

センサネットワークのロバスト性に見る 集中制御と自己組織型制御の違い

木利 友一[†] 菅野 正嗣^{††} 村田 正幸[†]

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

^{††} 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: [†]{y-kiri,murata}@ist.osaka-u.ac.jp, ^{††}sugano@rehab.osakafu-u.ac.jp

あらまし 多数のノードから構成されるネットワークにおける制御について、自己組織的な制御が集中制御より望ましいという考えが一般的なものとなってきている。しかし、なぜ自己組織的な制御が望ましいのかというその理由が、十分な根拠を持って語られているわけではない。本稿では、センサネットワークにおける両手法の特性をシミュレーションにより比較し、自己組織的な制御はホップ数や遅延といった性能面では集中制御に劣るものの、無線チャネルの品質の悪化やセンサノード故障の発生といった環境の悪化に対しては集中制御よりも高いデータ収集率を保持し、観測領域全体からのデータ収集を維持できるなどといった、優れたロバスト性を持つことを示す。

キーワード センサネットワーク, 自己組織型制御, 集中制御, ロバスト性

Differences between Centralized Control and Self-Organized Control in Robustness of Sensor Networks

Yuichi KIRI[†], Masashi SUGANO^{††}, and Masayuki MURATA[†]

[†] Graduate School of Information Science and Technology

1-5 Yamadaoka, Suita-shi, Osaka, 565-0871, Japan

^{††} School of Comprehensive Rehabilitation, Osaka Prefecture University

3-7-30 Habikino, Habikino-shi, Osaka, 583-8555 Japan

E-mail: [†]{y-kiri,murata}@ist.osaka-u.ac.jp, ^{††}sugano@rehab.osakafu-u.ac.jp

Abstract Regarding control of networks composed of a large number of nodes, one vague notion seem to be common among researchers. It is that self-organized control is preferable to centralized one. The reason of the notion is, however, not described with clear rationale. To clarify the correctness of the notion, we compare the characteristics of both control approaches through simulation experiments on sensor networks. We show in this paper that although hop counts and delay of self-organized control are not beyond centralized ones, self-organized control can keep higher data collection rate and can suppress generation of the region where data gathering cannot be performed in poor quality of wireless channels and sensor node failures.

Key words sensor network, self-organized control, centralized control, robustness

1. はじめに

ネットワークは、より大規模化、複雑化している。例えばインターネットには多様なデバイスがつながれるようになり、アドホックネットワークやP2Pネットワークではノードの参加や離脱、あるいは故障などによって、ネットワークの状態は刻々と変化していく。このような、規模の大きい、あるいは変化の激しいネットワークにおける制御のアプローチとして、現在注目を浴びているのが自己組織型の制御である。そもそも自己組織型の制御が注目されているのは、一つまたは少数の制御局が

ネットワーク全体にわたる正確な情報を把握し、その情報を基に各構成要素の動作を決定し、その情報を構成要素に伝達するという集中制御の枠組みが、大規模あるいは変化の激しいネットワークに十分に対応できないためである。たとえば、制御対象の増加に伴って全体の情報を把握することが困難になる、制御局からの情報を待っていたのでは変化し続けるネットワークに追従できない、ノード障害をローカルに処理できないために迅速な回復処理が難しいといった適応性やロバスト性の問題が限界説の根拠となっている。自己組織的な制御への期待は、まさにそれらの問題を解決することに対する期待である。自己組

織型の制御においては、全体を統括する制御局は存在しない。ネットワークの各構成要素は、制御局からの指示を待つ代わりに周囲の構成要素と局所的な情報をやり取りし、自身の次の行動を決定する。そしてその個々の構成要素の行動が、全体としては望ましい制御として発現する。

自己組織的な制御の好例は、蟻が餌への最短経路を選択したり、蛍の明滅が全体として同期するというような自然界における現象である。蟻や蛍が個々として持つ能力は非常に単純であり、当然ながら各個体は、大多数の個体から構成される全体としての行動を把握することなどできない。また、季節や天候といった様々な環境の変化にもさらされる。しかしそのような環境変化を受けても多数からなる群れとしての調和を取り、その目的を果たす“群知能”とも呼ばれる彼らの自己組織的な制御は、集中制御とは根本から異なるアプローチであり、センサネットワークやアドホックネットワーク、P2P ネットワークへの自己組織的な制御の適用が多く提案されている [1-3]

自己組織的な制御が注目を浴びる中で数多く言及されている、集中制御に対する自己組織的な制御のロバスト性や適応性の優位な点は必ずしも自明なものではない。しかしながら我々の知る限り、ネットワークにおける自己組織的な制御と集中制御を比較した研究というのはほとんど存在しないまま、一般論としての自己組織的な制御の望ましさを概念が定着しているのが現状である。本稿の目的はロバスト性という観点で自己組織型制御と集中制御を比較し、その性質の違い、集中制御の限界と自己組織型制御の有用さを示すことである。

本稿では、大規模かつ変化し続けるネットワークの題材としてセンサネットワークを取り上げる。センサネットワークは観測領域中に配置された個々のセンサノードが、センシングによって周囲の情報を取得しシンクへ送信する、情報収集を目的としたネットワークである。配置されるセンサノードの数は時に数百、数千に及ぶとともに、センサノードの故障や電力枯渇によりネットワークの状況変化が著しく、スケーラビリティや適応性が強く求められる分野である。我々は、[4] で提案しているセンサネットワーク用の自己組織型の制御と、[5] で提案されている制御を元にした集中制御を様々な無線チャネルの品質やノードの故障シナリオの下で比較し、それらの環境下で両制御手法がどのような振る舞いを見せるかを評価する。

以下、2章では比較に用いた集中制御、自己組織型制御の概要についての説明を行う。3章では、両手法を様々なシナリオの元で比較し、集中制御の限界と自己組織型制御の利点を明らかにする。最後に4章で、本稿のまとめと今後の課題を述べる。

2. 制御手法の概要

ここで対象とするのは、観測領域中に複数のシンクが配置されるセンサネットワークであり、シンクは有線によってサーバに接続されることを想定する。このネットワークの例を図1に示す。以下、集中制御と自己組織型制御の概要について説明する。

2.1 集中制御

[5] では、電力制約がなく処理能力に優れた、ゲートウェイと呼ばれる特別なノードの存在を前提としたデータ収集手法を提案している。それは多数のセンサノードをゲートウェイと同数のクラスタに分割し、ゲートウェイが各クラスタ内のノードを管理する集中型の制御である。[5] で述べられているクラスタの役割は、我々が[4] で用いているものとほぼ同じであり、比較を行うに適していた。ただし、[5] ではクラスタをどう構成するかなどの詳細な点については述べられておらず、前提としているスリープ制御についても本稿では考慮しないため、我々はそこで提案されている制御を元に、クラスタリングやコスト

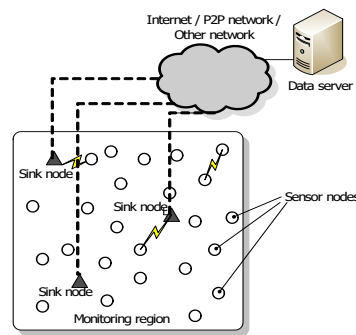


図1 ネットワークモデル

関数、故障検出など、いくつかの修正を施した集中制御手法を用いることとした。

我々の用いた集中制御手法においては、サーバは全センサノード、シンクの座標と、センサノードの残余電力を管理する。サーバはまず、センサノードをシンクと同数のクラスタに分割する。ここでのクラスタとは、各センサノードがどのシンクへセンシング情報を送信するかを表すものであり、センサノード n_i がシンク s_j を宛先としてセンシング情報を送信する場合、“ n_i はクラスタ s_j に属する” という。サーバはシンクを基準点とした Voronoi 分割によってクラスタリングを行い、各センサノードが最も近いシンクへとセンシング情報を送信するようにセンサノードを分割する。

クラスタリングの後、サーバはパケットの辿る経路の構築を行う。経路構築は [5] で提案されているとおり、各センサノードを結びリンクに対してコストを割り振り、各センサノードとあて先となるシンクを結ぶパスのコストの合計が最小になるように Dijkstra のアルゴリズムを用いて行われる。パスのコストには若干の変更を施し、センサノード n_i から n_j へのパスのコスト C_{ij} は式 (1) のようにセンサノードの残余電力とノード間の距離によって定義されるものとする。この式は、電波伝搬モデルとして Two-ray ground refraction model [6] を使用したときの消費電力を考慮したものである。

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

ここで、 E_{I_j} 、 E_{R_j} はそれぞれノード n_j の初期電力と残余電力を、 λ は電波の波長、 h はアンテナの高さ、 $d(n_i, n_j)$ はノード n_i 、 n_j 間の距離を表している。閾値 δ は $\delta = \frac{4\pi h^2}{\lambda}$ で定義される定数であり、 r_{\max} はセンサノードの通信距離である。

経路計算を終えると、サーバはシンクに経路情報を送信する。便宜上、この情報を含むパケットを“コマンドパケット”と称する。シンクは自身のクラスタに属し、かつ自身から最も離れたセンサノードがその情報を受信できるだけの電力を用いて、送信を行う。このパケットは各センサノード n_i に対して、以下の情報を提供する。

- 属すべきクラスタ
- n_i にパケットを送信してくる前ホップのノードと、そのパケットを送信する次ホップとなるセンサノードの組

n_i は、この情報を自身のテーブルに保持し、パケットの中継を行うことになる。

センサノードの故障検出は soft-state モデルに基づく。各センサノードは一定の間隔 t_{hello} で hello メッセージを送信している。センサノード n_i の周辺に位置するセンサノード n_j は、 n_i からの hello メッセージを受け取ると、 n_i のエントリを隣

接センサノードテーブルへと登録し、 n_i は故障していないと判断する。エントリは、expiry-time フィールドを含む。このフィールドは n_i から hello メッセージを受け取るごとに更新されるが、一定時間以上 hello メッセージを受信できず、テーブルに追加されてからの時間が expiry-time フィールドの値を超えた場合、 n_j は n_i が故障したと判断する。故障を検出したセンサノード n_j は、 n_i に関する故障通知パケットをシンクへと送る。このパケットは、サーバが計算した経路を通り、シンクへと到達する。シンクからこの情報を受信したサーバは、 n_i を経由しない経路を新たに計算し、その経路情報を、シンクを介して即座にクラスタ内のセンサノードへと送信する。

干渉や伝送誤りによって、センサノードが正常なのにもかかわらず、そのセンサノードから hello パケットが一定時間届かないという状況も起こりうる。 n_j は n_i の故障を検出したことを記憶している。この状態で n_i から hello パケットが届くようになると、 n_j は故障検出が誤りであったと判断し、故障通知パケットと同様に故障回復通知パケットをシンクへと送信する。これを受信したシンクは、サーバにこの情報を中継し、サーバは n_i を考慮にいれた経路を新たに計算し、やはりシンクを介してクラスタ内のセンサノードにブロードキャストする。以上の処理によって、故障の誤検出に対応する。

2.2 自己組織型制御

自己組織型制御は、蟻の群知能の 1 つである Ant Colony Optimization (ACO) [7] と Ant-based clustering [8, 9] に着想を得た手法である。センサノードをシンクと同数のクラスタに分割し、その各クラスタ内でルーティングを行う。

蟻は餌を探す際、他の蟻が残したフェロモンが多く残っている経路へ誘引されやすいという性質により、餌への効率的な経路を辿ることができる。そしてこの蟻は経路を辿ると同時に、自身もその経路にフェロモンを残す。結果として、経路にフェロモンが多いほど多くの蟻をひきつけ、その経路のフェロモンはさらに多くなっていく。このようにして蟻は集団として効率の良い経路を形成していくことができる。

ACO は、この蟻の採餌行動に見られる郡知能から生まれた、組み合わせ最適化問題への確率的なアプローチであり、現在は我々が用いているのと同様にルーティングにも利用されている [2]。ACO をルーティングに利用する場合の重要な問題は、どのような経路にフェロモンを多く残すべきか、つまりどのような経路を望ましい経路と定義するかである。我々は望ましい経路を下記 2 点を満たすものと定義した。

- 少ないホップ数を經由してシンクへ到達する
- 残余電力の豊富なセンサノードのみを經由する

そして上記の考えを反映するフェロモンの分布を実現するために、シンクは定期的に *backward ant* と呼ばれる制御パケットをフラッディングする。*backward ant* は $P_b = P_{\max}$ のフェロモン値を持ってシンクから送出される。この *backward ant* を受信したセンサノード n_j は、*backward ant* の持つフェロモン値を、自身のフェロモン値として記憶するとともに、*backward ant* の運ぶフェロモン値を以下のように更新して再ブロードキャストする。

$$P_b \leftarrow \alpha \left(1 - \exp \left(\beta \frac{E_{R_j}}{E_{L_j}} \right) \right) P_b \quad (2)$$

ただし $0 < \alpha < 1, \beta > 0$ である。*backward ant* の運ぶフェロモン値を式 (2) のように更新することにより、経路上のセンサノードに残余電力が十分にあり、シンクに近づくほどがフェロモン値が高くなるようにフェロモン値が分布する。

センサノードは、周囲のセンサノードとのフェロモン値を交換し、周辺のフェロモン分布を把握するため、集中制御と同様

に hello メッセージを定期的やりとりする。自己組織型制御における hello メッセージは、送信ノード n_i の持つフェロモン値 P_{n_i} と、 n_i の所属するクラスタ s_{n_i} 、クラスタ s_{n_i} の持つクラスタとしてのフェロモン $C_{n_i}(s_{n_i})$ を運ぶ。ここで $P_{n_i}(n_j, s_k)$ を、 n_i がシンク s_k に対してパケットを送る際、次ホップノードとして n_j を選ぶときのフェロモンの値とすると、 P_{n_i} は n_i の知っているフェロモン値の平均で定義される。

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

hello メッセージを受信したノード n_j は、次ホップノードとしての n_i のフェロモン値 $P_{n_j}(n_i, s_{n_i})$ を次の式のように更新する。

$$P_{n_j}(n_i, s_{n_i}) \leftarrow \gamma P_{n_j}(n_i, s_{n_i}) + (1 - \gamma) P_{n_i} \quad (4)$$

ここで、 $\gamma \in [0, 1]$ である。以上のようにして構築されたフェロモン分布を利用して、センサノードは次ホップノードを確率的に選択しながらパケットを中継していく。センサノード n_i があて先を s_{n_i} とするパケットを転送する際、次ホップノードの候補となるセンサノードの集合を N_{n_i} とすると、次ホップノードとして $n_j \in N_{n_i}$ を選択する確率は以下の式で表される。

$$p_{n_i}(n_j) = \frac{P_{n_i}(n_j, s_{n_i})^2}{\sum_{k \in N_{n_i}} P_{n_i}(k, s_{n_i})^2} \quad (5)$$

クラスタリングも、蟻の群知能の 1 つである Ant-based clustering にヒントを得たものである。ルーティングに用いたフェロモンの値を元に、クラスタについてのフェロモン（クラスタフェロモン）を定義し、その値に応じて各センサノードはどのクラスタに所属すべきかを判断する。 n_i がクラスタ s_{n_i} に所属しているときの、 n_i にとっての s_{n_i} のクラスタフェロモンは以下のように定義される。

$$C_{n_i}(s_{n_i}) = \frac{\sum_{k \in \text{bng}_{n_i}(s_{n_i})} C_k(s_{n_i}) + \text{avg-ph}_{n_i}(s_{n_i})}{|\text{bng}_{n_i}(s_{n_i})| + 1} \quad (6)$$

ここで $\text{bng}_{n_i}(s_{n_i})$ は、 n_i の隣接センサノード集合のうち、クラスタ s_{n_i} に属するセンサノードの集合を表す。ここで、 $\text{avg-ph}_{n_i}(s_{n_i})$ は宛先となるシンクが s_{n_i} であるようなエントリのフェロモン値の合計である。

$$\text{avg-ph}_{n_i}(s_{n_i}) = \frac{\sum_{k \in \text{bng}_{n_i}(s_{n_i})} P_{n_i}(k, s_{n_i})}{|\text{bng}_{n_i}(s_{n_i})|} \quad (7)$$

クラスタフェロモンは hello パケットによって運ばれ、各センサノードは hello パケットの受信によって近隣のクラスタのフェロモンを知ることが出来る。センサノードはクラスタフェロモンの値の大きいクラスタほど所属すべきクラスタであると判断し、確率的に自身のメンバシップをクラスタフェロモンの大きいクラスタへと切り替える。センサノード n_i が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は以下の式で定義される。

$$p_{n_i}(s_{n_i} \rightarrow s_k) = \left(\frac{f_{n_i}(s_{n_i}, s_k)}{k_{th} + f_{n_i}(s_{n_i}, s_k)} \right)^2 \quad (8)$$

ここで k_{th} は確率を制御する定数である。また、 $f_{n_i}(s_{n_i}, s_k)$ は次の式で計算される。

$$f_{n_i}(s_{n_i}, s_k) = \max \left(0, \frac{|\text{bng}_{n_i}(s_k)|}{N_{n_i}} \frac{C_{n_i}(s_k) - C_{n_i}(s_{n_i})}{C_{n_i}(s_k)} \right) \quad (9)$$

表 1 センサノードに関するパラメータ

送信電力	0 dB
送信距離	10 m
干渉が及ぶ範囲	16.8 m
送信周波数	2450 MHz
ビットレート	250 kbps
アンテナの高さ	20 cm
初期電力	25 J
送信状態における消費電力	40.95mW/ms
受信状態にあるときの消費電力	45.78mW/ms

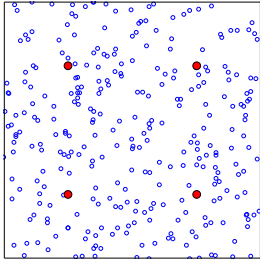


図 2 ノード配置の例

表 2 シミュレーションパラメータ

t_{hello}	1 s
t_{expire}	5 s
α	0.7
β	7
γ	0.875
k_{th}	0.5
hello パケットのサイズ	10 byte
故障通知用パケットのサイズ	10 byte
データパケットのサイズ	64 byte

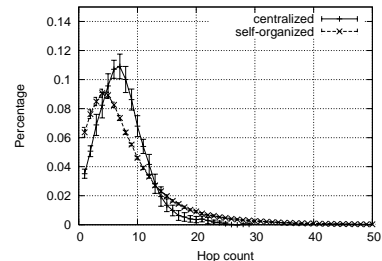


図 3 両制御手法の構築する経路の効率

[4] では、いったん中継したパケットが再びそのセンサノードに戻ってきた場合、ルーティングループとして検出し、そのパケットを廃棄していた。そして、この振る舞いがパケット廃棄率を上げる大きな要因であった。本稿ではこの点に修正を施し、中継したパケットであっても、何度でも再中継を行うこととした。

3. 両制御手法の比較

3.1 シミュレーションモデル

両制御手法の比較にあたっては、 100×100 m の観測領域に 300 台のセンサノードをランダムに配置するとともに、4 台のシンクをそれぞれ $(25, 25)$, $(75, 25)$, $(25, 75)$, $(75, 75)$ に配置したときのネットワークを用いる。この一例を図 2 に示す。原点は、観測領域の左下隅に取っている。

電波の伝播モデルには Two-ray ground refraction model を用い、MAC 層は IEEE 802.15.4 で動作する。集中制御を行う場合、シンクから送信される、制御情報を含むパケットのサイズは IEEE 802.15.4 の規定値を超えてしまうため、IEEE 802.15.4 で受け取ることのできる最大バイト数を表す $aMaxPHYPacketSize$ は事実上無限であるとした。また、センサノード n_i の前ホップ・次ホップノードを示すエントリの組が e_{n_i} 個、シンク s_j のクラスタに属するセンサノードが N_{s_j} とするとき、この制御パケットのサイズは、次式で与えられる。

$$\sum_i 6 \cdot e_{n_i} N_{s_j} + 7 \quad (10)$$

ここではエントリの組を表すのに 6 byte、そしてヘッダが 7 byte であると考えている。その他、シミュレーションに用いているセンサノードのパラメータを [10] を一部参考にして決定し、表 1 に、シミュレーションのパラメータを表 2 に示す。また、本稿では伝送誤りなどの影響に注目するため FEC などは用いず、1 ビットでも伝送誤りが発生すれば、そのパケットは棄却されるものとした。

本稿では、各センサノードが $t_{\text{interval}} = 10$ 秒ごとに周囲の状況をセンシングし、得られた情報をマルチホップ通信によってシンクへと送信するデータ収集モデルを考える。ただし、センサノードは互いに同期しておらず、したがってそれぞれのセンサノードの送信時間は他のセンサノードに無関係である。

3.2 パラメータの決定

情報収集を目的とするセンサネットワークにおける最も重要

な指標の一つは、データをシンクに届けることのできる信頼性である。それを踏まえて我々は本稿で用いるメトリック、データ収集率を定義する。今、正常に稼働しているセンサノードが N_{act} であるとき、 t_{interval} 間に生成されるデータパケット数も N_{act} である。このうちシンクまで伝送誤り無く到達するパケット数を r とするとき、データ収集率は r/N_{act} であるとした。

集中制御手法で最もデータ収集率に影響を与えるパラメータは、コマンドパケットの送信間隔である。この間隔が長ければ、伝送誤りによってパケットが棄却された場合に、センサノードは自身が何をすべきかという情報を長時間にわたり知ることができない。一方、あまりにコマンドパケットの送信間隔が短いと、コマンドパケットの受信のために、センサノードは大きな電力を必要とする。我々は間隔を 1 秒、10 秒、100 秒、500 秒とした場合のシミュレーションを行い、データ収集率の値や消費電力を鑑みて、コマンドパケットの送信間隔として 10 秒を選んだ。

自己組織型制御についても同様である。backward ant はフラッディングされるため、あまりに送信間隔が短いと干渉を引き起こし、一方あまりに送信間隔が長いと適切なフェロモン分布が構築できない。送信間隔として 10 秒、100 秒、500 秒とした場合のシミュレーションを行い、データ収集率の点から、backward ant の送信間隔として 100 秒を選択した。

なお、集中制御ならコマンドパケット、自己組織型制御なら backward ant の送信は、各センサノードが少なくともシミュレーション開始 100 秒までに行うこととしている。そのため以下に示す結果においては、開始 100 s までは過渡状態であるとしてグラフ中にプロットはしていない。

3.3 効率の良い経路構築

集中制御と自己組織型制御において、データパケットがセンサノードに送信されたからシンクに到達するまでのホップ数の分布を、95%信頼区間とともに図 3 に示す。このとき、BER は 0、故障も起こらない理想的な環境を想定しているが、BER の値を変えても大きな変化は見られなかった。また、遅延についてもほぼ同じ形の分布を示しており、集中制御はホップ数、遅延ともに値の小さい部分に多く分布していることが分かった。一方の自己組織型制御では、ホップ数の小さい部分にできてい山の高さは集中制御のそれより低く、逆にテールが長くなっている。グラフに示している x 軸の範囲は 50 ホップまでであり、集中制御の全パケットはこの範囲の中に納まるが、自己組

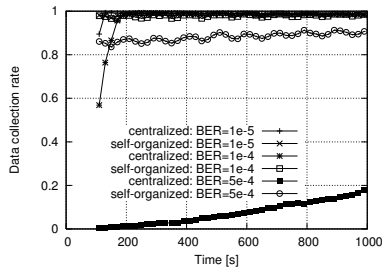


図 4 両制御手法に伝送誤りが与える影響

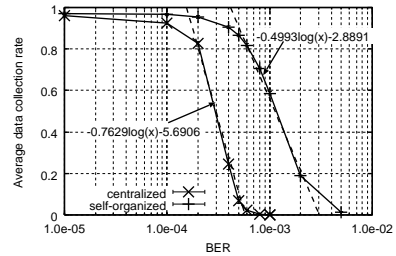


図 5 BER に対する両制御手法の耐性

織型制御のシミュレーションで生成されるパケットの中には、ホップ数では 900 を超えるものも存在していた。

集中制御では、制御局であるサーバはネットワーク全体の情報を取得可能であり、その情報に基づいて最適な経路を選択することができる。しかし自己組織型制御においては、各センサノードが周辺の情報のみに基づいて判断を下すため、集中制御と比較すると利用できる情報が少なく、必ずしも最適な経路を選ぶことはできない。つまり、構築される経路の品質という点で評価すると、集中制御のほうが良い経路を構築できている。

3.4 伝送誤りに対するロバスト性

センサノードの故障が生じないという仮定の下で、データ収集率の推移を調べたのが図 4 である。BER が 10^{-5} のときは両制御手法はほぼ同じデータ収集率を示すが、それより BER が高い通信環境でデータ収集を行う場合は、集中制御手法のデータ収集率の立ち上がりが著しく遅くなるが見取れる。

集中制御の形をとると、制御に必要な全ての情報がサーバに集まるため、サーバはそれらの情報を元に、ネットワーク全体として最適な制御を行うことができる。しかし、その最適な制御はセンサノードに伝わって初めて意味を成し、もし伝送誤りなどによってパケットが届かなければ、パケットをどのセンサノードに中継すべきか分からない、あるいは誤ったセンサノードにパケットを中継してしまうことになり、結果としてそのパケットは廃棄されてしまう。時間とともにデータ収集率が改善されるのは、10 秒という短い間隔で送信されるコマンドパケットが、伝送誤りによってそれ以前に廃棄されたコマンドパケットを補償するためであるが、立ち上がりには長い時間がかかるという事実は結局、経路が変更された場合の適応の遅さを示していると言える。さらに、もともと制御がネットワーク全体を視野に入れているために、一部分でこのような綻びが生じると、その影響は全体に波及する。たとえあるセンサノードが正しくパケットを受信できても、その上流のセンサノードが受信できていなければパケットはシンクへ到達することはできない。

一方で、自己組織型制御は各センサノードが周辺の情報を用いて自身の行動を決定する。もちろん個々のセンサノードはネットワーク全体の情報を知ることにはできないが、その分、個々のセンサノードの行動が影響する範囲もその周辺に限定され、ネットワーク全体に影響が波及することは無い。したがって完全な情報を手に入れる必要は無く、パケットが伝送誤りによって棄却されたとしても、その影響はごく僅かなものに抑えられる。それが結局図 4 のような、伝送誤りに対する高いロバスト性につながっている。

伝送誤りに対する両制御手法の違いは図 5 にも表れている。これは両制御手法の 100–1000 秒の間の平均データ収集率を各 BER についてプロットしたものであり、さらにそれぞれの制御手法について、データ収集率の立下りを対数近似している。集中制御手法の片対数グラフにおける傾きは -0.7620 であり、自己組織型制御の傾きは -0.4993 である。この傾きの差は、環

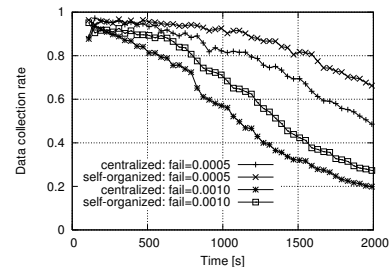


図 6 ランダム故障に対する耐性

境変化に対するロバスト性という形で現れる。低下の傾きが急な場合、わずかな環境変化、BER の変化によってネットワークの機能が破綻しかねない。一方でこの傾きが穏やかならば、多少の BER の変動があっても大きな影響を受けることなくデータ収集率が続けられる。したがって伝送誤りという点については、自己組織型制御のほうがロバストであると言える。

3.5 センサノード故障に対する耐性

故障率 p_{fail} を各センサノードについて一秒あたりに故障する確率と定義し、センサノードを時間とともに、ランダムに故障させた。例えば、 $p_{fail} = 0.01$ としたときは、各センサノードは毎秒 1% の確率で故障することになる。図 6 が 2000 s のシミュレーションを行ったときのデータ収集率の推移である。 p_{fail} の値として 0.0005 と 0.0010 の 2 つを用いて両制御手法を比べているが、どちらの値であっても、自己組織型制御の方が高いデータ収集率を示している。グラフからは読み取りにくいですが、例えば $p_{fail} = 0.0005$ の場合に、自己組織型制御が 1000 s までデータ収集率をある程度維持しているのに対して、集中制御はその時点で下降を始めている。

また、異なるシナリオとして、センサノードが集中的、かつ同時に故障する場合の影響も調査した。特定の座標を選び、その座標を中心として半径 10 m の円内にあるセンサノードを、 $t = 200$ s において同時に故障させている。図 7(a) がそのときのデータ収集率の推移を表しているが、自己組織型制御は故障エリアの中心が (15, 15), (20, 20) の場合においてはほとんど影響を受けない。また、故障の中心がシンクの座標 (25, 25) と一致する場合は、シンク周辺のセンサノードが全て故障するためにそのクラスタからの情報収集は不可能になるが、ant-based clustering によって他のクラスタの形状が時間とともに変化し、クラスタが 3 つに収斂して、データ収集率は回復傾向を示す。一方で集中制御は、中心点が (15, 15) にある場合は 200 s 以後もデータ収集率を維持できるものの、中心点が (20, 20), (25, 25) にある場合にはいったん収集率が落ち込んだ後、その落ち込みが回復することは無い。図 7(b) および図 7(c) は、故障の中心が (20, 20) であったときの集中制御、自己組織型制御におけるネットワークの状態を表している。二重円となっているノードが故障していることを表し、個々のノードの色は、各センサノードが $t = 200$ s 以後に送信したパケットがどれだけシンクに到達しているかという、パケット到着率を表している。色が白ならば送信した全てのパケットがシンクへ到着しているこ

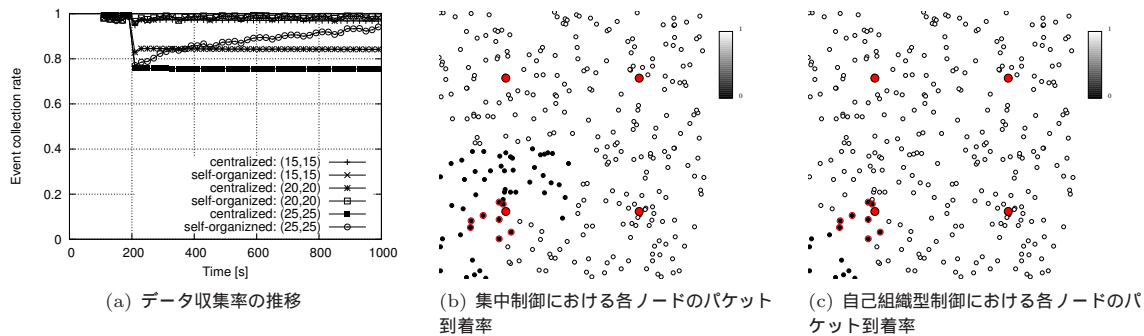


図 7 センサノードの同時・集中故障の影響

とを表し、黒ならばまったく到着していないことを意味する。自己組織型制御では、故障によりシンクへのコネクティビティを失ったノードを除けば、他のノードのケット到着率はほぼ 100% である。一方で集中制御では、故障の影響がクラスタ全体に及び、多くのセンサノードはケットをまったくシンクへ送ることのできない状況へ陥っている。このように特定エリアからの情報収集が不可能になることは、センサネットワークにおいては望ましいことではない。

集中制御がこのような故障に対して脆弱なのは、集中制御がネットワーク全体の正確な情報把握を前提としていることにある。もしセンサノードの故障などの情報が正しくサーバに伝わらなかった場合、サーバは誤った情報を基にした制御を行ってしまい、そしてその制御情報をネットワーク全体に配布してしまう。図 7 で起こっているのはまさにこの現象であり、サーバへセンサノードの故障通知が伝わらないことによって、サーバは故障しているノードを正常であると誤解し、そのノードを組み入れた経路を他のセンサノードに伝えてしまう。我々は、たった一つのセンサノードの故障によっても同様の現象が発生し、1 クラスタ中の多数のノードが、情報をシンクへと伝えられない場合があることも確認した。

逆に自己組織型制御では、故障への対応その他の判断は各センサノードそれぞれに任せられる。もちろん検出を誤ることも無いわけではないが、各センサノードは自身の周辺状況のみしか評価しないがゆえに、その誤りの影響も自身の周辺のみに限られる。また、周辺の状況进行评估するという事は、誤った判断は周辺のセンサノードの正しい判断によって修正されることの裏返しでもある。このような周辺のセンサノードとのローカルなやり取りは局所的には僅かな影響しか与えないが、それこそがネットワーク全体にロバスト性をもたらしている。

4. ま と め

本稿では、多数のノードから構成され、環境が著しく変化するセンサネットワークを題材として、集中制御と自己組織型制御の違いをロバスト性の観点から評価した。計算機シミュレーションを通して、集中制御は正確な情報が入手できるという前提の下では自己組織型制御よりも効率のよい制御を行える一方で、伝送誤りやセンサノードの故障などに対しては脆弱であり、自己組織型制御の方が優れたロバスト性を示すことを明らかにした。正確な情報の取得は、センサネットワークのように環境が変わりやすく、その影響も受けやすいネットワークでは特に困難であり、経路の品質を多少犠牲にしても、各種環境要因に対してロバストな手法である自己組織型制御を望む向きは間違っていないと言える。

本稿では全ての制御をサーバという集中制御局が行う完全な集中制御と自己組織型制御を比較したが、センサノードに一定

の判断を任せるような、自律分散的な面と集中制御的な面を併せ持つ制御手法も考えられる。今後はそのような別の制御手法との比較を基に、自己組織型制御の更なる性質を明らかにすることを考えている。

謝 辞

本研究の一部は、科学研究費補助金基盤研究 (A) 16200003 および (C)19500060 によっている。ここに記して謝意を表す。

文 献

- [1] K. H. Low, W. K. Leow and J. Marcelo H. Ang: "Task allocation via self-organizing swarm coalitions in distributed mobile sensor network", Proc. 19th National Conference on Artificial Intelligence, pp. 28-33 (2004).
- [2] 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).
- [3] Y.-W. Hong and A. Scaglione: "A scalable synchronization protocol for large scale sensor networks and its applications", IEEE Journal on Selected Areas in Communications: Special issue on Advanced Military Wireless Communications, **23**, pp. 1085-1099 (2005).
- [4] 木利 友一, 菅野 正嗣, 村田正幸: "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", 電子情報通信学会技術研究報告 (IN2006-256), pp. 453-458 (2007).
- [5] M. Younis, M. Youssef and K. Arisha: "Energy-aware routing in cluster-based sensor networks", Proc. 10th IEEE International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunications Systems (2002).
- [6] "ns-2 - the network simulator", online available at <http://www.isi.edu/nsnam/ns>.
- [7] M. Dorigo, V. Maniezzo and A. Coloni: "The ant system: Optimization by a colony of cooperating agents", IEEE Trans. on Systems, Man, and Cybernetics, **26**, 2, pp. 29-41 (1996).
- [8] D. Zaharie and F. Zamfrache: "Dealing with noise in ant-based clustering", Proc. IEEE Congress of Evolutionary Computation, pp. 2395-2402 (2005).
- [9] 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).
- [10] Moteiv Corporation: "Telos (Rev B): PRELIMINARY Datasheet" (2004).