

広域センサネットワークシステムにおける ノード間分散協調処理機構

森 香樹^{1,a)} 野口 拓^{2,b)}

概要：センサネットワークにおいて各センサノードの状況をネットワーク内で共有し、各ノードが協調して観測動作を適応的に制御することで、効率的な観測や、より多様なサービスを提供することが可能となる。しかし、ノード間で直接通信が行えないような広域にセンサノードが配置された環境ではノード間で協調処理させることが難しく、大規模なセンサネットワークにおいて協調動作を実現することが困難である。そこで本稿では、複数のセンサネットワーク間でインターネットを介してノード間を接続し、協調処理を可能とするためのフレームワークを提案する。提案フレームワークでは集中制御に依らない擬似的なP2Pネットワークを構成することで、複数のセンサネットワークにおける統一的な協調動作をサポート可能とする。

キーワード：広域センサネットワーク、協調処理、プラットフォーム

Distributed Cooperative Processing System between Nodes for Wide-Area Sensor Networks

MORI KOKI^{1,a)} NOGUCHI TAKU^{2,b)}

1. はじめに

近年の計算機や無線通信機器の小型化により、小型計算機にセンサと無線通信機器を取り付けそれらを相互に接続した、無線センサネットワークの利用が広がってきている。センサネットワークは主に環境観測等に利用され、無線でセンサノード間を接続することにより、通信インフラのない環境においても一定範囲内からのセンサデータ収集を可能にしている。従来のセンサネットワークは、センサノード間で直接通信が行えるような狭い範囲でのローカルなネットワークで構成されることが一般的であった。しかし近年では通信機器・インフラの発達により、あらゆるモ

ノがインターネットに繋がる「IoT」の考え方が広まってきた。これに伴い、従来よりもより広い範囲を対象とした環境情報の取得が求められるようになってきており、そのためのシステムの基盤技術の一つとしてセンサネットワーク技術の重要性は失われていないと考えられる。

一方で、センサネットワークにおける課題として、センサノード間の協調が指摘されている[2]。センサノードが観測結果に基づいて自律的に状況の判断を行い、その情報を他のノードと共有することにより、観測データの集約を行うホストによる集中管理に依らずにノード間での協調動作を実現することができる。センサネットワークにおいて協調動作をサポートすることで、他のノードの観測状況に適応して動作を最適化し観測動作を省電力化することや、ノードが自律的に他のノードと情報の共有・交換・要求を行うことにより従来よりも多様なサービスを提供することが可能になると考えられる。

しかし、センサデータを収集・利用する形態のシステム

¹ 立命館大学情報理工学研究科
Graduate School of Information Science and Engineering,
Ritsumeikan University

² 立命館大学情報理工学部
College of Information Science and Engineering,
Ritsumeikan University

a) kmori@nsl.ics.ritsumei.ac.jp

b) noguchi@is.ritsumei.ac.jp

の多くは、中央のサーバが全ての機器を管理する、集中制御型のシステムとして構築されていることが一般的である。集中制御システムはセンサノードなどの端末の数が増大すると制御遅延の発生など様々な問題が存在する。そのため、集中制御を行わない分散型のアーキテクチャにより構成することができれば有用であると考えられる。また一方で、前述のように広範囲からのデータ収集を目的としセンサノードを広域に分散して配置すると、ノード間で直接通信が行えないような環境での利用も想定される。このような環境では、ノード間における協調動作を実現する方法が課題となる。そこで本稿では、分散型アーキテクチャによる広域センサネットワークシステムを提案する。本システムは、ノード間における分散協調処理の仕組みと、この機構を広域センサネットワークシステムにおいて利用可能にするためのフレームワークの2つから成る。フレームワークは、ローカルセンサネットワークとインターネットに接続するゲートウェイ機能を持つシンクノード、ノード間のデータ中継のみを行うサーバにより構成される。サーバの役割をノード間のデータ中継のみに限定して設計することで擬似的なP2Pネットワークを構成し、ノードはサーバを意識せずに遠隔のセンサネットワークを利用できる。これにより、ノード上で動作するプログラムを主体としたアプリケーション開発を可能とする。また、ノード間の協調処理のために、各センサノードが観測結果とあらかじめ与えられた動作設定に基づいてイベントを生成し、これを他のノードに共有させる仕組みを設計する。本システムにより、離れたセンサネットワークが互いに統一的に扱えるようになり、大規模なセンサネットワークにおいて協調処理が可能となる。

2. 研究背景と関連分野

本章では、本研究の背景となる課題や技術について、既存研究とともに述べる。

2.1 広域センサネットワーク

広域センサネットワークという言葉は現在、特定の規模や構成を指しているものではなく、使用される場面や研究によって様々なネットワーク形態が存在する。その中で大別すると、2つの構成のタイプに分類できると考えている。ひとつは(1):従来よりも高出力な通信デバイスを使用したセンサネットワーク、もう一方は(2):複数の独立したセンサネットワークを統合して利用できるようにするもの、である。(1)はノードの配置できる規模が大きくなるが、構成としては一つのローカルネットワークという点で一般的なセンサネットワークと共通である。(2)は独立したローカルなネットワークが複数存在し、それらを統合して利用するためのシステムを設計・実装することに主眼が置かれている。本稿の研究で扱う広域センサネットワークは(2)に

分類され、ノード間で直接通信の不可能な配置を含むものを対象とする。ただし、それぞれのセンサネットワークは個別に運用されるものではなく、一つの大きなネットワークとして運用されるシステムを対象としている。

広域センサネットワークのためのプラットフォームの研究としては文献[1]がある。[1]では統合通信基盤としてJXTAを用いてP2Pネットワークを構築している。しかし、このシステムはサービスの利用者によるセンシングデータの利用を目的としており、センサノード間での通信は考慮されていない。本稿の提案システムでは、センサノード上のアプリケーションから直接他のノードへアクションを起こさせるような記述を可能とする。

2.2 センサネットワークにおける協調処理

センサネットワークにおいて、協調処理を行うことの有用性が指摘されている。ノードを主体とした協調処理の実現として、文献[2],[3]の研究がある。[2]では、センサノードが行うべき動作をルールとして定め、ルールを切り替えることにより適切な観測を行う仕組みを提案している。また、環境観測により自律的に状況を判断し、それをイベントとして他ノードに伝搬させることで、受け取ったノードがイベントに基づいてルールを切り替えることにより協調処理を実現する機構を提案している。ただし、[2]では各センサノードが高機能なデータベースを備えていることを前提としており、センサノードにも搭載可能な小容量・高機能なデータベース管理機構の作成も行われている。しかし、様々なデバイスがノードとして使用されることを考慮すると、これはあまり現実的でない。本稿では提案されている仕組みを参考にし、センサデータを蓄積せずともイベントの生成・伝搬を行える機能を設計している。また、[2]における協調処理は単一のローカルネットワークを対象としており、複数のセンサネットワークに跨った協調については考えられていない。本研究では複数のセンサネットワーク間で統一的に協調処理を可能とする機構を、インターネット上で動作するプラットフォームとローカルセンサネットワークにおいて実現している。

2.3 分散協調アーキテクチャ

システムの構成として、中央のサーバやホストが全ての端末の動作を管理・制御する集中制御型アーキテクチャと、各端末が独自に行うべき動作を判断し、実行する分散型アーキテクチャの2つがある。文献[4]はM2Mシステムを対象として、分散型アーキテクチャの有用性を述べている。M2Mシステムはサーバ集中型のアーキテクチャで構築されることが一般的であるが、ネットワーク負荷の増大や機器制御の遅延等の弊害がある。これらの問題の解決のために[4]ではM2Mゲートウェイがサーバと役割の分担を行うことにより、分散型アーキテクチャによるプラット

フォームを提案している。M2M ゲートウェイにおいてセンサデータを分析し、必要な精度でサーバに送信させることによりサーバ間での通信量を削減するイベント駆動データ収集と、あらかじめ定められたルールにより M2M ゲートウェイが自律的に機器を制御するルールベース自律分散機器制御により分散協調 M2M システムアーキテクチャが構成されている。一方本研究の提案システムでは、高機能なゲートウェイ機器を用いず、各センサノードが自律的に状態を共有することを実現しており、これによりノード自身の判断でセンサデータの送信量を調整することを可能とする。

3. 広域センサネットワークシステムにおけるノード間分散協調処理機構

本章では、本稿で提案する広域センサネットワークシステムにおけるノード間分散協調処理機構について、その課題や方式について述べる。

3.1 課題と要件

センサネットワークでは、センサノード同士が通信し協調動作を行うことにより、協調動作をサポートしない場合と比べ、効率的な観測動作を実現したり、より多様なサービスを提供することができる。しかし、センサノードが広範囲に配置され、それらが直接通信を行うことができない環境では、ノード間の協調が不可能となる。

この問題を解決する方法として、インターネットを通じて分散したセンサノード間で通信を行い、センサネットワークを構築する方法が考えられる。しかし、広範囲の様々なセンサからデータを収集するサービスは存在しているものの (Xively[5] など)、それらはノード間の通信について考慮されていない。そのため、インターネットを利用した、分散したセンサノード間の通信を行うためのシステムが必要になると考えられる。

このようなシステムを実現するためには、インターネットを利用して広範囲に分散したセンサノード間で通信を行うための方法が必要となる。多くの場合、センサノードは複数のノードによるローカルなセンサネットワークの一端末として配置されていると考えられる。そのため、分散したノード間の通信のためには、ローカルネットワーク内での通信と、分散したセンサネットワーク間の通信の両方における手順や制御を考えなければいけない。また、広域でのセンサネットワークにおいては、種類の異なる様々なセンサが用いられることが考えられるが、それらが共通のネットワークに参加でき、扱えることが必要となる。加えて、それほど高機能ではなく、機能や資源に制限のあるノードデバイスでも使用できるようなシステムであることが望ましい。すなわち、ユーザにシステムの存在を意識させないような設計とすべきである。

3.2 提案システムの概要

本システムは、センサネットワークが広域に分散し、異なるネットワーク内のノード間で直接通信が不可能な環境において、インターネットを通して仮想的な一つのセンサネットワークを構築し、協調処理を可能にするためのプラットフォームである。広域にまたがったセンサネットワークにおいて、任意のセンサノード間での通信を可能にすることで、中央サーバによる集中管理を行うことなく、広域センサネットワークにおける分散型協調処理を実現する。

提案システムではノード間の通信のために、インターネット上のサーバを使用する。図 1 に、離れたセンサネットワーク間で通信を行うシステムの概要を示す。システムはインターネットに接続可能なセンサノード (シンクノード) および通常のセンサノード (必須ではない)、インターネット上のサーバから構成される。サーバがインターネットを介してデータの中継を行うことで、遠隔のノード間での通信を実現する。

提案システムは協調処理の仕組みを設計したノード間分散協調処理機構と、それを広域センサネットワークにおいて実現するための、広域センサネットワークシステムのためのフレームワークの 2 つの柱で構成される。それぞれ、3.4 節と 3.5 節で述べる。

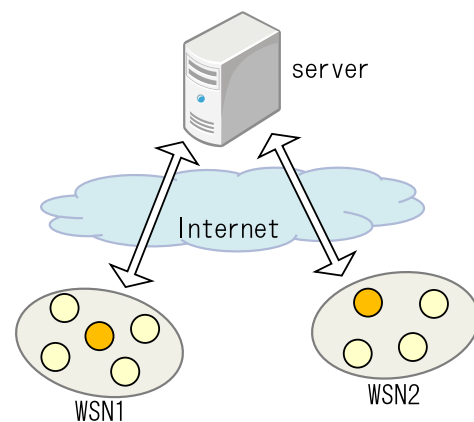


図 1 提案システムの概要

Fig. 1 Overview of proposal system.

3.3 想定利用シナリオ

河川監視システム

河川の氾濫を事前に予測するために、上流地域で雨量を観測し一定値を超えると下流の水位センサなどに通知し、アクションを起こさせる監視システム。

斜面防災システム

雨量センサと多くの地中水分センサからなる斜面防災のための観測システムにおいて、雨量センサが一定以上の雨を観測した場合に、それをイベントとして地中

水分センサに通知を行い、地中水分センサはセンシング間隔を短くして頻繁に土壌の状態を観測する。どちらの例も、種類の異なるノード間の情報共有によって観測動作の変更を行うことが共通している。またどちらも、雨量が多く危険性が高いと判断されるときのみ頻繁に観測を行い、そうでなければセンシングや送信の間隔を長くするという動作パターンで稼働させることで、運用時の省電力化が実現できる。

3.4 ノード間分散協調処理機構

本節では、提案システムを構成する2つの柱のうち、ノード間分散協調処理機構について述べる。

3.4.1 概要

ノード間分散協調処理機構は、センサノードが主体となって自律的に動作し、他のノードと情報を共有するといった協調動作を行うために必要な手順や、交換するメッセージの仕様などを設計したものである。本機構そのものの自体はこれまでに述べたような広域センサネットワークにそのまま適用できるものではなく、単一のローカルセンサネットワーク上での動作を想定している。

3.4.2 設計

本機構では、協調処理のために以下の2つを用いる。

- ユーザによるノードの動作設定
- イベントを用いた情報共有

図2に、ノード間分散協調処理機構の概要を示す。各センサノードの動作は、あらかじめセンサネットワークの管理者が設定を行う。動作は有限ステートマシンで表現される複数の状態パターンで構成し、設定された条件により状態遷移を行う。また、各ノードは他のノードとの情報共有のために「イベント」の送受信を行う。イベントは各ノードが設定された条件を満たしたときに生成し、他のノードへ伝搬する。イベントの生成条件は、センサの観測値が設定された閾値を超えた場合、といった形で設定する。イベントは「気温が25度を超えた」のように、発生した事象を記述する形で表現する。すなわち、例えばノードAが「ノードBは観測周期を変更せよ」といった、命令の形での記述は行わない。なぜならば、他のノードに対して命令を行うためには、対象のノードの動作設定を把握している必要があるためである。命令の記述を用いないことにより、ノード間の結合度を下げ、ノードの独立性を高めることでノードプログラムの設計が容易になる。

イベントを受信したノードは、その内容に応じて状態遷移を行い、動作を変更する。つまり、各ノードにおける他のノードに由来する状態遷移は、他のノードにおいて発生した事象に応じて行われる。このような機構により、他のノードにおける観測や動作に反応して適切な動作変更を行う、センサノード間の協調処理を可能とする。

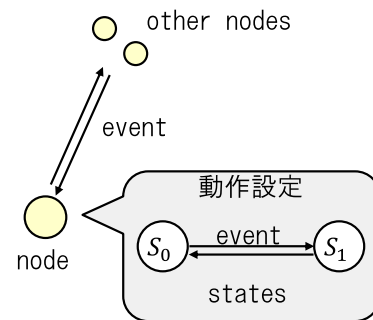


図2 ノード間分散協調処理機構の概要

Fig. 2 Overview of the distributed cooperative processing mechanism between nodes.

3.4.3 モデルによる解析

これまで述べてきた協調処理機構は、2.2節で述べたように[2]を参考にしたものである。しかし、[2]では提案されている協調処理の仕組み全体についての評価は行われていない。そこで本節では、提案機構を用いる汎用的なセンサネットワークシステムのモデルを考え、協調処理を行うことによる有用性の解析を行う。

汎用モデルを、以下の要素から構成するものとする。

- サーバ
- 2台以上のセンサノード

3.5節で後述するが、提案システムにおいて広域センサネットワークの構築にサーバを使用するため、解析モデルでもサーバの概念を用いる。サーバはノード間の仲介機能を提供し、各ノードは(直接的には)サーバに対してデータや要求を送信する。詳細は3.5節にて述べる。モデルではノードの数やそれぞれの動作にかかる消費電力などにパラメータを割り振り、その動作を解析する。

評価項目は一定時間システムを稼働させたときの各センサノードの消費電力とサーバの負荷とし、協調処理の有無それぞれのパターンで比較する。また、協調処理を行う場合については分散型と集中型の両アーキテクチャにおける実現を想定した比較も行う。以下に、それぞれのパターンにおける解析モデルを述べる。

協調有・分散型 サーバ1台、イベントを送信するセンサノード(S-Node)1台、イベントを受信し自身の動作を変更するアクチュエータノード(A-Node)1台で構成されるモデルを考える。S-Nodeは動作中にある確率でイベントAとイベントBを交互に生成し、送信する。A-Nodeは状態1と状態2の2つの状態を持ち、S-Nodeから受信したイベントにより遷移する。ただし、状態2の場合に状態1よりも動作にかかる消費電力が多いものとし、この状態となる時間の割合は状態1よりも少なくなるようにS-Nodeのイベント生成タイミングを設定する。これは、3.3節で示したような、稼働時間中の一部が消費電力の大きい動作状態となるようなアプリケーションを想定したものである。図3

に A-Node の動作を状態遷移図で示す．また，S-Node は常に同じ動作状態とする．

協調有・集中型 サーバ，ノードの構成は分散型と共通とし，A-Node の動作設定も同一とする．ただし，集中型アーキテクチャではサーバがノードの状態を全て管理し，制御する点を意識する必要がある．

協調無 サーバ，ノードの構成は共通であるが，A-Node は状態遷移を行わず常に同じ動作（状態 2）を続ける．協調処理を行わないため，A-Node は S-Node に影響を受けず，一定の頻度で動作する．

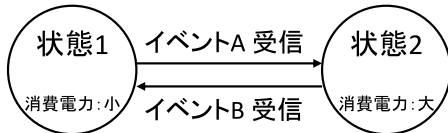


図 3 A-Node の状態遷移図

Fig. 3 State transition diagram of A-Node.

ここで，ノードの消費電力を $W_{\text{total}}[\text{J}]$ とし，以下の式で計算する．

$$W_{\text{total}} = (a(f_1 \cdot E_{t_1} + E_w) + (1-a)(f_2 \cdot E_{t_2} + E_w))T$$

この評価式に，表 1 に示すパラメータを設定する．ただし， T を総稼働時間， a を総稼働時間中に状態 1 となる時間の割合 (0-1)， E_w をスリープ中の単位時間あたり消費電力， f_1 ， f_2 を状態 1,2 における通信頻度， E_{t_1} ， E_{t_2} を状態 1,2 における一回の通信にかかる平均電力とする．なお，評

表 1 モデルのパラメータ

Table 1 Parameters of a model.

項目 [単位]	分散協調	集中協調	協調無
$T[\text{s}]$	6000000		
$E_w[\mu\text{W}]$	15		
$E_{t_1}[\text{mJ}]$	25		
$E_{t_2}[\text{mJ}]$	25		
a	1		
S-Node	-----		
$f_1[1/\text{s}]$	$\frac{1}{60}$		
$f_2[1/\text{s}]$	$\frac{1}{60}$		
A-Node	-----		
a	0.9	0.9	0
$f_1[1/\text{s}]$	$\frac{11}{600}$	$\frac{11}{600}$	$\frac{1}{10}$
$f_2[1/\text{s}]$	$\frac{7}{60}$	$\frac{7}{60}$	

価式では通信にかかる時間は無視している．また，通信頻度は提案機構に適用できるアプリケーションを想定して設定した．

評価式にパラメータを代入し計算した結果を図 4 に示す．

S-Node はいずれのパターンでも同じ動作であるため，同じ解析値となる．A-Node は協調を行わない場合，常に最も高い頻度で送信を続けるので消費電力が大きくなる．協

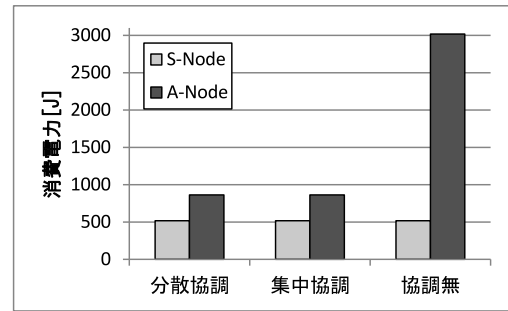


図 4 モデルによる解析結果

Fig. 4 Analysis result of a model.

調を行う場合，アーキテクチャによる違いは無い．ただし，この場合サーバの負荷は異なる．集中型アーキテクチャでは，ノードの動作状態は全てサーバが管理し，制御する．そのため集中型アーキテクチャの方が分散型に比べ処理が多く必要となるため，負荷が高くなると考えられる．ネットワークに参加するノードの数が大きく増えると，違いが明確に現れることが予想される．以上のことから，各センサノードの消費電力とサーバの負荷を合わせて考慮すると，分散型アーキテクチャにおいて協調処理を行った場合に最も効率的に稼働できると言える．

3.5 広域センサネットワークシステムのためのフレームワーク設計

3.4 節で述べた協調処理機構は，そのままでは広域センサネットワークで利用できない．そのため，提案機構を適用し，実現するためのプラットフォームが必要である．そこで本稿では，広域センサネットワークにおいて協調処理機構を適用するプラットフォームに必要な機能をまとめた，汎用的なフレームワークを提案する．

提案フレームワークは，サーバとローカルセンサネットワーク部から構成される．以下，本節ではそれぞれについて，協調処理を実現するために各機器が持つべき機能や通信の手順の指針を示す．

3.5.1 サーバ

本項では，提案機構を実現するプラットフォームにおいてサーバが持つべき機能について述べる．サーバに必要な機能は，以下の 3 つである．

- ノード情報の管理
- データ中継機構
- ノードへのインタフェース

以上の機能を加えたシステムの概要を，図 5 に示す．以下で機能の詳細を述べる．

ノード情報の管理

ノード間の中継を行うためには，ネットワークに参加しているノードの情報を把握する必要がある．センサネットワークシステムを最初に開始する際やノードを新たに追加する場合には，ノードアプリケーション上からサーバへ，

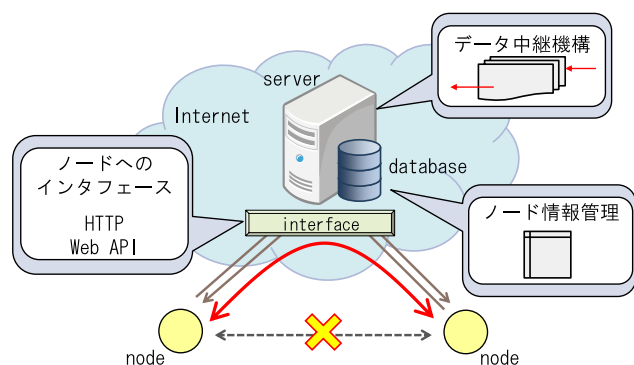


図 5 サーバの 3 つの機能
Fig. 5 Three functions of the server.

ネットワークへの参加要求を送信する。サーバはノードごとに識別番号を振り、ノードへ通知する。ノード情報の削除や更新にも対応すべきであり、簡易な DBMS を用いるとよい。また、各ノードの持つ属性によりグループ分けを行うことで、多数のノードが存在する環境での処理を容易にできる。グループ分けする属性には、ノードの配置地域や、動作しているアプリケーションの種類などが考えられる。グループの属性を表すメタ情報を管理すると、ノード上のアプリケーションやユーザからの検索要求に応答することができる。

データ中継機構

サーバにおけるノード間中継の実現方法は、センサノードに用いるハードウェアなどにより最適な実装手法が異なると考えられる。センサノードに様々なデバイスを使用することを想定し、機能や資源に制限のあるデバイスでも利用できる方法を採用すべきである。そのために、ノード間のデータのやり取りを非同期に行うこととする。ここで非同期とは、データの受信側ノードが、送信側ノードのデータを送信するタイミングを把握する必要がないことを表す。すなわち、ノードは任意のタイミングでデータを送信して構わない。送信されたデータを、中継先ノードへ確実に中継することがデータ中継機構の役割となる。

非同期なノード間の通信を実現するために、サーバは中継するデータを一時的に保管する必要がある。保管領域でのデータ管理にはいくつかの手法が考えられるが、単純なものでは、前述したノード情報を管理するデータベースに格納されたノードごとにデータ保管領域を割り当てるものが考えられる。ここで、保管領域を各ノードに対応するキュー構造とすることで、データの到着順に取り出して中継することができるため、管理が容易である。

ノードへのインタフェース

ノードはサーバの持つ機能をインターネットを介して利用するため、この機能をウェブサービスの一種と見なすことができる。そのため、サーバの機能を外部から利用可能とするためには、Web API を提供することが有用である。

センサノードに様々なデバイスが使用されることを想定し、Web API はどのようなデバイスからも容易に利用できる必要がある。

そのためには、RESTful な仕様の API が効果的である。RESTful API は同じく Web アプリケーションで利用される SOAP と比較して軽量であるため、様々なデバイスへ対応できる。

また、送受信されるセンサデータやイベントを統一的な表現で記述することで、多様なノードデバイス間やサーバにおけるデータの扱いを簡単にできる。データの統一的な記法には、従来 XML が多く用いられてきたが、より少ない記述量で表現できる形式である、JSON を用いることが好ましい。

なお、上記のような HTTP や JSON を用いる利点は、これらが広く一般に利用されており、情報の入手性が高いことやライブラリ等の開発資源が豊富なことによる実装の容易さにある。より効率のよいプロトコルやフォーマットを利用したり、独自に作成することも可能である。

3.5.2 ローカルネットワーク

ネットワーク構成

ローカルなセンサネットワークは各センサノードが無線で接続しネットワークを構成する。ローカルネットワークを構成する要素を、シンクノードとそれ以外の一般ノードの 2 種類のセンサノードとする。シンクノードは一般ノードに比べ高機能であり、インターネットに接続可能なセンサノードとする。シンクノードは自身がセンシング・外部との通信を行うとともに、一般ノードと外部との通信の中継を行うゲートウェイとしての役割を持つ。シンクノードは HTTP により外部と通信を行う一方、ローカルネットワーク内におけるセンサノード間の通信には ZigBee 等のセンサネットワーク向けプロトコルを用いる。一般ノードはシンクノードを通じて外部と通信する。なお、ネットワークを構成せずインターネット接続可能なノード 1 台のみでローカルセンサノードを考えることも可能である。

ノードの間欠動作

ノードは 3.4 節で述べたように、あらかじめ設定されたパターンに従い動作する。そのため、必要な周期で観測と通信を行い、それ以外の時間にノードをスリープさせることでノードの寿命を伸ばすことができる。しかし、間欠動作を行うにはノード間の通信のため、周囲のノードとの同期が必要である。しかし、イベントの送受信のために双方向での通信に対応したり、様々な動作周期を持つノードが混在する環境においては各々にとって最適な間欠動作の周期の決定が難しい。加えて、協調処理によるノードの動作の動的な変更への対応も考えなければいけない。また、ノードの搭載する通信デバイスによっては、ノードからの通信要求に対して時間制約の厳しい動作周期を守れない可能性もある。

以上のことから、全てのノードにおいて間欠動作を導入することは困難であると考えられる。具体的には、以下のような環境・ノードは間欠動作が難しいと言える。

- ノード数が多いネットワーク
- 動作変更が頻繁に起こる
- 動作周期の異なるノードの種類が多い
- 接続（中継する）ノードが多いノード
- 通信デバイスの制御性が低いノード

一方、通信デバイスの問題を除いて、間欠動作を行うノードの数を限定することで問題を簡単にすることができる。ネットワークのトポロジにおいて、一つのノードのみと接続されているような末端ノードは、間欠動作の導入が比較的容易である。そのため、間欠動作を導入するためには末端ノードが多くなるようにネットワークを構成することが望ましい。ただし、特定のノードに多くの末端ノードが接続される場合には各末端ノードの間欠周期が重複しないように工夫が必要である。

3.5.3 サーバとローカルネットワーク間の通信

シンクノードとサーバ間の通信

シンクノードから中継用データを受信したサーバは、直ちに送信先ノードへ転送を行うことが理想的である。しかし、サーバに用いられるプラットフォームの多くはクライアントからのアクセスに応答するものとして設計がなされており、それらを利用する開発者には、リアルタイムな通信を実現することは困難である。そこで提案フレームワークでは、従来より用いられてきた非同期双方向通信を実現する仕組みである、ポーリングを用いる。ノードが定期的に自身宛のデータや要求が無いかをサーバに問い合わせることで、非同期なデータの受け渡しを実現することができる。

疑似的な P2P ネットワークの構成

ノードが外部の端末に対してデータを送信する際、ノード上のアプリケーションとして記述する開発者にとって、自身のノードからシンクノード、サーバ、目的の送信先ノードへの通信経路という低レイヤの制御を考えることは負担となる。従って、通信経路に関わる処理を隠匿し、アプリケーション開発者に経路途中のノードやサーバを意識させないようなプラットフォームとすべきである。そのため、これらの処理を隠匿するライブラリを作成し、アプリケーション上から経路を意識せずに目的のノードまでのデータ送信を実現できるとよい。サーバとシンクノードをデータ中継に特化して設計していることに加え、アプリケーション開発者が通信経路を意識しなくてもよいプラットフォームとすることで、広域なセンサネットワークにまたがる疑似的な P2P ネットワークを構成することができる。

4. 実装と評価

これまでに述べた広域センサネットワークプラットフォーム

と、その上に適用するノード間分散協調処理機構が実装可能であるかを検証するため、実機を用いたプロトタイプ実装を行った。また、協調処理による効果を実験により評価した。本章では実装と評価結果について述べる。

4.1 実装環境

プロトタイプシステムは、研究室内で構築した。サーバの構築には Node.js を使用し、センサノードの実機プロトタイプには Arduino Uno を用いた。センシングデータの保存には micro SD カードを使用した。通信デバイスとして、ローカルネットワーク部には XBee Series2 を、またシンクノードにおけるインターネットの接続には有線 LAN を使用し、プロトコルには HTTP を用いた。シンクノードとなる Arduino には、Ethernet シールドとワイヤレス SD シールドを重ねることで LAN と XBee の同時使用を可能にしている。図 6 に、作成したシンクノードを示す。また、将来的に多数のセンサノードが配備された環境における評価を行うことを想定し、PC 上で動作する擬似ノードプログラムを作成した。



図 6 Arduino によるシンクノード

Fig. 6 Sink node using Arduino.

4.2 プロトタイプ実装

3 章で述べた提案機構について、必要な機能が実現可能であるかを確認するための実装を行った。

プロトタイプシステムにおいて、実装可能であることを確認した機能を以下に示す。

- サーバ
 - RESTful API によるインタフェース
 - ノード情報の登録
 - 中継データの一時保管
 - ポーリングを受けてデータ中継
- シンクノード
 - XBee ↔ Ethernet のデータ中継
- 一般ノード
 - データの POST, ポーリング, 受信
 - 受信したイベントによる状態遷移
 - ライブラリによるプラットフォーム機能の隠匿
 - スリープ (Arduino)

以上の機能から、直接通信できないセンサノード間で非同期に通信を行い、イベントの送受信により協調処理が可能であることを実装により確認した。

4.3 評価

3.4.3 項では、協調処理の有用性を解析評価した。本節ではプロトタイプシステムを使用し、実装されたシステムにおいても協調処理が効果的であるか評価する。

評価環境として、以下のようなアプリケーションを想定し実装した。

- 防災システムを想定した、雨量を計測するノードと河川の水位を計測するノードから構成される広域センサネットワーク
 - 協調処理
 - 雨量センサノードが降雨を検知すると rain イベントを、降雨がなければ fine イベントを発行する
 - 水位計ノードは動作周期ごとにポーリングとセンサの値を送信する。rain イベントを受信すると動作周期を 5 秒に、fine イベントを受信すると 60 秒に変更する。
- なお、これは分散型アーキテクチャで協調処理を行うパターンでの動作設定である。集中型の場合は雨量センサノードはセンサの値をそのまま送信し、サーバにてセンサ値からイベントを生成する。協調処理無しでは水位計ノードは常に動作周期 5 秒となる。

評価実験は、以下の 2 つを比較対象とした。

(1) 協調処理有り、分散型/集中型

(2) 協調処理有り（スリープ動作含む）/協調処理無し
それぞれの環境において、アクチュエータノードとなる Arduino を電池で稼働させ、電圧推移を測定した。また、電圧値が急激に下がり始めると動作が不安定になると考えられるため、稼働していると見なす電圧の下限値を 4200mV とした。なお、Arduino の電圧測定にはサイト [6] を参考にした。

(1) について、分散型、集中型の両アーキテクチャの実装ごとに 3 回実験を行った結果、平均稼働時間がそれぞれ約 222 分、223 分であった。これにより、アクチュエータノードにおける両アーキテクチャによる消費電力はほぼ同じであり、解析モデルによる評価と一致した。また、(2) の結果を図 7 に示す。

協調処理有りで、動作周期 60 秒のとき、そのうち 55 秒間は Arduino をスリープさせている。協調処理の有無ごとに 3 回実験を行い、平均稼働時間がそれぞれ約 273 分、219 分となった。協調処理無しの場合に (1) の結果とあまり差が出なかったことから、XBee は受信待機時と実際に送受信を行っている時の消費電力の差が小さいと考えられる。そのため (2) の実験では協調処理の効果をより明確にするため XBee のスリープを試行したが、実装上の問題があり Arduino のスリープで代用とした。今回の結果から、

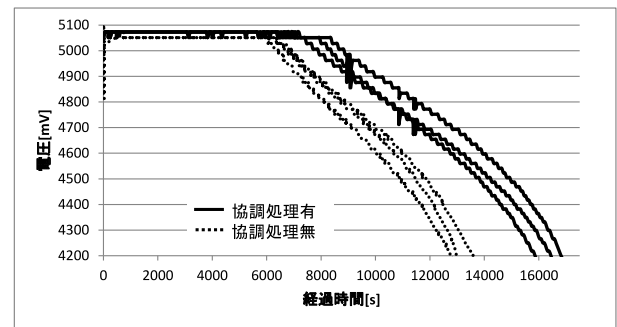


図 7 A-Node の電圧推移

Fig. 7 Voltage transition of A-Node.

協調処理によって適応的に動作変更し、センシングやデータ送信の頻度を下げてノードの休止時間を取ることで、電力消費量を削減し長寿命化が実現できることが示された。

5. おわりに

本稿では、広域センサネットワークにおける協調処理の仕組みをまとめたノード間分散協調処理機構と、それを適用可能なプラットフォームの汎用的な機能を示した広域センサネットワークシステムのためのフレームワークの 2 つを提案した。また、プロトタイプ実装により提案フレームワークと機構が実現可能であることを示した。さらに、協調処理による効果を評価するために比較実験を行い、協調処理によって稼働時間が最適化され、長寿命化を実現できることを確認した。

今回は研究室内における、比較的単純な構成によるプロトタイプシステムを用いた評価を行った。しかし、実際に広域にセンサノードを配置し、インターネット上に設置したサーバによりシステムを構成し、実際の運用に近い形での評価が必要である。

謝辞 本研究は JSPS 科研費 26289122, 15K0014 の助成を受けたものである。

参考文献

- [1] 磯村学, 井戸上彰, 堀内浩規: 広域に分散するセンサネットワークのための統合通信基盤の提案, 情報処理学会論文誌, vol.49, No.6, pp.1803-1818(2008).
- [2] 富森英生, 横田裕介, 大久保英嗣: ルールベースの問合せ処理機構による協調型センサネットワークの実現, 情報処理学会研究報告, MBL, vol.2009, No.8, pp.57-64(2009).
- [3] 横田裕介: データベースアプローチによる協調型センサネットワークシステムの実現, システム/制御/情報: システム制御情報学会誌, vol.54, No.12, pp.457-464(2010).
- [4] 北上真二, 他: 分散協調 M2M システムアーキテクチャの提案, 情報処理学会第 76 回全国大会講演論文集, vol.2014, No.1, pp.7-8(2014).
- [5] Xively(online), 入手先 < <https://www.xively.com/> > (2016.01.18).
- [6] Secret Arduino Voltmeter - Measure Battery Voltage(online), 入手先 < <https://provideyourown.com/2012/secret-arduino-voltmeter-measure-battery-voltage/> > (2016.01.31).

正誤表

ページ	該当箇所	誤	正
p.5	表 1 $T[s]$	6000000	1200000
p.6	3.5.2 項 11 行目	ZIgBee	ZigBee