission	40.95 mW
Power consumption for reception	45.78 mW
Transmission power	0 dB
Transmission distance	10 m
Interference distance for transmission	16.8 m
Frequency	2450 MHz
Radius of sensing area	10 m
Transmission rate	250 kbps

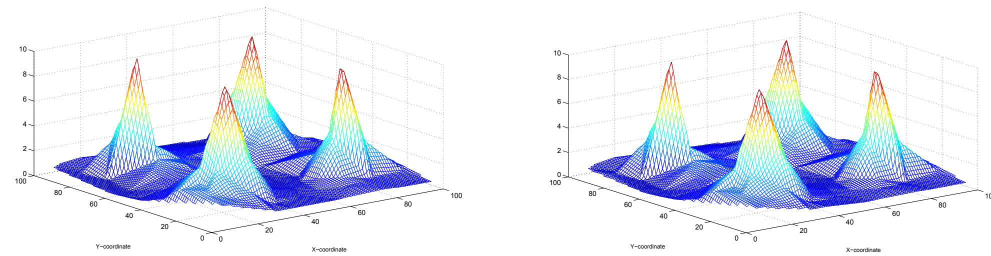
Table 2. Simulation Parameters

Initial power	1 J
α	0.7
β	7
δ_n	3 s
δ_S	9 s
Interval in transmitting hello ants	0.5 s
Interval in transmitting backward ants	10 s



(a) Scenario 1 (b) Scenario 2 (c) Scenario 3

Figure 5. Event notification rate over time for all three scenarios



(a) PER=0 (b) PER=0.3

Figure 6. Pheromone distribution over region being monitored at $t = 400$ s for Scenario 1.

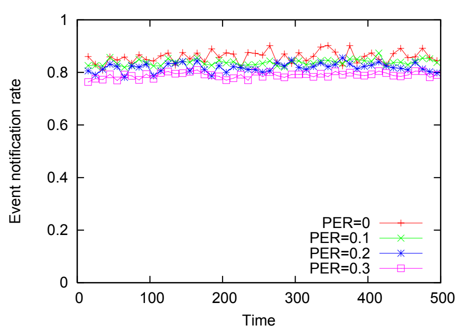


Figure 7. Event-notification rate with 400 nodes for Scenario 1.

4.3. Self-recovery from node failure

We simulated sensor node failure as follows to study the robustness of our proposal. Failure in this paper is defined as a state in which neither transmission nor reception is possible. We randomly selected $200 \times p_{\text{fail}}$ ($p_{\text{fail}} \in [0, 1]$) sensor nodes and made them fail at $t = 100$ s simultaneously in sensor networks where 200 sensor nodes had been deployed. This is far more serious than what can actually happen in reality. We set the PER to 0.1 and consider Scenario 1. We are interested in observing how soon and to what extent the event-notification rate would immediately recover from a decline after $t = 100$ s.

The results we obtained are shown in Fig. 8. The event-notification rate hovers at about 90% until $t = 100$ s, and then experiences a sudden drop just after $t = 100$ s. However, the event notification rate recovers within the short time to its previous state. That is, a certain degree of sensor node failures has little influence on the event-notification rate in our scheme, and it continues to collect event information. Most sensor nodes have some candidates for the next-hop node in their pheromone table and they stochastically select the next-hop node from these due to our ACO approach. When a sensor node detects failure in a neighboring node, it only eliminates the entry for the failed node, and the event-detection rate quickly recovers. These actions do not require global information or coordination, but the current state of neighbors which is easily acquired. With

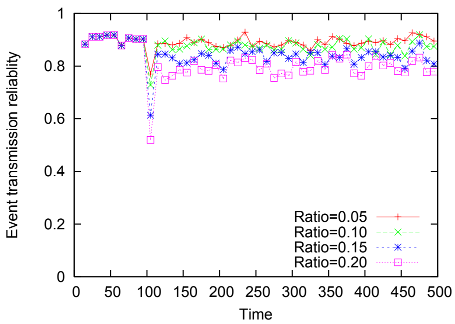


Figure 8. Self-recovery feature when sensor nodes fail simultaneously.

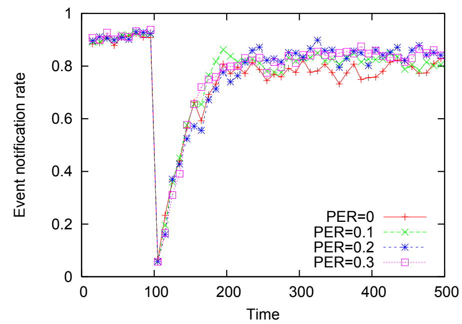


Figure 9. Self-repair features when sink node fails.

efficient combination of node-failure detection using a soft-state model and an ACO approach which selects the next-hop node stochastically, we can achieve self-recovery capabilities and it largely contributes to the attainment of more reliable sensor networks.

4.4. Self-Recovery after Sink Failure

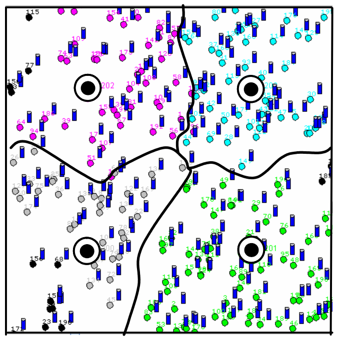
Sink nodes can obviously fail as well. Most studies with single-sink configurations do not accommodate sink node failures, and in addition, only a few countermeasures against such failures have been mentioned even in multi-sink configuration sensor networks thus far. We took sink node failures into account, and proposed that sensor nodes belonging to the failed sink node transfer their membership to another cluster and continue to collect event information. We conducted new simulation experiments as follows to demonstrate the effectiveness of this proposal. Using Scenario 1, we made sink node S_1 located at the point of $(25, 25)$ fail at $t = 100$ s with PER $= 0.1$. However, it was difficult to interpret the effectiveness when events occur far from S_1. Therefore, we limited the area where events occurred to a circular area with a radius 20 m at the center of S_1 in our simulations.

Fig. 9 shows the experimental results. At $t = 100$ s, the event-notification rate falls steeply to almost 0% because all the packets are destined for S_1. However, after that, it recovers rapidly to the same rate as before S_1 failed. The state of the clusters is shown in Fig. 10. The single circles represent sensor nodes and the double circles are sink nodes. The gauge located to the upper right of a sensor node shows its residual power of the sensor node. Before S_1 failed, four clusters emerged, for each sink node. After S_1 failed, the clusters converge to three. We can achieve about the same event-notification rate as that before sink failure, through all sensor nodes repeating interactions with their neighbors and changing the shapes of clusters in a self-organized way. This strongly indicates our proposal is extremely robust and has strong self-recovery capabilities.

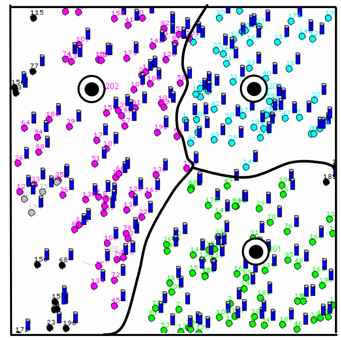
5. Conclusion

We described sensor networks with multi-sink configurations in this paper that were preferable in terms of scalability and fault tolerance and proposed a novel data-gathering scheme suited to these types of networks. It was inspired by the swarm intelligence of ants such as that found in ant-based clustering and ant-colony optimization. Each sensor node determines its next action through repeating interactions with neighboring nodes, and these actions enable the sensor network to evolve toward an optimal state. This data-gathering scheme, which splits sensor nodes into various clusters using ant-based clustering and conveys packets to the sink nodes with ACO routing, had a high event-notification rate and was stable and robust against poor-quality channels. It also had good self-recovery capabilities. In the case that sensor or sink node failure occurs, our scheme detects it properly, recovers from it, and can continue data collection in the way same as it was before the failure. Especially, as far as we know, there are no papers that deal with any countermeasures against sink node failure and evaluate it.

Our proposed data-gathering scheme, which is a combination of self-organization and self-recovery, is preferable for sensor networks because it can be deployed under harsh conditions. We did notice, however, that backward ants frequently interfered with packets conveying event information, and robustness gradually decreased with longer intervals between transmissions between backward ants. So far we have not fully examined the influences of the parameters. We adopt a probabilistic approach on clustering, but a deterministic approach might show better properties. We intend to tackle these issues, and would like to clarify the differences in characteristics between self-organized and centrally controlled sensor networks.



(a) $t = 99$



(b) $t = 120$

Figure 10. Cluster shapes over time when sink node fails

Acknowledgements

This research was supported in part by "Special Coordination Funds for Promoting Science and Technology: Yuragi Project" of the Ministry of Education, Culture, Sports, Science and Technology, Japan, Grant-in-Aid for Scientific Research (A) 16200003 and (C) 19500060. We also would like to especially thank Kenji Leibnitz and Shin'ichi Arakawa of Osaka University for their constructive comments throughout this work.

References

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

[2] I. F. Akylidiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci. A survey on sensor networks. *IEEE Communications Magazine*, Aug. 2002.

[3] 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.

[4] 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, Oct. 2005.

[5] Y. Chen, E. Chan, and S. Han. Energy efficient multipath routing in large scale sensor networks with multiple sink nodes. In *Proceedings of the Sixth International Workshop on Advanced Parallel Processing Technologies*, Hong Kong, China, Oct. 2005.

[6] Chipcon. *Chipcon AS SmartRF®CC2420 Preliminary Datasheet*, 1.2 edition, June 2004.

[7] 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.

[8] 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*, Las Vegas, Nevada, USA, Apr. 2006.

[9] Y. Kiri, M. Sugano, and M. Murata. On characteristics of multi-hop communication in large-scale clustered sensor networks. *IEICE Transactions on Communications*, E90–B(2):228–238, Feb. 2007.

[10] 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 (RPMSN 2005)*, Nov. 2005.

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

[12] E. I. Oyman and C. Ersoy. Multiple sink network design problem in large scale wireless sensor networks. In *Proceedings of the IEEE ICC 2004*, Paris, France, June 2004.

[13] 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, 2005.

[14] H. F. Wedde, M. Farooq, T. Pannenbaecker, B. Vogel, C. Mueller, J. Meth, and R. Jeruschkat. BeeAdHoc: An energy efficient routing algorithm for mobile ad hoc networks inspired by bee behavior. In *Proceedings of the 2005 conference on Genetic and evolutionary computation*, pages 153–160, Washington DC, USA, June 2005.

[15] D. Zaharie and F. Zamfirache. Dealing with noise in ant-based clustering. In *Proceeding of IEEE Congress of Evolutionary Computation*, pages 2395–2402, Edinburgh, UK, Sept. 2005.

[16] 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*, Brussels, Belgium, Sept. 2004.