

# センサネットワークにおけるゲートウェイサービスの構成

市岡 怜也<sup>†</sup> 鈴木 和久<sup>†</sup> 毛利 公一<sup>††</sup> 大久保 英嗣<sup>††</sup>

<sup>†</sup>立命館大学大学院理工学研究科 <sup>††</sup>立命館大学情報理工学部

近年、センサノードと呼ばれる無線小型端末を用いてネットワークを構成するセンサネットワークが注目されている。センサネットワークを用いることで、広域の環境情報を取得することができる。我々は、センサネットワークの適用範囲を広げるために、センサノードが移動し、異種のセンサノードが混在する環境を想定している。このような環境では、センサネットワークの構成が変化し、機能・性能が異なるセンサノードが混在するため、アプリケーションの構築が困難となる。そのため、本研究では、異種のセンサノードが混在するセンサネットワーク構築を支援するシステムプラットフォームであるゲートウェイサービスを構築している。本論文では、センサネットワークにおけるゲートウェイサービスの構成、実装、性能について述べる。

## A Construction of Gateway Service in Sensor Networks

RYOYA ICHIOKA<sup>†</sup> KAZUHISA SUZUKI<sup>†</sup> KOICHI MOURI<sup>††</sup> EIJI OKUBO<sup>††</sup>

<sup>†</sup>Graduate School of Science and Engineering, Ritsumeikan University

<sup>††</sup>College of Information Science and Engineering, Ritsumeikan University

Recently, the sensor network where the networks is composed with wireless small terminals that are called sensor nodes is actively researched. Environmental information on a wide area can be easily acquired by using a sensor network. In order to expand the range of the application on the sensor network, we assume the environment where each sensor node moves between the networks and the different kind of sensor nodes exist together. In such an environment, since the composition of the sensor network changes, the construction of applications becomes difficult. Therefore, we have been developing a gateway service which is the system platform for supporting the heterogeneous sensor network construction. In this paper, the design, implementation and performance evaluation of the gateway service in sensor networks are described.

### 1 はじめに

近年、無線通信技術の発達やコンピュータの小型化が進んでいる。これらの技術を用いて、無線通信機能、計算機能、温度や湿度などを取得するセンシング機能を持つ小型のセンサノードが開発されている。センサノードを複数用いてネットワークを構成するセンサネットワークが注目され、さまざまな研究が行われている。また、センサノードを広域に配置してセンサネットワークを構築することにより、温度、照度、湿度などの環境情報を取得することが可能となる。取得し

た情報は、人や環境の状況を判断するために利用したり、センサデータを利用するアプリケーションに提供することができる。このような特徴から、部屋、車、街頭などにセンサネットワークを構築し、それを利用するアプリケーションの実現が検討されている。

従来のセンサネットワークでは、センサノードが複数のセンサネットワーク間を移動することや、さまざまな種類のセンサノードが混在する環境が実現されていない。これらを実現することにより、センサネットワークの適用範囲を広げることができると考えられる。具体的には、センサノードの移動により、センサ

ネットワークに固定配置されているセンサノードから取得できる情報と、人や物に取りつけられた移動するセンサノードから取得できる情報を組み合わせて利用するサービスが構築できる。また、センサノードの混在により、観測対象ごとにセンサノードを使い分けることや、センサノードの種類に依存しないセンサネットワークの構築が可能となり、センサネットワークの適用範囲が拡大する。しかし、このようなセンサネットワークでは、センサネットワークの構成が変化するため、センサネットワークを利用するアプリケーションに負担がかかる。具体的には、全ての種類のセンサノードの情報を把握しなければならない点や、センサネットワークを利用する機能をアプリケーションごとに実装する必要があるといった点が挙げられる。

以上のような背景から、本研究では、センサネットワークの構築を支援するソフトウェアプラットフォームであるゲートウェイサービスを構築している。ゲートウェイサービスは、センサネットワークを利用するアプリケーションに対して、センサネットワークを利用するためのインタフェースを提供する。ゲートウェイサービスが提供するインタフェースを用いることで、センサネットワークを利用するアプリケーションを容易に実装することが可能となる。

以下、2章で本研究の想定環境について述べ、3章で異なるセンサノードが混在するセンサネットワークの適用例について述べる。4章でゲートウェイサービスの設計方針について述べ、5章でゲートウェイサービスの構成について述べ、6章で実装と評価について述べる。

## 2 想定環境

我々は、センサネットワークの適用範囲を広げるために、センサネットワーク間をセンサノードが移動し、さまざまな種類のセンサノードが混在する環境を想定している。センサノードの移動とは、センサノードがセンサネットワークに参加・離脱することを意味する。具体的には、観測対象である人や物にセンサノードを取り付け、観測対象が移動することである。これにより、センサネットワークに配置されているセンサノードから取得できる情報と、人や物の状態を示す情報を組み合わせて利用するサービスが構築できる。また、センサノードの混在とは、センサネットワークにさまざまな種類のセンサノードが参加していることを意味する。具体的には、センサノード上に実装されているセンサデバイスや通信デバイスが異なるセンサノード

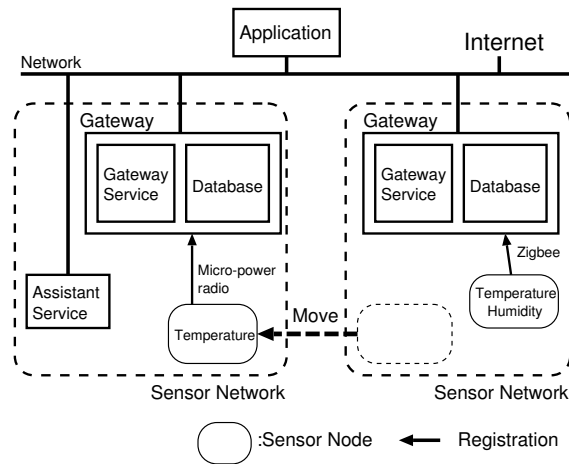


図1 想定環境

が混在することである。これにより、(1) 観測対象ごとにセンサノードを使い分ける、(2) センサノードに依存しないセンサネットワークの構築が可能となる、(3) 種類の異なるセンサデータを組み合わせて利用することが可能となるといった利点があり、センサネットワークの適用範囲が拡大する。

センサネットワークの構成要素として、ゲートウェイ、センサノード、アプリケーション、アシスタントサービスがある(図1参照)。

- センサノード

温度、湿度、照度などの値を取得するセンシング機能と、他のセンサノードと通信する機能を備える小型機器である。

- ゲートウェイ

ゲートウェイは、ゲートウェイサービスとデータベースを備える。ゲートウェイサービスは、センサネットワークを管理し、アプリケーションに対してセンサネットワークを利用する機能を提供する。データベースには、過去のセンサデータが格納されており、アプリケーションが過去のセンサデータを利用する際に参照する。

- アプリケーション

センサネットワークを利用したサービスを実現するソフトウェアである。ゲートウェイサービスが提供する機能を用いて実装される。

- アシスタントサービス

ゲートウェイサービスの機能を補助するアプリケーションである。位置情報などのセンサノードに関するプロパティを更新する。

### 3 適用例

センサノードの移動とセンサノードの混在環境をゲートウェイサービスを用いて実現することにより、新しいサービスが提案できると考えている。以下に、ゲートウェイサービスを利用した適用例を挙げる。

**空調管理サービス** 体温と位置を測定するセンサを身につけた人が、温度センサが配置された部屋に入る。そして、部屋にある複数の空調を管理している空調管理サービスが、人と部屋のセンサからデータを受け取る。空調管理サービスは、受け取ったデータから各空調の設定温度を決定する。空調管理サービスは、体温の高い人の周りにある空調のみ他の空調より設定温度を低く設定することが可能となる。

**物流管理システム** 食品を配送するかごにセンサノードを取り付け、状態を観測する。これにより、品質管理や配送路の見直しが可能となる。センサノードの混在により、観測対象によって適したセンサを用いることができ、詳細なデータの取得やコストの軽減が可能となる。

## 4 設計方針

### 4.1 解決すべき課題

センサノードを構成する通信デバイスとセンサデバイスには、数多くの種類がある。例えば、センサノードに備えられているセンサデバイスが異なる場合や、温度や湿度などの同じ種類のセンサデータを取得するセンサデバイスでも異なるデバイスが実装されている場合がある。このため、センサデバイスから取得した電圧値をセンサデータに変換する変換式やセンサデータの形式が異なる。また、通信デバイスは、zigbee や微弱無線など各センサノードで異なる通信デバイスが実装されており、通信プロトコルが異なる。センサノードの違いをアプリケーションが対応するためには、すべてのセンサノードの情報や機能を保持する必要があるが、困難である。そのため、ゲートウェイサービスは、センサノードの違いを隠蔽する必要がある。

センサノードの移動により、センサノードの位置や数といったセンサネットワークの構成が変化する。これらの変化は、アプリケーションの振舞いを変化さ

せる引き金として利用される場合がある。そのため、ゲートウェイサービスは、アプリケーションがセンサネットワークの変化を感知する機能が必要となる。

### 4.2 ゲートウェイサービスが提供するインタフェース

ゲートウェイサービスは、アプリケーションに対してセンサネットワークを利用するためのインタフェースを提供する。センサネットワークの適用場面として、移動端末上のアプリケーションが利用する場合と固定端末上のアプリケーションが利用する場合が考えられる。そのため、ゲートウェイサービスは、利用用途に応じて移動端末上で動作するアプリケーションと固定端末上で動作するアプリケーション用の2種類のインタフェースを提供する。前者は、アプリケーションがゲートウェイサービスを知らなくてもよいという利点があるが、同一セグメントのネットワークに存在するゲートウェイサービスのみ利用できる。後者は、違うネットワーク間でゲートウェイサービスを利用可能であるという利点があるが、あらかじめゲートウェイサービス、もしくはゲートウェイサービスを管理するディレクトリサーバの情報を知る必要がある。

アプリケーションがゲートウェイサービスを利用してセンサネットワークからセンサデータを取得する場合、センサデータの種類、センサノードの位置情報、取得周期、取得期間、センサデータの取得時間を指定してゲートウェイサービスにセンサデータ取得を要求する。センサネットワークを利用するアプリケーションがセンサデータを取得する場合、センサデータの発生源を特定することが重要なのではなく、取得したいセンサデータを特定することが重要である。そのため、ゲートウェイサービスが提供するインタフェースでは、センサノードを特定する情報を指定するのではなく、センサデータを特定する情報を指定させる。

### 4.3 センサノードの違いの隠蔽化

ゲートウェイサービスは、各センサノードに対応した情報や機能が含まれたオブジェクトを、統一したインタフェースとしてアプリケーションに対して提供する。ゲートウェイサービスが提供するインタフェースは、センサノードに依存しない。これにより、ゲートウェイサービスは、アプリケーションに対して、センサノードの違いを隠蔽する。

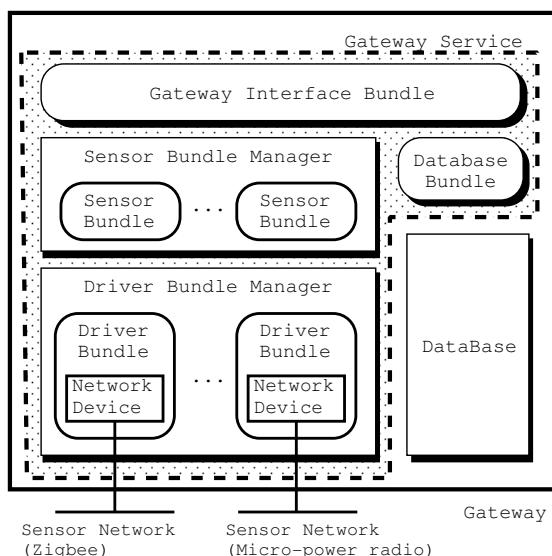


図2 ゲートウェイサービスの全体構成

#### 4.4 センサネットワークに発生する変化の通知機能

ゲートウェイサービスは、センサネットワークに発生する変化を感知し、アプリケーションに通知する。センサノードに位置情報や電力残量などのプロパティを設定し、プロパティの変化からセンサネットワークの変化を感知する。

### 5 ゲートウェイサービスの構成

#### 5.1 概要

以上の設計方針を基本に、ゲートウェイサービスを設計した。ゲートウェイサービスの全体構成を図2に示す。ゲートウェイサービスは、Driver Bundle、Sensor Bundle、Database Bundle、Gateway Interface Bundleから構成されている。

#### 5.2 Driver Bundle

ゲートウェイは、センサノードと通信するために、Zigbee や微弱無線などの通信デバイスを持つ。通信デバイスを制御するために、ゲートウェイサービスは通信デバイスと1対1に対応するDriver Bundleを持つ。Driver Bundleは、センサノードが用いる通信デバイスを抽象化したものであり、センサノードが持つ通信デバイスの違いを隠蔽する。また、センサノードの通信デバイスの種類に関するプロパティと、Network Deviceを用いたセンサノードとの通信機能を実現するメソッドを持つ。Driver Bundleは、Driver Bundle

Managerによって管理され、Driver Bundle ManagerがDriver Bundleを追加・削除する。

#### 5.3 Sensor Bundle

Sensor Bundleは、センサノードが備えるセンサデバイスの違いをアプリケーションに対して隠蔽する。このため、1種類のセンサノードに対して1つのSensor Bundleが定義される。Sensor Bundleは、センサノードのプロパティとセンサデータを取得するためのメソッドが実装されている。メソッドは、センサノードからセンサデータを取得する機能や、センサデータを変換する機能など実装されている。また、センサノードのID、センサノードが備えるセンサデバイスの種類、センサノードの位置情報といったセンサノードに関する情報をプロパティとして持つ。センサノードとの通信はDriver Bundleによって実現される。このため、Sensor Bundleのプロパティとメソッドを用いて通信に利用すべきSensor Bundleを特定する。Sensor Bundleは、この処理によって関連づけられたDriver Bundleを用いて、センサノードからセンサデータを取得する。Sensor Bundleは、Sensor Bundle Managerによって管理され、Sensor Bundle ManagerがSensor Bundleを追加・削除する。また、Sensor Bundleのプロパティを管理し、プロパティの変化からセンサネットワークの変化を検出する。

#### 5.4 Database Bundle

ゲートウェイは、過去に取得したセンサデータを格納するデータベースを持つ。Database Bundleは、このデータベースを管理する。また、Sensor Bundle Managerから通知されるセンサノードの参加・離脱といったセンサネットワークの変化を基に、データベースの内容を更新する。データベースに対応するDatabase Bundleを用いることにより、アプリケーションに対してデータベースの違いを隠蔽することが可能となり、センサネットワークでさまざまなデータベースを適用することができる。

#### 5.5 Gateway Interface Bundle

Gateway Interface Bundleは、アプリケーションに対してゲートウェイサービスを利用するための機能を提供する。Gateway Interface Bundleがアプリケーションに対して提供する機能を以下に示す。

- Sensor Data Acquisition  
センサデータを取得する機能を提供する。
- Discovery

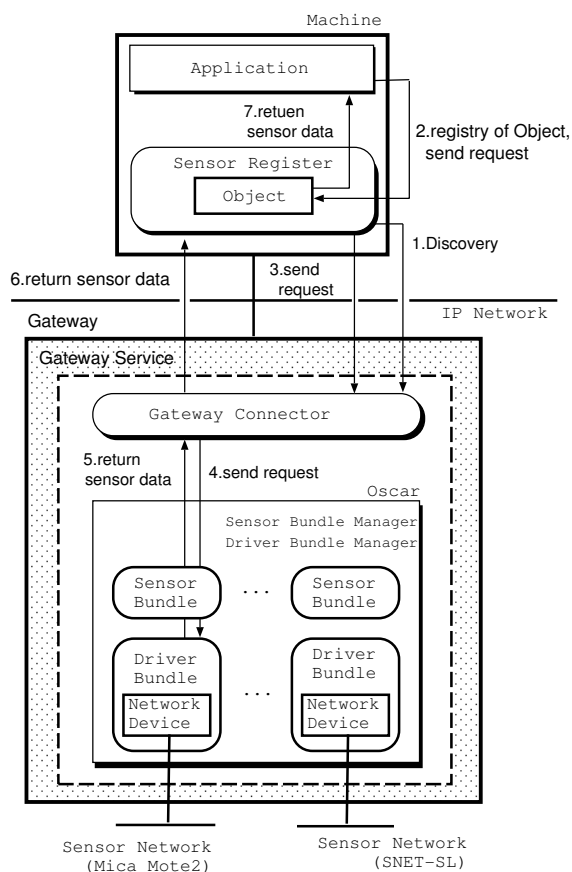


図3 ゲートウェイサービスとアプリケーションの実装

ゲートウェイサービスを発見する機能を提供する。

- Registry

アプリケーションは、定期的にセンサデータを取得するために、ゲートウェイサービスにオブジェクトを登録する必要がある。Registryは、このオブジェクトを登録する機能を提供する。ゲートウェイサービスは、登録されたオブジェクトのメソッドを周期的に呼び出す。これにより、アプリケーションは定期的にデータを取得することができる。

- Property Control

Sensor Bundleのプロパティを追加・削除・更新する機能を提供する。

## 6 実装

前章までに述べたゲートウェイサービス、およびゲートウェイサービスを利用するアプリケーションを、Java (J2SE version 1.4.2.02) を用いて実装した(図3参照)。特に、センサノードの1つである Mica

Mote2 [2] と三菱電機が開発している SNET-SL を対象として実装した。また、Mica Mote2 と SNET-SL は、温度や照度などが取得可能なセンサデバイス、および計算処理機能と無線通信機能を備える小型端末である。

Sensor Bundle Manager と Driver Bundle Manager の実装は、OSGi (Open Service Gateway interface) [1] フレームワークを実装した Oscar [3] を用いた。Mica Mote2 と SNET-SL に対応した Driver Bundle と Sensor Bundle は、Java を用いて実装した。また、Driver Bundle での Mica Mote2 との通信処理は、TinyDB [4] を用いて実装した。

Gateway Interface Bundle の機能を実現するために、Sensor Register と Gateway Connector の2つのモジュールで実現されている。これらも Java を用いて実装した。Sensor Register は、アプリケーションから利用されるライブラリであり、アプリケーションが動作する計算機上で実行される。Gateway Connector は、ゲートウェイ上で実行され、Sensor Register からの接続要求に応答する。Sensor Register と Gateway Connector により、アプリケーションはセンサデータを取得することができる。この処理手順を以下に示す。

1. アプリケーションは、ディスカバリ機能を用いてゲートウェイサービスを発見する。
2. ゲートウェイサービスを発見後、アプリケーションは、Sensor Register に対してセンサデータの取得要求とオブジェクトの登録を行う。
3. Sensor Register は、アプリケーションのセンサデータの取得要求を Gateway Connector に通知する。
4. Sensor Connector は、アプリケーションの要求を満たす Sensor Bundle を検索し、Sensor Bundle のメソッドを呼び出す。Sensor Bundle は、自身を利用する Driver Bundle を検索し、Driver Bundle のメソッドを用いてセンサノードにデータの取得を要求する。
5. Driver Bundle は、センサノードからセンサデータを取得した場合、Sensor Bundle にセンサデータを返す。Sensor Bundle は、取得したセンサデータを変換し、それを Gateway Connector に返す。
6. Gateway Connector は、取得したセンサデータを Sensor Register に通知する。
7. Sensor Register がセンサデータを取得した場合、アプリケーションが登録したオブジェクトのメ

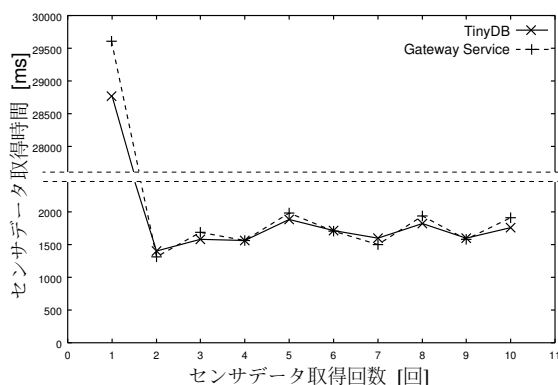


図4 データ取得時間

ソッドを呼び出す。これにより、アプリケーションは定期的にセンサデータを取得することができる。

## 7 評価

実験では、アプリケーションがゲートウェイサービスを用いて温度データを周期 2048ms 以下で 10 回取得する場合について、データ取得要求から取得までの時間の平均値を計測した。また、ゲートウェイサービスの性能は Sensor Bundle, Driver Bundle に依存するため、Driver Bundle の実装に用いた TinyDB の性能を計測し、比較を行った。ゲートウェイサービスは、Intel Pentium III プロセッサ 500MHz, 448MB のメモリを持つ IBM PC/AT 互換機上で動作させている。

実験結果を、図 4 に示す。TinyDB の場合、センサデータの取得に要する時間は、データ 1 回目は 28771ms, 2 回目以降は 1658ms であった。ゲートウェイサービスを用いた場合、センサデータの取得に要する時間は、データ 1 回目は 29664ms, 2 回目以降は 1684ms であった。データ取得時間は、1 回目のみ時間がかかり、2 回目以降は、データ取得時間が少ない。これは、ゲートウェイサービスがアプリケーションからの要求の解析・変換によるオーバーヘッドと Sensor Bundle と Driver Bundle の検索によるオーバーヘッドが、1 回目に発生するためだと考えられる。また、TinyDB のみの場合の値よりゲートウェイサービスを用いた場合の方が取得時間が大きい場合が生じた。これは、Mica Mote2 のネットワークがデータ取得時間に与える影響に対してゲートウェイサービスが与える影響が小さいことに起因すると考えられる。この実験結果から、ゲートウェイサービスのオーバーヘッドは、1 回目

のデータ取得では約 1 秒、2 回目以降のデータ取得で 100 ミリ秒以下であることが分かった。

次に、センサデータの取得にかかる処理時間に対する時間分解能に関して考察する。データの取得周期が 1 秒単位、1 分単位、1 時間単位である場合、データ取得の 1 周期目の分解能はそれぞれ 0.03 倍、2.02 倍、121 倍となる。また、データ取得の 2 周期目以降の分解能は、それぞれ 0.59 倍、35.6 倍、2137 倍となる。したがって、データの取得周期の単位時間が 1 秒単位の場合、その周期を満たすことが困難であるが、取得周期の単位時間を長く設定することにより、時間分解能が向上する。例えば、10 倍以上の時間分解能が必要である場合、周期を数 10 秒から 1 分単位に設定すればよいことが分かる。

## 8 おわりに

本論文では、異なるセンサノードが混在するセンサネットワーク構築を支援するシステムプラットフォームであるゲートウェイサービスの構成について述べた。ゲートウェイサービスは、アプリケーションに対してセンサネットワークからセンサデータを取得するためのインタフェースと、センサネットワークに発生するイベントを感知するインタフェースを提供する。また、その際に、センサノードが備えるセンサデバイスと通信デバイスの違いを意識せずに、センサネットワークを利用することが可能となる。そのため、本ゲートウェイサービスが提供するインタフェースを用いることで、センサネットワークを利用するアプリケーションの構築が容易になる。

また、本論文では、ゲートウェイサービスを構築し、センサデータの取得時間を測定した。その結果、1 時間、1 分を単位としたアプリケーションにとってゲートウェイサービスがデータ取得時間に与える影響は少なく、適用可能であることを示した。

## 参考文献

- [1] OSGi Alliance, web page, <http://www.osgi.org/>
- [2] Crossbow Technology, web page, [http://www.crossbow.com/Products/Wireless\\_Sensor\\_Networks.htm](http://www.crossbow.com/Products/Wireless_Sensor_Networks.htm)
- [3] Open Service Container Architecture, web page, <http://oscar-osgi.sourceforge.net/>
- [4] Sam Madden, Joe Hellerstein, and Wei Hong: TinyDB: In-Network Query Processing in TinyOS, web page, <http://telegraph.cs.berkeley.edu/tinydb/>