

# マルチシンク構成のセンサネットワークにおける ロバストな情報収集メカニズムの提案と評価

木利 友一<sup>†</sup> 菅野 正嗣<sup>††</sup> 村田 正幸<sup>†</sup>

<sup>†</sup> 大阪大学大学院情報科学研究科 〒 565-0871 吹田市山田丘 1-5

<sup>††</sup> 大阪府立大学大学総合リハビリテーション学部 〒 583-8555 羽曳野市はびきの 3-7-30

E-mail: <sup>†</sup>{y-kiri,murata}@ist.osaka-u.ac.jp, <sup>††</sup>sugano@rehab.osakafu-u.ac.jp

あらまし 我々は複数のシンクノードを配置するマルチシンク構成のセンサネットワークを対象に、群知能のアイデアを用いたデータ収集手法を提案する。提案手法は蟻の行動に着想を得たもので、個々のセンサノードは周辺のセンサノード、あるいは環境との相互作用を繰り返しながら、自身の次の行動を決定する。そしてその個々の行動が、長期間のデータ収集を実現するようなルーティング、クラスタリングを実現する。本稿では、この提案手法が十分な信頼性を保ちながら、劣悪な無線環境にも適応し、センサノード、シンクノードの故障にも耐えうる耐故障性、自己回復性を持つことを示す。

キーワード センサネットワーク, マルチシンク構成, 群知能, 耐故障性, 自己回復性

## Adaptive and Robust Mechanism for Data Gathering for Multi-Sink Sensor Networks

Yuichi KIRI<sup>†</sup>, Masashi SUGANO<sup>††</sup>, and Masayuki MURATA<sup>†</sup>

<sup>†</sup> Graduate School of Information Science and Technology  
1-5 Yamadaoka, Suita-shi, Osaka, 565-0871, Japan

<sup>††</sup> School of Comprehensive Rehabilitation, Osaka Prefecture University  
3-7-30 Habikino, Habikino-shi, Osaka, 583-8555 Japan

E-mail: <sup>†</sup>{y-kiri,murata}@ist.osaka-u.ac.jp, <sup>††</sup>sugano@rehab.osakafu-u.ac.jp

**Abstract** We propose a new scheme of gathering data from sensor networks with multi-sink configurations inspired by the swarm intelligence. Each sensor node determines its next action through repeated interaction and feedback from its neighbors and environments in this scheme. Advantageous clustering and routing emerge in network level from these actions. Our simulation results revealed that the proposed scheme can reliably deliver event information to the sink nodes, is robust over very-poor-quality wireless channels, and has self-recovery capability to deal with sensor-node failure and even that in sink nodes.

**Key words** sensor network, multi-sink configuration, swarm intelligence, robustness, self-recovery capability

### 1. はじめに

センサネットワークは現在、最も注目を浴びているネットワーク技術の一つである。センサネットワークを構成する各センサノードは周辺環境の情報をセンシングにより取得し、センサノード間で構築されるネットワークを介して、情報収集拠点であるシンクノードにその情報を集める。アドホックネットワークならではの敷設コストの安さや、従来は点としてしか得られなかった情報を面として捉えられることなどから、センサネットワークは医療、防犯、防災など、様々な分野において大きな期待を集めている。

一方で、センサネットワークの長寿命化を達成し、効率よく情報を収集し続けることは我々に課せられた大きな課題である。

センサネットワークに用いられるセンサノードの数は数百、数千に及ぶ場合があり、その多さゆえにセンサノードの生産コストは1\$以下に抑える必要があるとの指摘がある[1]。このような生産コストを実現するためには、センサノードのメモリ容量や処理能力、また、ノイズ等に対するハードウェア的な対策なども犠牲にならざるを得ない。センサネットワークの敷設される環境は様々であり、そこでは劣悪な通信品質、あるいはシンクノードの故障すら考えられる。そのためセンサネットワークの長寿命化のためには、様々な通信品質、あるいはセンサノード、シンクノード故障への耐性を考慮しなければならない。

スケラピリティの問題もある。一般的な単一シンク構成のセンサネットワークにおける通信は、各センサノードが送出したパケットが全てシンクノードに集中するという多対一の形態

をとる．そのため，センサノードの数が多くなるほど，シンクノード周辺に位置するセンサノードの中継負荷が大きくなる．一旦これらのセンサノードが電力枯渇を起こすと，シンクノードとセンサノード間のコネクティビティは失われ，情報収集機能は破綻してしまう．我々は [2] においてこの問題を指摘した．このような消費電力の偏りを抑えるために，残余電力をメトリックとするルーティングプロトコルも種々存在するものの，マルチホップ通信を用いる限りシンクノード周辺に中継負荷が集中するという意味で本質的な解決にはならない．スケーラビリティを向上させるためには，複数のシンクノードを配置するマルチシンク構成を取らなければならないというのが我々の考えであり，また，同種の理由によってマルチシンク構成のセンサネットワークを対象としている研究も少なくない [3-6] ．

マルチシンク構成を取ることによって得られるメリットはスケーラビリティばかりではない．どんなシステムにおいても単一故障点をなくすことは重要である．マルチシンク構成のセンサネットワークとすることでシンクノードが単一故障点になることを防ぐことができれば，耐故障性という意味でも大きな意味がある [7] 上に，シンクノードが故障してもその他のシンクノードを用いて故障前と同等の性能が確保できれば，自己修復機能を有するデータ収集機構となり，センサネットワークの信頼性は大きく向上する．

センサネットワークに配置されるセンサノードの数を考慮すると，各センサノードがネットワーク全体の情報を得るのは不可能である．したがって，各センサノードはその周辺の情報のみに基づいて自律的に自身の行動を決定し，そしてその個々の行動がセンサネットワーク全体としての適切な行動に繋がらなければならない．自然界，特に昆虫の世界では，個々は単純であっても群れとしては望ましい制御が行われる“群知能”と呼ばれる性質が見られる．蟻や蜂は周辺の状況の評価，そして周辺の個体との相互作用を繰り返し，種として持っている単純なルールに従いながら自身の行動を決定する．そしてその行動が群れ全体としての秩序を作り出す．群知能は環境への高い適応性を誇り，多くの分野から注目を集めている．そしてこのモデルは，やはり単純な機能しか持たないセンサノードから構成されるセンサネットワークについてもよく適合すると考えられる．

我々は蟻の群知能を用いて，マルチシンク構成のセンサネットワークを対象とした新しい枠組みを考案した．提案手法ではまず，ant-based clustering [8-10] のアイデアを用いて，各センサノードがいずれかのシンクノードに属するようなクラスタを構成する．クラスタは状況に応じて形状を変えながら，パケットの中継負荷を分散するクラスタ内で行うルーティングには蟻の採餌行動に見られる群知能である ant colony optimization (ACO) を適用し，確率的に次ホップのセンサノードを切り替えながらシンクノードへとパケットを送り届ける．本稿では計算機シミュレーションにより，提案手法が劣悪な通信環境においても情報収集を可能にするとともに，蟻の群知能が持つ環境への適応性，自己治癒能力を生かし，センサノードの故障，さらにはシンクノードの故障にも適応する耐久力，自己治癒能力を持ったセンサネットワークを実現することを示す．

本稿の構成は次の通りである．第 2 章では提案手法の詳細について説明する．第 3 章で提案手法の効果を計算機シミュレーションによって確かめる．最後に，第 4 章で本稿のまとめと今後の課題について述べる．

## 2. 提案手法

### 2.1 対象とするセンサネットワーク

我々が対象とするのは，イベント検知型，そしてマルチシンク構成を取るセンサネットワークである．図 1 のように，複数のセンサノードおよび多数のセンサノードが観測領域に設置される．各センサノードは温度の上昇や異常音の発生などといっ

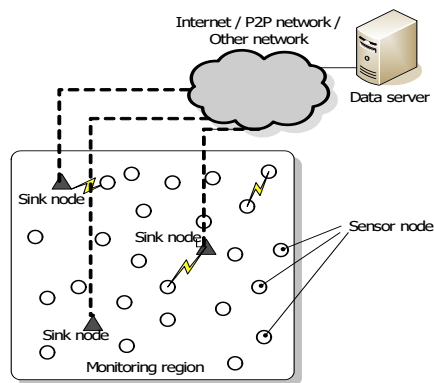


図 1 対象とするセンサネットワークの概略図

たイベントを検出し，その情報を含めたパケットを生成する．これらのパケットは他のセンサノードによって中継されながらシンクノードへと到達する．シンクノードはインターネットなど他のネットワークへのゲートウェイとして働き，シンクノードへ集められたデータはそのネットワークを介してデータサーバに届けられ，ユーザによるイベント情報の分析が可能となる．ただし，我々はセンサノードからシンクノードまでのパケットの伝送に注目し，シンクノードにおける処理やデータサーバへの伝送などは本稿の対象外とする．

### 2.2 提案手法の概要

我々は，ネットワークをシンクノードの数と同数のクラスタに分割し，それぞれのクラスタで，そのクラスタに割り当てられたシンクノードまでのルーティングを行うことを提案する．特定のクラスタへ負荷が集中した場合などの残余電力の偏りやシンクノード故障に対応するためには，クラスタは環境に応じて形状を変えていくことが求められる．そこで提案手法においては，クラスタリングに ant-based clustering を適用し，状況に応じてクラスタを拡大あるいは縮小させるようにした．これにより，ネットワークの状況に適応したクラスタが形成でき，ネットワーク全体としての長寿命化が期待できる．また，このような動的なクラスタリングを行うことで，シンクノードが故障した場合についても故障前と同程度の性能を発揮させることが可能となる．

クラスタ内のルーティングには ACO の考えを応用したホップバイホップのルーティングを適用する．各センサノードはフェロモンテーブルを持ち，隣接ノードの次ホップノードとしての望ましさを，フェロモンとして蓄えている．イベントの情報を運ぶパケットを送出するとき，あるいは転送するとき，各センサノードは自身のフェロモンテーブルを参照し，フェロモンの値に応じて確率的に次ホップのノードを決定する．従って，各センサノードは次ホップノードとして望ましいと考えられるセンサノードほど次ホップノードとして選ぶ確率が高くなり，また，もし同じ程度の望ましさを持つセンサノードが複数存在する場合は，それらは同じ程度の頻度で次ホップノードとして選択されるため，中継負荷を分散することも期待できる．

### 2.3 ルーティングアルゴリズム

ACO を適用しているルーティングアルゴリズムは種々あり，パケット交換ネットワークに対する提案 [11] や MANET に対するもの [12] ，センサネットワーク用に改良したもの [13] などがあるが，それらの基本的な考えを図 2 に示す通りである．送信ノードは経路構築のため，“ant” と呼ばれる小さな制御パケットをフラッディングする．ant は宛先ノードを探しながらネットワーク中を歩き回るとともに，自身の通った経路の品質を記憶する．宛先ノードに到達した ant は，それまでにたどった道を逆にたどりながら送信ノードへ戻り，その途中のノードに経路の品質に応じたフェロモンを残す．その後送信ノードから送信される ant は，高いフェロモンの残った経路を確率的に選ん

表 1 各項が持つ意味

|                     |   |
|---------------------|---|
| $n_i$ :             | センサノードの ID                                    |
| $S_k$ :             | シンクノードの ID およびそのシンクノードが担当するクラスタ               |
| $Pb_{S_k}(n_i)$ :   | $S_k$ が送出した backward ant に $n_i$ が持たせるフェロモン値  |
| $S_{n_i}$ :         | $n_i$ が属するシンクノードの ID                          |
| $P_{n_i}(n_i)$ :    | $n_i$ が申告する自分のフェロモン値                          |
| $P_{n_i}(j, S_k)$ : | $n_i$ において、宛先を $S_k$ 、次ホップノードを $j$ とするフェロモンの値 |
| $C_{n_i}(S_k)$ :    | $n_i$ の考える $S_k$ のクラスタに関するクラスタフェロモン           |

でいくことで、効率よく宛先ノードに到達し、やはり帰り道にフェロモンを残す。このような正のフィードバックが ACO の原理であり、これによってパケットはより良い経路を通りやすくなる。

我々の提案手法中で用いるルーティングも基本的には同じであるが、センサネットワークにおいて宛先となるのは少数のシンクノードに限られるため、全てのセンサノードがシンクノードを探すために ant をフラディングするより、シンクノードがフラディングする方が unnecessary 電力消費、そして帯域の消費を抑えることができる。この ant を *backward ant* と呼ぶことにする。backward ant の役割は、望ましい次ホップノードに、つまり隣接センサノードのうち、(1) unnecessary な迂回をすることなくシンクノードへ向かう、(2) シンクノードへ向かう経路上のセンサノードの残余電力が十分にある、という 2 条件を満たす経路上にあるセンサノードに高いフェロモンを持たせることである。

ここで説明のために、表 1 に説明に用いる項とその意味を定義する。 $S_a$  は、最大のフェロモン  $Pb_{S_a}(S_a) = P_{max}$  を持たせた backward ant  $B$  をブロードキャストする。 $B$  を受信したセンサノード  $n_i$  は、 $B$  の持つフェロモンの値  $P_{max}$ 、 $B$  の送信元である  $S_a$ 、そして  $B$  を中継したノード（この場合は  $S_a$ ）を 1 つのエントリとして図 3 のように自身のフェロモンテーブルに書き込む。こうして  $n_i$  は  $S_a$  にパケットを送信する際、 $S_a$  の次ホップノードとしての望ましが  $P_{max}$  であることを記憶する。 $n_i$  はその後、 $B$  に新たなフェロモン値を持たせ、 $B$  を再び転送する。新たなフェロモン値  $Pb_{n_i}(S_a)$  は次の式で計算される。 $E_{r_i}$ 、 $E_{m_i}$  はそれぞれ、 $n_i$  の残余電力、初期電力である。

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

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

$n_i$  によって転送された  $B$  を受信したセンサノード  $n_j$  は、 $n_i$  と同様にフェロモンテーブルにエントリを追加する。同様の計算によって新たなフェロモン値を  $B$  に与え、再度転送する。これを繰り返すことで、理想的なフェロモン分布が形成される。

backward ant の実体は非常に小さなパケットであるが、フラディングを行う際はやはり帯域、転送に要する電力などの消費が大きくなる。したがって、backward ant の送出頻度は低く抑えるべきである。一方で backward ant が送出される間にネットワークのトポロジの変化—人間などの遮蔽物の通過など—がある場合、最新のネットワークの状態をフェロモンテーブルに反映させなければ、効率の良い経路が形成できない。そこで私達は、フェロモン分布を最新の状態に保つ役割を、*hello ant* に持たせることにした。*hello ant* は backward ant より高い頻度で定期的に出送されるブロードキャスト性のパケットであり、送信ノード  $n_i$  の持つフェロモン値  $P_{n_i}$  と、 $n_i$  の属するシンク ID  $S_{n_i}$ 、そのシンクノードが担当するクラスタのフェロモン値  $C_{n_i}(S_{n_i})$ （詳しくは 2.4 で述べる）を運ぶ。*hello ant* の交換というセンサノード間のインタラクションによって、最新のネットワークの状態がフェロモン分布に反映させる。*hello*

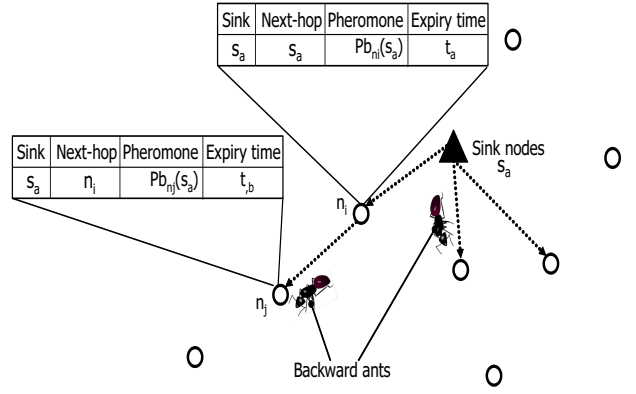


図 3 backward ant によるフェロモンテーブルの更新

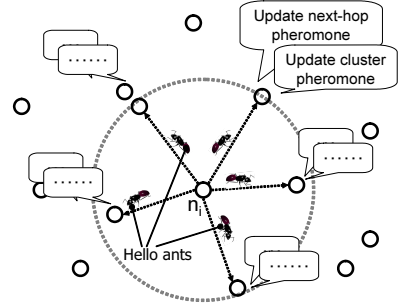


図 4 hello ant を受信したセンサノードはテーブル中のフェロモンを更新する

ant を受信したノード  $n_j$  は、自身のフェロモンテーブルの中に次ホップノードとして  $n_i$  が記憶されているエントリがある場合のみ、そのエントリのフェロモン値  $P_{n_j}(n_i, S_{n_i})$  を以下のように更新する（図 4）

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

ただし、 $\gamma \in [0, 1]$  である。また、自身が *hello ant* を送出する際に運ばせる自身のフェロモン  $P_{n_j}$  はフェロモンテーブル中のエントリのフェロモン値の平均

$$P_{n_j} = \frac{\sum_l \sum_m P_{n_j}(l, m)}{\sum_l \sum_m 1} \quad (3)$$

で定義される。

フェロモン分布が適切に構築されていれば、イベントを通知するパケットをシンクノードまで届けることは難しい。センサノード  $n_i$  がイベント通知パケットを転送する際は、宛先を自分の所属するシンクノード  $S_{n_i}$  に設定し、フェロモンの値に応じて確率的に次ホップノードを選択する。 $n_i$  の次ホップノードの候補の集合を  $N_{n_i}$  とするとき、次ホップノードとして  $n_j$  を選択する確率  $p_{n_i}(n_j)$  は次の式で表される。

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

## 2.4 Ant-based clustering

ant-based clustering は元々、卵や幼虫をその大きさ順にクラスタリングする swarm intelligence である。蟻は歩き回りながら周囲の幼虫と大きさ（類似度）の異なる幼虫その場所から拾い、類似度の高い幼虫の多い場所に運ぶことを繰り返すことで自己組織的に、大きさの異なる幼虫のクラスタを構成する。我々は ant-based clustering における“類似度”を“クラスタとしての望ましさ”に置き換えて、状況に応じたクラスタリングを行う。

クラスタはセンサノードの集合である。したがって、クラス

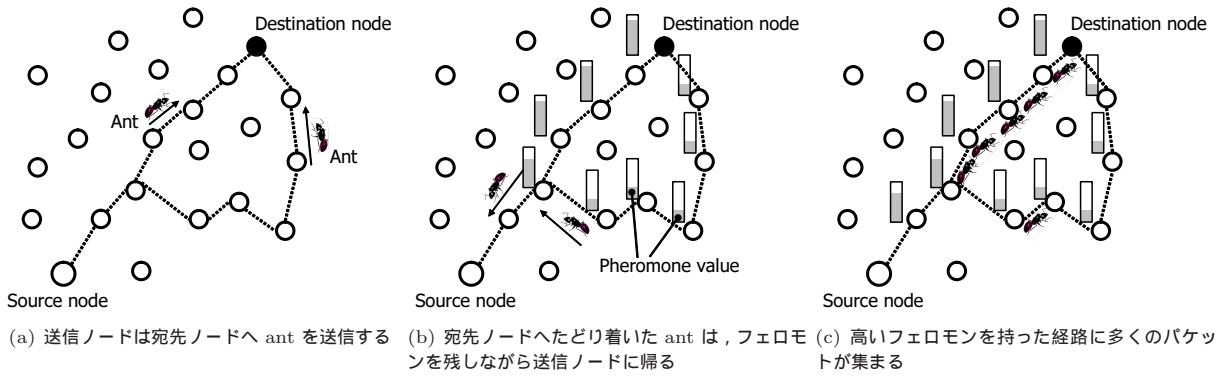


図2 ACO のアイデア

タの望ましき—クラスタフェロモン—を決める際は、そのクラスタを担当するシンクノードが正常に稼動していることはもちろん、そのクラスタを構成するセンサノードの状態、つまりクラスタに所属するセンサノードの残余電力が豊富であることや、それらセンサノードのシンクノードまでの距離が短いことを考慮すべきである。これらは既に 2.3 で紹介したフェロモン値に反映されており、したがってクラスタフェロモンの計算にはこのフェロモンを用いる。センサノード  $n_i$  がクラスタ  $S_j$  に所属している場合、 $n_i$  は hello ant に、自身の考える  $S_j$  のクラスタフェロモン  $C_{n_i}(S_j)$  を運ばせる。これを受信したセンサノード  $n_k$  は、 $n_i$  が主張する  $S_j$  のクラスタフェロモンが  $C_{S_j}$  であることを *Neighbors table* に記憶しておく（現在はフェロモンテーブルとは別のテーブルとしているが、この 2 つのテーブルは容易に 1 つに結合できる）。ここで  $C_{n_i}(S_j)$  は

$$C_{n_i}(S_j) = \frac{\sum_{k \in \text{belong}_{n_i}(S_j)} C_k(S_j) + \text{avg-ph}_{n_i}(S_j)}{|\text{belong}_{n_i}(S_j)| + 1} \quad (5)$$

で定義される。 $\text{belong}_{n_i}(S_j)$  は、 $n_i$  の隣接センサノードのうちクラスタ  $S_j$  に属するセンサノードの集合を表す。 $\text{avg-ph}_{n_i}(S_j)$  は式 (6) で表され、フェロモンテーブルにおいて宛先ノードが  $S_j$  であるようなエントリのフェロモン値の平均である。

$$\text{avg-ph}_{n_i}(S_j) = \frac{\sum_{k \in \text{belong}_{n_i}(S_j)} P_{n_i}(k, S_j)}{|\text{belong}_{n_i}(S_j)|} \quad (6)$$

式 (5) は難しく見えるかもしれないが、単に  $S_j$  に属する隣接ノードのクラスタフェロモンと、 $n_i$  の考える  $S_j$  のクラスタフェロモン  $\text{avg-ph}_{n_i}(S_j)$  の平均を取っているだけである。

hello ant は所属しているクラスタ ID も運んでいるため、センサノードは周囲のセンサノードがどのクラスタに所属しているかどうか、そして自身がクラスタの境界にいるかどうかを知ることができる。このような境界にいるセンサノードは、蟻が幼虫を別の場所へ運ぶかどうかを選択するのと同様に、現在加入しているクラスタから他のクラスタへメンバシップを変更するかどうかを確率的に選択する。実際の蟻は幼虫を運ぶかどうかを、幼虫とその周りの幼虫との類似度で判断する。我々は、この類似度をクラスタとしての望ましさに置き換え、[10] のモデルを参考に、センサノード  $n_i$  が、クラスタフェロモン  $C_{n_i}(S_j)$  を持つクラスタ  $S_j$  からクラスタフェロモン  $C_{n_i}(S_k)$  を持つクラスタ  $S_k$  へメンバシップを変更する確率  $p_{n_i}(S_j \rightarrow S_k)$  を以下のように定義する。

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

ここで  $f_{n_i}(S_j, S_k)$  は以下の式で計算される。

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

この確率に基づいて、逐次クラスタは変化していく。

## 2.5 故障検出

センサノードは安価に製造されるため一般に故障しやすく、また、長期間の稼動による電力枯渇も起こる。これはシンクノードとて例外ではない。長期にわたりセンサネットワークを用いて情報収集を行うには、これらの故障を検出し、そしてそれに対応することが必須である。

故障の検出は、2.3 で述べた hello ant を利用した Soft state モデルによって行われる。図 3 中にあるように、フェロモンテーブルの各エントリには expire time が設定されている。ノード  $n_i$  から hello ant を受信する度に、 $n_i$  を次ホップノードに持つエントリの expire time は更新されていくが、もしノードの故障や電力枯渇が起こると、そのノードから hello ant が届かなくなる。そして expire time を過ぎると、そのノードに関するエントリを削除する。このような Soft state モデルを用いることで、故障の検出については明示的に行う必要はない。センサノードが特別な処理を行わなくても自律的かつ適切に、式 (4) によって次ホップノードを切り替える。

シンクノードの故障検出に関しても同様で、シンクノードも定期的に hello ant をブロードキャストしている。この hello ant が一定時間届かなくなると、シンクノードの周囲のセンサノードは、そのシンクノードが故障であると判断する。シンクノードが故障した場合、そのクラスタはもはや望ましいクラスタではない。そのため、故障を検出したセンサノードは hello ant に運ばせるクラスタフェロモンに小さな値しか格納しない。この hello ant が伝播することで、故障したシンクノードのクラスタに所属していたセンサノードはそのメンバシップを放棄し、他のクラスタにメンバシップを移行する。

シンクノードの故障検出がセンサノードの故障検出と大きく違うところは、故障の誤検出が許されないことである。hello ant の伝送誤りやパケットロスなどによって一定時間 hello ant が届かないと、周囲のセンサノードはセンサノードやシンクノードを故障として誤検出してしまう。センサノードの故障の誤検出はそのセンサノードの周辺という極めて局所的な領域にしか影響しないが、シンクノードの故障の誤検出は、そのシンクノードの作るクラスタに所属していたセンサノード全てのメンバシップの変更を引き起こす。そこで、我々はシンクノードについてのエントリを削除するまでの時間  $\delta_S$  をセンサノードのエントリを削除するまでの時間  $\delta_n$  より長くすることにした。時間を長くすればするほど、誤検出をする確率は指数関数的に小さくなる。一方で、故障に迅速に適應するためには無効になるまでの時間は短いほうが良い。このトレードオフをどのように調節するかは、今後の重要な課題である。

表 2 センサノードのパラメータ

|         |          |         |          |
|---------|----------|---------|----------|
| 送信時送信電力 | 40.95 mW | 受信時消費電力 | 45.78 mW |
| 送信電力    | 0dB      | 送信距離    | 10 m     |
| 干渉範囲    | 16.8 m   | 送信周波数   | 2450 MHz |
| センシング範囲 | 10 m     | 送信速度    | 250 kbps |

表 3 シミュレーションパラメータ

|                   |      |                |       |
|-------------------|------|----------------|-------|
| 初期電力              | 1 J  | $\alpha$       | 0.7   |
| $\beta$           | 7    | $\delta_n$     | 3 s   |
| $\delta_S$        | 9 s  | hello ant 送出間隔 | 0.5 s |
| backward ant 送出間隔 | 10 s |                |       |

### 3. 性能評価

我々は ns-2 [14] を用いて、提案手法の性能評価を行った。以下特に述べない限り、各実験においては 1 辺 100 m の正方形領域に対し 200 台のセンサノードをランダムに配置し、また、4 台のシンクノードをあらかじめ決めた座標に配置した。シンクノードの配置される座標に関しては、以下のような 3 つのシナリオを用意した。

- 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)

MAC 層には IEEE 802.15.4 を用い、電波伝搬モデルには two-ray ground reflection モデルを用いた。センサノードのパラメータ値は [15, 16] を参考に表 2 のように、また他のパラメータについては表 3 に示す値を設定している。パラメータの影響についてはまだ十分に調べられていないが、ここに示した値であれば良い性能を示している。まだこれらの各実験において、500 秒のシミュレーションを 15 回繰り返し、本稿に示す結果はそれら 15 回の平均値である。イベントは毎秒、ランダムな位置に発生するものとした。

#### 3.1 イベント通知率

イベント検知を役割とするセンサネットワークにおいて、イベントの通知をどれだけシンクノードに届けられるかは最も重要な指標の一つである。我々はこれを反映するため、イベント通知率というメトリックを新たに定義する。イベント通知率はあるイベント  $e$  についてセンサノードが送出した全パケット数を  $s_e$ 、シンクノードが受信した  $e$  に関するパケット数を  $r_e$  とするとき  $r_e/s_e$  で表される。各シナリオについてこのイベント通知率を計測した結果が図 5 である。伝送誤りやルーティングループ等によっていくつかのパケットは棄却されてしまうが、各シナリオとも約 90% という高いイベント通知率を維持できている。また、BER の値が 0 と 0.3 のときを比較すると、イベント発生からシンクへ通知するまでの遅延は 100 ms から 150ms へと増加するが、イベント通知率についてはほとんど影響を受けない。つまり劣悪な通信品質下でも、イベントの情報をシンクノードに届けられることがわかる。

#### 3.2 ノード故障からの回復

提案手法がどれだけロバスト性を持つかを調べるために、以下のようにセンサノード故障のシミュレーションを行った。なお、“故障”の定義はいかなるパケットの送受信も不可能になる状態であるとしている。200 個のセンサノードが配置されているセンサネットワークにおいて、 $200 \times p_{\text{fail}}$  個のセンサノードを選び、 $t = 100$  s において一斉に同時故障させる。これは時間とともにノードが故障していくよりも、はるかにシビアな状況である。故障するノードは観測領域上のセンサノードからランダムに選択した。また、BER は 0.1 とし、scenario 1 を用いる。我々の興味は、 $t = 100$  [s] から落ち込むと予想されるイベント通知率が、どれだけ早く、そしてどこまで回復するかを評価することである。図 6 を見ると、 $t = 100$  s まで 90% 程度で推移してきたイベント通知率の線が、 $t = 100$  s で落ち込んでいることが確認できる。そしてさらに、わずかな時間でイベン

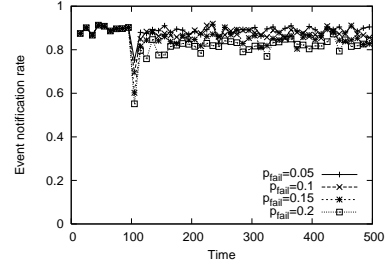


図 6 センサノードの同時故障に対する自己回復性

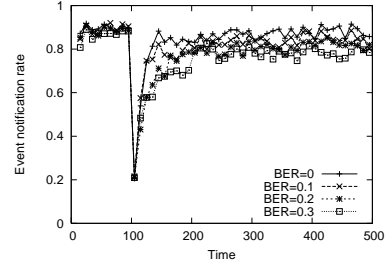


図 7 シンクノードの故障に対する自己修復性

ト通知率が回復し、 $p_{\text{fail}}$  が 0.1 程度であれば故障前とほとんど変わらないイベント通知率を示していることがわかる。ほとんどのセンサノードは、次ホップノードの候補をフェロモンテーブルに保持し、それらを確率的に選択している。ノードの故障を検出すると、そのノードを次ホップノードの選択肢からはずすだけでよく、すぐにイベント通知率は回復する。このような自己修復性は、よりディペンダブルなセンサネットワークの実現に大きく寄与すると考えられる。

#### 3.3 シンクノード故障からの回復

ノードが故障しえるのであれば、当然シンクノードも故障しえる。しかし、このようなシンクノード故障に対する対策は、これまでほとんど言及されることがなかった。我々は提案手法がシンクノードの故障についてどれだけ有効性を持つかを示すため、以下のようなシミュレーションを行った。scenario 1 を用い、(25, 25) に位置するシンクノード  $S_1$  を  $t = 100$  [s] において故障させる。 $S_1$  から遠くはなれた場所で起こるイベントはこの故障による影響をほとんど受けず結果が分かりにくいいため、本実験においては、イベントの発生場所を  $S_1$  から半径 20 m の円内に制限した。

実験の結果が図 7 である。 $t = 100$  s において、イベント通知率は約 20% まで大きく低下する。しかし、その後は急激に回復し、BER が小さければ  $S_1$  の故障前と同程度まで回復していることが確認できる。図 8 は故障前後のクラスタの状態を示したものである。二重円はシンクノード、単一円はセンサノード、センサノード横のゲージはセンサノードの残余電力を示している。故障前はシンクノードの個数と同じ 4 つのクラスタが形成されているが、 $S_1$  の故障によって、クラスタが 3 つに収斂する。各センサノードがインタラクションを繰り返し、自己組織的にクラスタの形を変えていくことで、シンクノードの故障前と同等のイベント通知率を達成できるようになった。これは提案手法のロバスト性と、自己修復性を極めて強く示している。

## 4. まとめ

本稿において、我々はスケーラビリティと耐故障性という 2 つの観点からマルチシンク構成のセンサネットワークが望ましいことを述べ、そしてそのセンサネットワークに適用する情報収集メカニズムを提案した。提案手法は ant-based clustering や ACO といった蟻の群知能に知見を得たもので、個々のセンサノードがセンサノード間でインタラクションを繰り返し、自

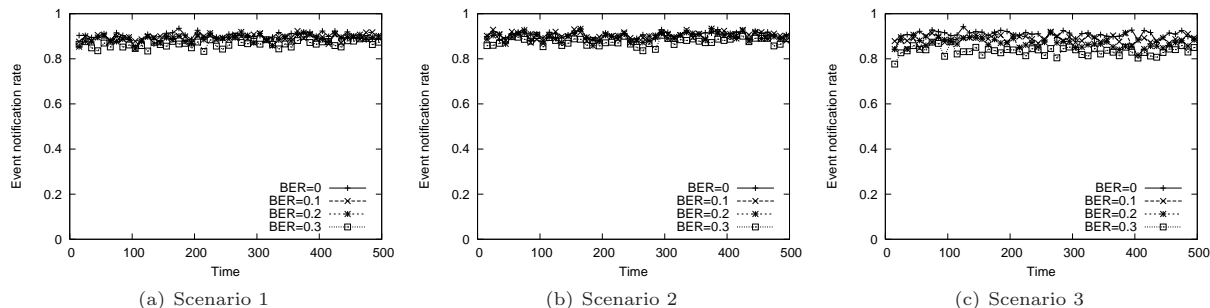


図 5 各シナリオにおけるイベント通知率の推移

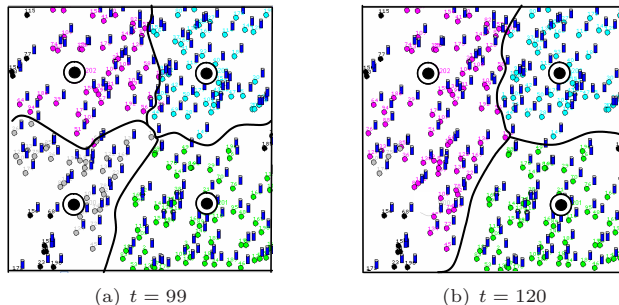


図 8 シンクノード故障前後のクラスタの形状

律的に個々の行動を決定し、センサネットワークを理想的な状態へと向かわせることができる。ant-based clustering によってセンサネットワークをクラスタに分割し、ACO を用いたルーティングによってイベント通知パケットをシンクノードまで運ぶこの情報収集機構は、安定して高いイベント到達率を示し、また、劣悪な通信チャネルに対してもロバストである。また、センサノードの故障、さらにはシンクノードの故障に対してさえも自己修復を行えることを示した。

自己組織性、自己修復性を併せ持つ我々の提案する情報収集機構は、劣悪な環境にも配置されうるセンサネットワークに望ましい性質を持っている。しかし我々はすでに、backward ant がイベント通知パケットと頻繁に干渉を起こしてしまうことや、backward ant の送出間隔を大きくするとロバスト性が次第に低下してしまうことなどを発見している。また、時間の関係により、各パラメータが性能に与える影響についての調査も道半ばである。今後はこれらの課題に取り組み、自己組織性を持つセンサネットワークと中央制御型のセンサネットワーク双方の性質および特性の違いを明らかにしたいと考えている。

## 謝 辞

本研究の一部は、文部科学省科学技術振興調整費「ゆらぎ」プロジェクト、および科学研究費補助金基盤研究 (C)17500043 によっている。ここに記して謝意を表す。

## 文 献

[1] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam and E. Cayirci: “A survey on sensor networks”, *IEEE Communications Magazine* (2002).

[2] Y. Kiri, M. Sugano and M. Murata: “Performance evaluation of intercluster multi-hop communication in large-scale sensor networks”, *Proc. Sixth International Conference on Computer and Information Technology* (2006).

[3] E. I. Oyman and C. Ersoy: “Multiple sink network design problem in large scale wireless sensor networks”, *Proc. International Conference on Communications* (2004).

[4] H. Luo, F. Ye, J. Cheng, S. Lu and L. Zhang: “TTDD: Two-tier data dissemination in large-scale wireless sensor networks”, *ACM Wireless Networks*, **11**, pp. 161–175 (2005).

[5] Y. Chen, E. Chan and S. Han: “Energy efficient multipath routing in large scale sensor networks with multiple sink nodes”, *Proc. Sixth International Workshop on Advanced Parallel Processing Technologies* (2005).

[6] M. Kalantari and M. Shayman: “Design optimization of multi-sink sensor networks by analogy to electrostatic theory”, *Proc. Wireless Communication and Networking Conference* (2006).

[7] H. Lee, A. Klappenecker, K. Lee and L. Lin: “Energy efficient data management for wireless sensor networks with data sink failure”, *Proc. Workshop on Resource Provisioning and Management in Sensor Networks* (2005).

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

[9] D. Zaharie and F. Zamfirache: “Dealing with noise in ant-based clustering”, *Proc. IEEE Congress of Evolutionary Computation*, pp. 2395–2402 (2005).

[10] 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).

[11] B. Barn and R. Sosa: “AntNet: Routing algorithm for data networks based on mobile agents”, *Inteligencia Artificial, Revista Iberoamericana de Inteligencia Artificial*, **12**, pp. 75–84 (2001).

[12] 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**, (2005).

[13] Y. Zhang, L. D. Kuhn and M. P. Fromherz: “Improvements on ant routing for sensor networks”, *Proc. Fourth International Workshop on Ant Colony Optimization and Swarm Intelligence* (2004).

[14] “ns-2 – the network simulator”, online available at <http://www.isi.edu/nsnam/ns>.

[15] Moteiv Corporation: “Telos (Rev B): PRELIMINARY Datasheet” (2004).

[16] Chipcon: “Chipcon AS SmartRF®CC2420 Preliminary Datasheet”, 1.2 edition (2004).