

無線センサ・アクチュエータネットワークにおける 動的なサービスネットワーク構成のための端末選出機構の提案

岩井 卓也[†] 若宮 直紀[†] 村田 正幸[†]

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

E-mail: †{t-iwai,wakamiya,murata}@ist.osaka-u.ac.jp

あらまし アンビエント情報環境においては、環境内に配置された多数のセンサやアクチュエータなどのデバイスによって、その時、その場、その人に応じた情報サービスや環境制御が実施される。想定される多様な全てのサービスに対して、あらかじめ個別にデバイスを配置、設定し、ネットワークを構成するのは困難かつ冗長であるため、その場にあるデバイスを活用して柔軟にサービスを実現するための仕組みが求められる。そこで本稿では、随時発生するサービス要求に応じたネットワークを動的に構成するために、具備するデバイスや稼働状況を考慮して、その場で利用可能な端末の中からサービス要求を満たす適切なものが選出される機構を提案する。提案手法では、社会性昆虫における分業の仕組みの数理モデルである、反応閾値モデルを用いることにより、端末自身が自律分散的にサービスへデバイス提供を判断する。シミュレーション評価により、複数のサービス間で端末が共有される、効率的な端末の選出が行えることを確認した。

キーワード アンビエント情報環境, 無線センサ・アクチュエータネットワーク, 反応閾値モデル, タスク割当

Proposal of autonomous task allocation for dynamic formation of service networks over wireless sensor and actuator networks

Takuya IWAI[†], Naoki WAKAMIYA[†], and Masayuki MURATA[†]

[†] Graduate School of Information Science and Technology, Osaka University

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

E-mail: †{t-iwai,wakamiya,murata}@ist.osaka-u.ac.jp

Abstract In the ambient information society, embedded sensors detect and conjecture environmental and personal conditions and actuators provide users with information services and environmental control which are suited for time, place, occasion, and people. Although a single service cannot meet diverse requirements of different conditions, it is hard to deploy and configure a variety of devices for each of potential services. To solve the problem, we propose a mechanism to autonomously combine existing devices for dynamic organization of a service-oriented network. In our proposal, we adopt the mathematical model of division of labors in a colony of social insects to accomplish autonomous selection of devices taking into account their states and service requirements. Through simulation evaluation, we confirmed that active nodes are shared among multiple services requiring the same observation result.

Key words ambient information society, wireless sensor and actuator networks, response threshold model, task allocation

1. はじめに

環境に配置されたセンサによって取得された情報にもとづいてアクチュエータを制御し、環境制御や機器制御を行うセンサ・アクチュエータネットワークの研究開発が近年活発に行われている [1]. センサ・アクチュエータネットワークでは、適切なセ

ンサやアクチュエータを組み合わせることによって様々な制御やサービスを実現することができるが、多くの研究開発においては、サービスごとに必要なセンサやアクチュエータを有する端末が、あらかじめ定められた適切な場所、個数で環境内に配置され、ネットワークが構成されていることが前提となっている。なお、本稿では、温度や加速度などのセンサや、モータや

情報表示画面などのアクチュエータをデバイス、デバイスを搭載する無線通信機器を端末と呼ぶ。一方、来るべきアンビエント情報社会においては、環境に設置された多数のセンサによって環境や人に関する情報を取得し、アクチュエータによってその時、その場所、その人に応じた適切な情報サービスや環境制御が実施される。このような状況に応じた多様なサービスが求められる環境においては、配置するデバイスや端末の種類や数、場所をあらかじめ決定するのは困難であり、また、センシング領域が大きく重なり合うセンサが複数設置されるなど、サービス間で機能の重複したデバイスが導入されてしまう。したがって、サービスの区別なくあらかじめ様々な端末を配置しておき、その場で利用可能な端末からサービスの要求に応じた適切なものを選出して組み合わせることによって動的にサービスを実現するとともに、サービス間で積極的にデバイスを共用することによって通信帯域や電力の消費を抑制することが望まれる。また、通信帯域と電力消費の観点からは、稼働中の全てのサービスや端末の状態を集中管理して端末の組み合わせを決定するのではなく、端末自身の自律的な判断にもとづく自律分散的な手法が有効である。

このようなセンサ・アクチュエータネットワークの効率的利用に向けて、多様なデバイス、端末、ネットワークを効果的に組み合わせるサービスを実現するための手法が様々な提案されている [2-5]。文献 [2] では、搭載するデバイスや機能、OS などの異なる端末を統一的に取り扱うためのアーキテクチャ TinySOA を提案している。TinySOA により、開発者は端末の差異を意識することなく共通の API を用いてサービスアプリケーションを作成することができる。本稿においては、同様な SOA (Service Oriented Architecture) にもとづいて設計された端末を想定し、端末の選出機構を提案している。

また、複数のサービス間でセンサを共有するための手法としては、TinyONet が提案されている [3]。TinyONet は、定期的に全ての端末から基地局にセンシング情報を収集し、蓄積されたセンシング情報から適切なものを選んでサービスに提供する。その結果、サービスからは、サービスごとのネットワークが構築され、センシング情報が収集されているように見える。しかしながら、TinyONet は全端末から基地局への定期的なセンシング情報収集を前提としてセンシング情報の再利用を図る手法であり、アクチュエータの制御や端末間通信などには対応していない。一方、文献 [4] では Virtual Sensor Network (VSN) と呼ばれる、複数の異なるセンサネットワーク上をまたがる仮想的なセンサネットワークを構築する手法を提案している。VSN は異種ネットワークを統合的に利用するために有効であるが、VSN のメンバをあらかじめ別的手段を用いて決定する必要がある。また、VSN の構築、管理のために、VSN のメンバとメンバ間の経路上の端末からなる virtual tree と呼ばれるツリー構造を構成する必要があり、さらに、VSN 内の全ての通信がこの virtual tree を経由することから、管理オーバーヘッドが大きいという問題がある。

サービス要求に応じたデバイス割り当てについては、if-then-else ルールによって記述されたメッセージを拡散することによ

って条件にあった端末を選出する Generic Role Assignment と呼ばれる手法が文献 [5] で提案されているが、電力低下、障害、異なるサービス要求の同時発生など様々な状況に対処するためのルールをあらかじめ網羅しておく必要がある。また、Directed Diffusion [6] では、文献 [5] と同様に、サービスを要求する端末は、観測対象や条件、報告頻度などを記述した interest メッセージと呼ばれるサービス要求をネットワークに拡散することで、要求に応えることのできる端末を発見する。サービス要求に対してデバイスを提供できる端末は、センシング情報を含む exploratory data メッセージをサービス要求端末に向けて送信する。サービス要求端末は、最初に受け取った exploratory data メッセージの送信元端末に向けて、報告頻度を高く設定した reinforce メッセージと呼ばれるサービス要求メッセージを送信する。このことにより、サービス要求に応えることのできる端末のうち、望ましいものから頻繁にセンシング情報を受信するとともに、残りの端末からもセンシング情報を受信することで耐故障性を獲得している。Directed Diffusion において、exploratory data メッセージに残余電力など端末の状態に関する情報を付加すれば、サービス要求端末において様々な条件を考慮して reinforce メッセージを送信する端末を選択することができるが、Generic Role Assignment と同様に、その選択ルールの記述は容易ではない。

本稿では、具備するデバイス、電力容量や電力効率、処理能力などの様々な異なる端末が配置された環境において、複数のサービスの要求に対して、端末が自律的な判断にもとづいてサービスへのデバイスの提供の要否を決定することにより、サービス間で効果的にデバイスを使い分け、または共有する機構を提案する。提案手法は、社会性昆虫の群れにおける自己組織的な分業の仕組みを表した反応閾値モデル [7] を用いることで、複雑なルールの記述を必要とせず、サービス要求や端末の機能、稼働状態を考慮した適応的な端末選出を実現している。

本稿の構成は以下の通りである。まず 2 章において、反応閾値モデルについて説明する。3 章において、反応閾値モデルを用いてデバイスの割り当て、端末の選出を行う手法を提案し、4 章において、提案手法の有効性をシミュレーションにより評価する。最後に 5 章で本稿のまとめと今後の課題を述べる。

2. 反応閾値モデル

反応閾値モデル [7] は、社会性昆虫の群れにおける分業の仕組みをモデル化した数理モデルであり、それぞれの個体が仕事をするか、しないかを確率的に選択することによって、群れ全体で需要に応じた適切な割合の個体が仕事に従事するようになる。個体 i はある仕事に対して、仕事をするか、しないかを表す状態値 $X_i \in \{0, 1\}$ を持っており、定期的に状態値を更新する。 $X_i = 1$ の時には仕事に従事し、 $X_i = 0$ の時には仕事をしない。仕事をしていない個体 i が仕事を始める確率 $P(X_i = 0 \rightarrow X_i = 1)$ は次式で与えられる。

$$P(X_i = 0 \rightarrow X_i = 1) = \frac{s^2}{s^2 + \theta_i^2} \quad (1)$$

ここで、 $\theta_i (> 0)$ は個体 i の仕事に対する閾値であり、閾値 θ_i

が大きいほど個体 i は仕事をしにくく、また、閾値が小さいほど個体 i は仕事をしやすくなる。

また、 s (≥ 0) は仕事の需要を表す変数であり、需要が大きいほど個体は仕事をしやすくなる。ある時刻 $t+1$ における需要は次式で与えられる。

$$s(t+1) = s(t) + \delta - \frac{\alpha N_{act}}{N} \quad (2)$$

ここで、 δ ($1 \geq \delta \geq 0$) は需要の増加率を表す係数であり、 α ($1 \geq \alpha \geq 0$) は仕事の効率を表す係数である。 N_{act} は時刻 t において仕事をしている個体の数を表す変数であり、 N は群れの中で仕事を出来る個体の数を表す変数である。

一方、仕事をしてきた個体が仕事をやめる確率は定数 p ($1 \geq p \geq 0$) により次式で定義される。

$$P(X_i = 1 \rightarrow X_i = 0) = p \quad (3)$$

一時的に仕事をしない個体が増えたとしても、式 (2) により需要が増加することによって、閾値が高い個体もいずれ仕事をできるようになり、結果として仕事をする個体の割合が一定に保たれる。

さらに文献 [7] では、強化学習により仕事をする個体と仕事をしない個体の分化を図っている。個体 i が仕事をしなかったかによって、閾値 θ_i を次式により更新する。

$$\theta_i = \begin{cases} \theta_i - \xi, & \text{if individual } i \text{ performs a task} \\ \theta_i + \varphi, & \text{if it doesn't perform a task} \end{cases} \quad (4)$$

ξ (> 0), φ (> 0) は分化の速度と可塑性を定める定数である。この閾値調整により、いったん仕事を始めた個体の閾値が小さくなり、いったん仕事を辞めたとしてもすぐに仕事に戻るようになる。一方、仕事をしない個体はより仕事をしにくくなる。

3. 反応閾値モデルを応用した 端末選出機構

提案手法では、それぞれの端末を個体、サービス要求に対してデバイスを提供することを仕事とみなし、サービス要求に対応したデバイスを有する端末のうちデバイスを提供している端末の割合を需要とすることで、端末による反応閾値モデルを用いた自律的なデバイス提供の要否決定を実現する。

3.1 端末の管理する情報

端末は、サービス集合 S を管理している。自身のデバイスを提供可能なサービス要求を受信すると、サービス識別子 i とサービス要求メッセージの転送元の隣接端末の識別子、および需要 s_i を集合 S に追加する。また、デバイスの提供状態と稼働状態を表す状態値の集合 $\mathbf{X} = \{X_{i,j}\}$ と集合 $\mathbf{Y} = \{Y_j\}$ を管理している。サービス i に対するデバイス j の提供状態値 $X_{i,j} \in \{true, false\}$ は、反応閾値モデルによって決定される。一方、デバイス j の稼働状態値 $Y_j \in \{true, false\}$ は $\forall_i \vee X_{i,j}$ によって与えられ、 $true$ の時に稼働状態、 $false$ の時には休止状態を表す。デバイス j について稼働状態値 Y_j が $true$ である端末をデバイス j の稼働端末または単に稼働端末と呼ぶ。さらに、端末は、閾値の集合 $\Theta = \{\theta_{i,j}\}$ を管理している。

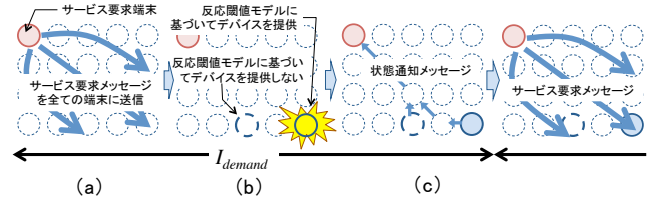


図1 提案手法の基本動作

3.2 端末の動作

サービスを要求する端末（以降、サービス要求端末と呼ぶ）は、サービスの内容を記述したサービス要求メッセージを、一定間隔 I_{demand} [s] ごとにフラッディングなどの機構を用いてネットワークに拡散する（図1(a)）。例えば、特定の座標のセンシング情報を I_{data} [s] (> 0) 毎に収集することを要求するサービスの場合には、 t 回目のサービス要求メッセージには、サービスの識別子 i 、サービスの需要 $s_i(t)$ 、観測したい地点の座標、センサの種類、情報収集間隔がサービス要求メッセージに含まれる。なお、例えば特定の領域の定時観測などのサービスの場合には、該当する領域にのみサービス要求メッセージを送信することでオーバーヘッドを抑えられる。

サービス i のサービス要求メッセージを受信した端末は、サービス要求を受信した隣接端末の識別子を記憶する。また、サービス i に対して自身の具備するいずれかのデバイスを提供することができるかどうかをセンシング対象の座標などから判断する。いずれかのデバイスを提供できる場合には、集合 S の需要 s_i として $s_i(t)$ を追加、または需要 s_i を $s_i(t)$ で更新する。端末は、デバイスごとに、集合 S に記録された需要 s_i にもとづき、3.3 節で述べる拡張反応閾値モデルを用いて提供状態値 $X_{i,j}$ と閾値 $\theta_{i,j}$ を求め、稼働状態値 Y_j を更新する（図1(b)）。さらに、サービス i に対するデバイス j の提供状態値 $X_{i,j}$ を状態通知メッセージよりサービス要求端末に通知する（図1(c)）。状態通知メッセージにはサービスの識別子 i が含まれ、サービス要求メッセージの経路を逆に辿るユニキャスト型のマルチホップ通信によってサービス要求端末に到達する。例えば、ある地点の温度観測のサービス要求に対して、その地点を観測可能な温度センサを有する端末は、観測するかどうかによらず、状態通知メッセージを送信する。これにより、サービス要求端末はサービス要求に対してデバイスを提供しうる端末の数と、デバイスを提供している端末の数を把握することができる。その後、サービス i に対してデバイス j を提供することを決定した端末、すなわち稼働端末は、サービス要求メッセージの情報にしたがい、例えば周期 I_{data} [s] ごとにセンシング情報を生成、送出する。なお、サービス要求メッセージを一定時間 E_i (> 0) 中に一度も受信しなかった場合には、端末はサービスが終了したものと判断し、集合 S, \mathbf{X}, Θ からサービス i に関する情報を削除し、稼働状態値 Y_j を更新する。

状態通知メッセージを受信したサービス要求端末は、3.4 節で述べるように、デバイスを提供できる端末数に対する稼働端末数の割合にもとづいて需要 s_i を更新し、 I_{demand} [s] 後に送信するサービス要求メッセージに含める。このようなサービス

要求メッセージによる需要の通知と、需要にもとづく状態変化、状態通知メッセージによる需要の更新を繰り返すことにより、端末の停止や観測条件の変化などに適応して、サービス要求に応じた適切な数の端末がデバイスを提供ようになる。

3.3 拡張反応閾値モデル

提案手法においては、あるサービスにデバイスを提供している端末が他のサービスにもデバイスを提供しやすくするよう、端末がサービス i に対してデバイス j を提供する確率 $P(X_{i,j} = false \rightarrow X_{i,j} = true)$ の定義式として、式 (1) に変数 $A_{i,j}$ (> 0) を加えた次式を用いる。

$$P(X_{i,j} = false \rightarrow X_{i,j} = true) = \frac{s_i^2}{s_i^2 + \theta_{i,j}^2 + A_{i,j}} \quad (5)$$

ここで、変数 s_i (≥ 0) は 3.4 章で述べる式によって求められるサービス i の需要を表し、変数 $\theta_{i,j}$ (> 0) はサービス i にデバイス j を提供する閾値を表す。

変数 $A_{i,j}$ の値はサービス i にデバイス j を提供することが望ましいほど小さくなるよう、デバイス j の提供状態の非効率さを表す変数 $l_{i,j}$ ($1 \geq l_{i,j} > 0$) を用いて次式で与えられる。

$$A_{i,j} = \frac{S_j}{1 + e^{-a \times (l_{i,j} - b)}} \quad (6)$$

ここで、変数 S_j は端末が受信したサービス要求メッセージのうち、デバイス j を提供できるものの数を表し、 $S_j = |\{X_{i,k} \in \mathbf{X} \mid k = j\}|$ で与えられる。定数 a (> 0) はシグモイド関数のゲインを表している。また、定数 b ($1 \geq b \geq 0$) を用いてシグモイド関数を右シフトすることで、変数 $A_{i,j}$ の値が 0 に近い値をとる変数 $l_{i,j}$ の値の範囲を調整することで、デバイスの提供状態がある程度非効率でもデバイスを提供する確率に影響を与えないようにできる。

提案手法においては、デバイスを提供できるサービス数 $S_j = |\{X_{i,k} \in \mathbf{X} \mid k = j\}|$ に対して、実際にデバイスを提供しているサービス数 $F_j = |\{X_{i,k} \in \mathbf{X} \mid k = j, X_{i,k} = true\}|$ が少ないほど、デバイスの提供状態は非効率であると考え、次式により変数 $l_{i,j}$ を大きくし、一方で、多くのサービスに対してデバイスを提供している端末を活用するように変数 $l_{i,j}$ を小さくする。

$$l_{i,j} = \begin{cases} 1 - \frac{F_j}{S_j - 1}, & \text{if } S_j > 1 \\ 1, & \text{if } S_j = 1 \end{cases} \quad (7)$$

式 (6) により、デバイスを提供できるサービス数が多いにも関わらず実際にデバイスを提供している数が少ない端末ほど、変数 $A_{i,j}$ が大きくなり、デバイスを提供する確率が小さくなる。

また、サービス i に対してデバイス j の提供を止める確率 $P(X_{i,j} = true \rightarrow X_{i,j} = false)$ の定義式として次式を用いる。

$$P(X_{i,j} = true \rightarrow X_{i,j} = false) = p_j \quad (8)$$

ここで、 p_j はデバイス毎に定められる定数である。

サービス i にデバイス j を提供する閾値 $\theta_{i,j}$ は次式に従い調整される。

$$\theta_{i,j} = \begin{cases} \theta_{i,j} - \xi_j, & \text{if device } j \text{ is allocated} \\ \theta_{i,j} + \varphi_j, & \text{if device } j \text{ isn't allocated} \end{cases} \quad (9)$$

表 1 パラメータ設定

需要の増加係数 δ_1	0.1
デバイスの提供を止める確率 p_1	0.01
分化の速度を定める係数 ξ_1	0.01
分化の可塑性を定める係数 φ_1	0.001
式 (6) のゲイン a	20
式 (6) の係数 b	0.8

ここで、係数 ξ_j 、係数 φ_j は 0 より大きい定数である。

3.4 サービスの需要

提案手法においては、サービス i の t 回目のサービス要求メッセージ送信時における需要 $s_i(t)$ を文献 [7] の基本式に基づいて次式で求める。なお、需要の初期値 $s_i(0)$ は 0 とする。

$$s_i(t+1) = s_i(t) + \delta_i - N_i/M_i \quad (10)$$

ここで定数 δ_i (> 0) はサービス i の需要の増加率を表す係数である。変数 M_i (> 0) は、 t 回目のサービス要求メッセージに対して受信した状態通知メッセージの総数を、変数 N_i は M_i 個の状態通知メッセージのうちサービス i の稼働端末から受信したものの総数を表す。

ただし、この方法では、状態通知メッセージの受信後に稼働端末が故障などによって減少しても、次の状態通知メッセージに含まれる需要に反映されない。そこで、例えばセンシング情報を I_{data} [s] $< I_{demand}$ [s] ごとに受信するサービスなど、サービス要求メッセージの送信間隔中に稼働端末からメッセージを受信する場合には、 I_{demand} 中に受信したセンシング情報のメッセージ数を N_i 、サービス要求に対してデバイスを提供しうる端末数 M_i に I_{demand}/I_{data} をかけたものをあらたな M_i として、式 (10) により需要を算出する。

4. シミュレーション評価

本章では、シミュレーションにより提案手法の有効性を評価する。

4.1 基本評価

100 [m] \times 100 [m] の領域に、領域の左下隅の座標 (0, 0) を起点として 10 [m] 間隔のグリッド状に 100 台の端末を配置した。全ての端末の通信範囲を、自身を中心とする半径 15 [m] の正円とする。全ての端末は、半径 20 [m] の正円内を観測できるデバイス 1 を 1 つずつ有しているものとする。反応閾値モデルのパラメータ設定を表 1 にまとめて示す。座標 (0, 0) に位置する端末が、サービス要求端末となり、1 [s] 毎に座標 (45, 45) の観測結果をサービス要求端末に通知するサービス要求メッセージを 10 [s] 毎に送信するものとした。なお、このサービスにデバイスを提供できる端末の数は 12 台である。以降では、500 回のシミュレーションで得られた平均値を示す。

シミュレーション開始時からの平均稼働端末数の変化、および、サービス要求端末で算出される需要の平均値の変化を図 2 に示す。図より、サービス要求開始後、稼働端末数が一時的に 3 台まで増加していることがわかる。これは、初期状態ではデバイスを提供している端末が存在しないために需要 s_1 が一時

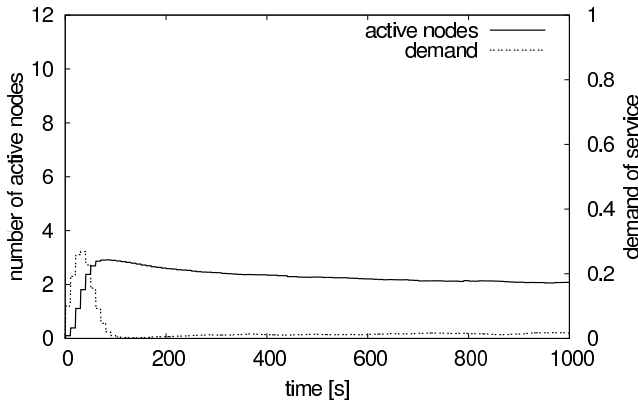


図2 稼働端末数と需要 s_1 の変化 (係数 $\delta_1 = 0.1$)

的に増加し、サービスにデバイスを提供できる多くの端末においてデバイスを提供する確率が高くなったためである。その後、これらの稼働端末により、需要 s_1 が減少する。その結果、式 (8) によりサービスの提供をやめた端末において、提供状態が再度 *true* になりにくくなるため、稼働端末数が減少する。サービスを提供可能な端末の台数である 12 に対して、需要の増加係数 δ_1 が 0.1 の場合、およそ 2 台の稼働端末で需要の増減のバランスが保たれるようになり、安定する。

4.2 需要の増加係数 δ_i の影響

安定時の稼働端末数は需要の増加係数の影響を受けるため、増加係数 δ_1 を 0.0 から 1.0 まで変化させた場合の、平均稼働端末数を図 3 に示す。図より、 δ_1 に対してほぼ線形に稼働端末数が増加していることがわかる。したがって、提案手法を用いて、望む台数の端末からデバイスの提供を受けるためには、サービス要求端末において需要の計算に用いる需要の増加係数 δ_1 を調整すればよいことがわかる。

4.3 サービス数の影響

提案手法では、反応閾値モデルに新たなパラメータ $A_{i,j}$ を追加することにより、サービス間でのデバイスの共有を図っている。そこで、座標 (0,0) に位置するサービス要求端末が、時刻 0 [s] から 5000[s] ごとに、同じ頻度、地点の観測の要求を、サービス識別子一つ加えた新たなサービス要求として順次追加するシミュレーション評価を行った。なお、サービス要求端末は、複数のサービス要求を一つのサービス要求メッセージとしてまとめて送信するものとした。図 4 には、横軸に 5000 [s] ごとに増えるサービスの数を、縦軸には新たなサービス追加直前における、稼働端末数をそれぞれ示している。なお、比較対象として、式 (5) よりパラメータ $A_{i,j}$ を除いた場合の結果を *normal* としてあわせて示す。

図 4 より、提案手法では、デバイスを要求するサービスの数によらず平均稼働端末数が変わらないことから、サービス間で効果的に同じデバイスが共有されていることがわかる。その結果、他の 10 台の端末はスリープ状態になって電力消費を抑制することができ、センサ・アクチュエータネットワーク全体の長寿命化が期待できる。一方、追加項を持たない反応閾値モデルでは、同じサービス要求端末からの同じ地点の観測要求であるにも関わらず、サービス数の増加に伴って観測を行う端末数

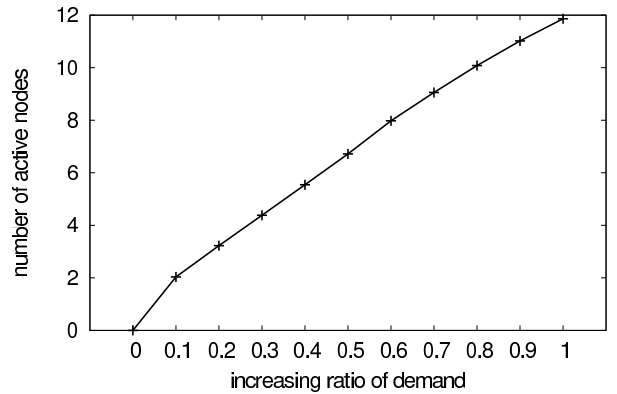


図3 需要の増加係数 δ_1 に対する平均稼働端末数

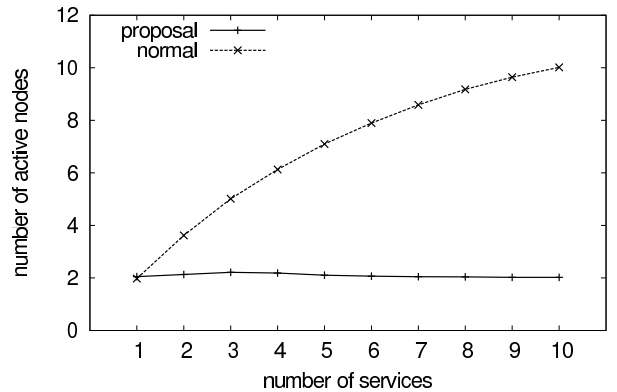


図4 サービス数に対する平均稼働端末数

が増えており、非効率的である。

4.4 比較評価

稼働端末数、端末あたりの平均送信メッセージ数の観点から Directed Diffusion [6] との比較評価を行う。なお、Directed Diffusion において *interest* メッセージを送信する端末を、提案手法と同様にサービス要求端末と呼ぶ。提案手法と Directed Diffusion のいずれについても、座標 (0,0), (0,90), (90,0), (90,90) に位置するサービス要求端末が、時刻 0 [s] からそれぞれ 5000 [s] ごとに、座標 (45,45) の観測結果を 1 [s] 毎にサービス要求端末に通知するサービス要求メッセージまたは *interest* メッセージを送信開始する。シミュレーション評価を 500 回行い、得られた結果の平均値を示す。

提案手法におけるパラメータ設定は 4.1 章と同じ値を用いる。Directed Diffusion においては、サービス要求端末は 10 [s] 毎に、センシング情報の送信間隔を 10 [s] よりも大きく設定した *interest* メッセージをフラッディングで送信し、最も早く受信した *exploratory data* メッセージの送信元端末に対してセンシング情報の送信間隔を 1 [s] とした *reinforce* メッセージを送信するものとする。

図 5 には、サービス数に対する稼働端末数の変化を、図 6 に時間に対する端末あたり平均送信メッセージ数の変化をそれぞれ示す。平均送信メッセージ数には、サービス要求メッセージ、状態通知メッセージ、*interest* / *exploratory data* / *reinforce* メッセージ、およびセンシング情報のメッセージが含まれる。なお、いずれの図においても、提案手法のシミュレーション結

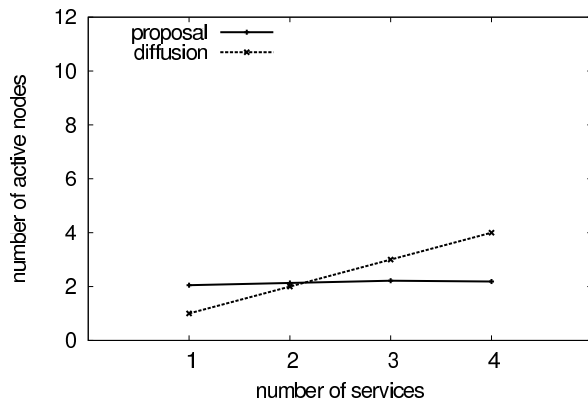


図5 稼働端末数の比較

果を *proposal*, Directed Diffusion の結果を *diffusion* としている。

図5より、提案手法においては、領域の4隅に位置するサービス要求端末に対して、平均2台の端末がデバイスを提供し、センシング情報を送信しており、サービス要求端末間で稼働端末を共有していることがわかる。一方、Directed Diffusionでは、interestメッセージを送信するサービス要求端末の増加にしたがって、稼働端末数が増えていることから、サービス要求端末ごとに異なる端末が稼働状態になっていることがわかる。これは、Directed Diffusionでは、サービス要求端末ごとに、最初に受信した *exploratory data* メッセージの送信元端末を選択するため、サービス要求端末に近いそれぞれ別の端末が稼働端末となりやすいことによる。一方、提案手法では、端末が自身のデバイス提供状態にもとづいて自律分散的に動作を決定しているため、サービス要求端末の台数や位置関係によらず、効率的に稼働端末が選出される。

また、図6から分かるように、提案手法により Directed Diffusionと比較してサービス数によらず平均メッセージ数がおよそ23%削減できることが分かる。なお、提案手法の方が稼働端末数の多いサービス数の場合でも送信メッセージ数が少ないのは、Directed Diffusionでは耐故障性のために interestメッセージ毎に *exploratory data* メッセージを送信するためである。

5. おわりに

本稿では、望むサービスネットワークを動的に構成するために、その場に配置された端末を、端末の持つデバイスの種別や稼働状態などを考慮して適切に組み合わせる、自律分散的な端末選出機構を提案し、シミュレーション評価によりその有効性を確認した。

提案手法では、反応閾値モデルにおいて個体が仕事に従事する確率を与える式に、端末の稼働状態を反映した指標を加えることで、サービス間の効率的な端末共有を実現している。同様に、電力消費や具備するデバイスの数などに関する指標を加えることにより、様々な条件を考慮したサービスへのデバイス提供を決定することができる。また、センサやアクチュエータといったデバイスだけでなく、稼働端末とサービス要求端末との

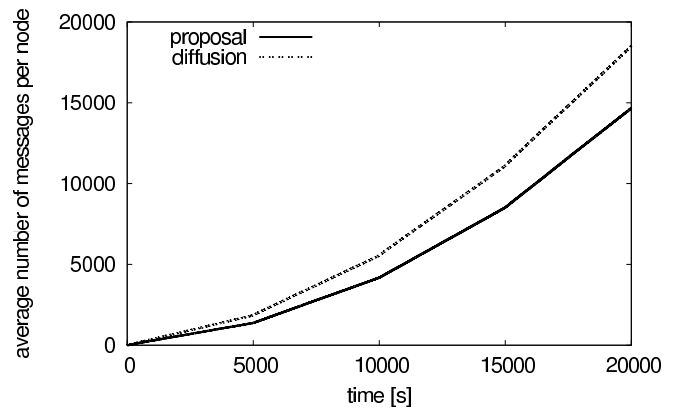


図6 端末あたり平均メッセージ数

間のメッセージ転送も端末が提供する機能の一つと見なすことにより、サービス間で経路を共有または隔離することにより、通信効率の向上を図ることができる。今後、これらの機能拡張に取り組み、その効果を検証するとともに、端末停止などに対する耐故障性についても評価する。

謝 辞

本研究の一部は、文部科学省グローバル COE プログラム (研究拠点形成費) および科学研究費補助金基盤研究 (B) 22300023 によるものである。ここに記して謝意を表す。

文 献

- [1] I. F. Akyildiz and I. H. Kasimoglu, "Wireless sensor and actor networks: research challenges," *Ad Hoc Networks*, vol. 2, pp. 351–367, May 2004.
- [2] E. Avilés-López and J. García-Macías, "TinySOA: a service-oriented architecture for wireless sensor networks," *Service Oriented Computing and Applications*, vol. 3, pp. 99–108, June 2009.
- [3] E. H. Jung and Y. J. Park, "TinyONet: A cache-based sensor network bridge enabling sensing data reusability and customized wireless sensor network services," *Sensors*, vol. 8, pp. 7930–7950, Dec. 2008.
- [4] H. M. N. D. Bandara, A. P. Jayasumana, and T. H. Illangasekare, "Cluster tree based self organization of virtual sensor networks," in *Proceedings of the International workshops on Wireless Mesh and Sensor Networks*, pp. 1–6, Nov. 2008.
- [5] C. Frank and K. Romer, "Algorithms for generic role assignment in wireless sensor networks," in *Proceedings of the International conference on Embedded Networked Sensor Systems*, pp. 230–242, Oct. 2005.
- [6] C. Intanagonwiwat, R. Govindan, and D. Estrin, "Directed diffusion: A scalable and robust communication paradigm for sensor networks," in *Proceedings of the International conference on Mobile Computing and Networking*, pp. 56–67, Aug. 2000.
- [7] E. Bonabeau, A. Sobkowski, G. Theraulaz, and J. L. Deneubourg, "Adaptive task allocation inspired by a model of division of labor in social insects," in *Proceedings of Biocomputing and Emergent Computation*, pp. 36–45, Jan. 1997.