

## 開放分散環境のためのセンサノード管理システム

河野 通宗<sup>†</sup> 太田 充彦<sup>††</sup>  
白石 陽<sup>†</sup> 安西 祐一郎<sup>†</sup>

人間の生活空間内における行動支援のために、種々のセンサを様々な家具・家電や非電子機器などに埋め込んで構築したセンサネットワークの利用を考える。このようなセンサネットワークは、利用者が自由に構築できる反面、センサノードの数や存在位置をあらかじめ定義することが不可能である。このため、従来のプラント等の組み込みシステムで用いられてきたセンサネットワークと異なり、アプリケーションはセンサネットワーク自体の動的な変化に追従する必要があり、一般に記述が困難になる。本研究では、以下のような機能を持つセンサノード群の管理システムによって、上記の問題点の解決を図る。本システムは、オブジェクト指向モデルに基づくセンサノードへのアクセスインタフェースを提供し、アプリケーション記述の単純化と種々のサービスの提供を同時に実現する。また、アプリケーションを停止させずに動的にセンサネットワークを再構成することを可能にする。これらの検証のために実装および評価した結果、アプリケーションからのセンサノードへのアクセス時間やセンサノード数との依存性などに関して目標を達成したことが分かった。

### A Sensor Node Management System for Open Distributed Environments

MICHIMUNE KOHNO,<sup>†</sup> MITSUHIKO OHTA,<sup>††</sup> YOH SHIRAIISHI<sup>†</sup>  
and YUICHIRO ANZAI<sup>†</sup>

This paper proposes a sensor node management system for a sensor network in a living environment. This research aims at constructing a sensor network consists of various sensors embedded in furniture and electrical equipments for support of human actions in living environments. Such a sensor network can be constructed by a user with a little constraints, however, it is impossible to pre-define how many and where sensor nodes are. This is one of major differences from sensor networks in plant systems, thus applications have to cope with dynamic changes of the sensor network itself. This makes applications difficult to be implemented. This paper solves the above problem by the proposed sensor network management system. The system provides programming interfaces to sensors based on an object oriented paradigm. It realizes both simple implementation of applications and provision of several services. In addition, it enables reconstruction of a sensor network without stopping any applications. We evaluated the implemented system and the result of the evaluation suggested that the system satisfied the requirements for sensor node access time in some conditions.

#### 1. はじめに

センサやアクチュエータを多数生活環境内に分散させて人間の活動を支援する研究がさかんに行われている<sup>2),6),7),12)</sup>。各種センサデバイスに演算処理装置とネットワークインタフェースを付加してセンサノードを構成し、それらを有線あるいは無線で接続した分散

ネットワークをセンサネットワークと呼ぶ。このセンサネットワークは、近年著しく進歩した計算機ネットワークの技術がホーム・オートメーションや組み込み機器内の制御にも応用されたものであり、現在標準化が進められている。中でも代表的なものは、ANSIに認定<sup>\*</sup>された LonWorks や、EIA で規格化が進められている CEBus などである。これらは各種オートメーション分野における分散制御ネットワーク構築のための規格であり、OSI 参照モデルに沿って各レイヤを規定している。LonWorks ではタスク処理とプロトコル

<sup>†</sup> 慶應義塾大学大学院理工学研究科  
Faculty of Science and Technology, Keio University  
<sup>††</sup> 富士ゼロックス株式会社  
Fuji Xerox Corporation

<sup>\*</sup> ANSI-135 Building Automation Standard (1997).

処理を1チップに収めたニューロン・チップというLSIを提供しており、これを用いることで容易に分散センサネットワークを構築できる。

本研究では、人間の生活環境内における行動支援のために、種々のセンサを様々な家具・家電や非電子機器などに埋め込んで構築したセンサネットワークの利用を考えている。このようなセンサネットワークは、利用者が自由に構築できる反面、センサノードの数や存在位置をあらかじめ定義することが不可能な開放系をなしている。このため、従来のプラント等の組み込みシステムで用いられてきたセンサネットワークと異なり、アプリケーションはセンサネットワーク自体の動的な変化に追従する必要がある、一般に記述が困難になる。

人間の生活環境内にセンサやアクチュエータを分散させた研究は今までもいくつかあるが、開放系という観点からネットワーク自体の動的な変化に着目した研究は見られなかった。

そこで上記の問題点を解消するため、センサノード数やトポロジの不定な開放分散センサネットワークのための、センサノード管理システムの枠組みを提案する。これにより、1) センサネットワークとセンサノードの各種情報の提供、2) 各センサデバイス固有のドライバをそれぞれ別々の動的ロード可能なライブラリとして実装し、実行中のアプリケーションを停止させずにネットワークを動的に変更すること(センサノードの Plug & Play)、3) センサやセンサデータへのアクセスのための統一されたプログラミング・インタフェースの提供、の3点が可能であることを示す。

この枠組みに基づき、WindowsNT 上に管理システムを実装した。センサネットワークには LonWorks 準拠のノードを 78.1 Kbps のより対線で結合したものをを用いた。実装においてはオブジェクト指向技術を用い、新しいノードがセンサネットワークに追加されたときには、自動的に対応するライブラリをロードしてインスタンスを生成するようにした。

またプログラミング・インタフェースに仮想センサクラスを提供することで、センサデータ取得とセンサフュージョン処理をアプリケーションから見かけ上区別なく実行できるようにした。基本性能に関して測定した結果、アプリケーションからのセンサノードへのアクセス時間について様々な条件下で目標仕様を達成できたことが分かった。

本論文は以下のように構成されている。まず次章において、本研究で構築するセンサネットワークの特徴と、その関連研究について詳しく述べ、本研究の位置

づけを明確にする。その次に、本研究の提案するセンサノード管理システムの枠組みについて説明し、その機能と動作について詳述する。そして4章において、実装したセンサノードと管理システム、およびそれらが構成するセンサネットワークについて述べ、具体的な構成を明らかにする。5章では実装した管理システムの基本性能とスケーラビリティに関する評価の結果を述べ、考察を加える。最後に6章で結論と今後の課題について述べる。

## 2. センサネットワーク

オフィスで使われているワークステーション等のコンピュータの多くは、分散システムを形成している。各マシンがディスクやCPU等の資源を互いに共有し、タスクはクライアント・サーバモデルに従って実行されるものがほとんどである。これらのマシンはいつネットワークから切断されるか、またいつマシンがダウンするか不明であるため、オペレーティングシステム、ファイルシステム、データベースなどに関して様々な研究がなされてきた<sup>9)</sup>。

一方センサを個々のネットワークノードとして構成したセンサネットワークでは、フォールトトレラント性やリアルタイム性が重視されてきた。これは特に、用途がプラントや組み込み機器であったためである。たとえば Smith<sup>8)</sup> は、LonWorks のニューロンチップを海底探査船の中のセンサ・アクチュエータノードに採用し、これらで構成するセンサネットワークの利点と欠点に関して議論している。また Vinyes ら<sup>10)</sup> は、1.25 Mbps のビル管理ネットワークをビデオ画像伝送に使用する際の伝搬遅延を解析している。

本研究におけるセンサネットワークの導入対象は、オフィスや公共施設などの、人間の生活環境である。以後、プラントにおけるセンサネットワークとの違いについて詳しく述べる。

### 2.1 生活環境内のセンサネットワーク

生活環境内のセンサネットワークもプラントと同様に、各センサノードを無線または有線で接続して構成する。1つのセンサネットワークは、1つの部屋や廊下など、物理的かつ論理的に区切られた空間を単位として構築するものとする。そしてこれらがルータを通して既設のコンピュータネットワークによって相互に接続され、分散センサネットワークを形成する(図1)。1つのセンサネットワークあたりのノード数に制約は考えないが、現在のところ多くても50個程度と考えている。

プラントで用いられているセンサネットワークと本

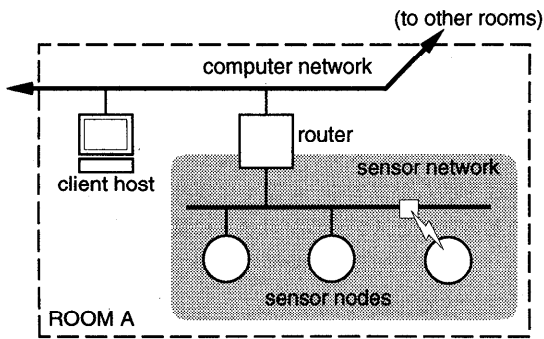


図1 生活環境内のセンサネットワーク

Fig. 1 Sensor network in a living environment.

表1 センサネットワークの違い

Table 1 The differences between sensor networks in a plant and a living environment.

	プラント	生活環境内
ノード数	一定*	不定
ノードの配置	一定	不定
システムの種類	閉鎖系	開放系

研究の想定しているセンサネットワークとの違いのうち、本研究で注目するものを表1に示す。生活環境にセンサを分散させる方法には、壁や天井に埋め込む方法と、家電や家具に埋め込む方法が考えられる。家電や家具に埋め込まれたセンサは、部屋の改築なしにセンサノードを設置あるいは移動できる反面、その数や配置が本質的に不定であるため、ノード数やトポロジが動的に変化する可能性が高い。これは、センサノードの埋め込まれた家電を、人が別の場所に動かす可能性があるという意味である。したがって、生活環境のセンサネットワークを利用するすべてのアプリケーションは、これを想定して記述されなければならない。

人間の生活環境にセンサやアクチュエータを分散させた研究には、TRON<sup>4)</sup>、ユービキタスコンピューティング<sup>11)</sup>、ユービキタスオートノミー<sup>6)</sup>、ロボティック・ルーム<sup>7)</sup>などがある。最初の3つは、センサやアクチュエータを環境内に分散させ、「物のネットワーク化」を図っている。しかし、開放系ネットワークという概念は持っておらず、センサネットワークの動的な変化に注目した研究ではない。ロボティック・ルームはそれらとはやや異なり、部屋自体をロボットと考え、カメラ、アーム、センサ等で内部の人間の活動を支援するように実装されている。この研究はセンサネットワークの応用として興味深い、1つの部屋で閉じた系を形成しており、本研究のアプローチとは異なる。

## 2.2 開放系センサネットワーク

アプリケーションが開放系センサネットワーク内のノードに対してアクセスするためには、今現在どんな種類のセンサノードが接続されているか、どの位置にセンサノードが存在するかという情報が必要である。閉鎖系であれば、あらかじめどこにどの識別番号のセンサが存在するかが既知であるので、単純にテーブルを引けば済む。しかし開放系の場合、それらの情報をあらかじめ定めておくことが不可能なので、実行時にそれらの情報を得る必要がある。

ここで、上記の情報を取得あるいは保持し、アプリケーションに提供するシステムを考える。すると、アプリケーションがセンサネットワークの状態を得る際には、このシステムに要求を出せば済むようになる。さらに、このシステムがセンサノードとの通信のプロキシサービスを行えば、アプリケーションはこのシステムとだけ通信すればよく、個々のセンサと直接通信する必要がなくなり、通信不能時のハンドリング等が容易になる。以後このシステムを「センサノード管理システム」と呼び、次章でその枠組みを提案し、詳細について説明する。

## 3. センサノード管理システム

本研究で提案するシステムの枠組みは、以下のことを行うものである。

- センサネットワークに関する種々の情報の提供
- アプリケーションとセンサノード間の通信のプロキシ
- アプリケーションへのセンサネットワークの動的な変化の通知
- センサノードへのアクセスのための統一されたプログラミング・インタフェースの提供

枠組みの全体を図2に示す。各構成要素はコンポーネントに分割して実装し、センサネットワークのハードウェアに依存する部分の取り替え・変更を容易にする。以下、この枠組みによって提供される機能について説明する。なお、この管理システムで扱うセンサネットワークとセンサノードは、以下の条件を満たしていることを前提とする。この前提は分散センサネットワークを構築するうえで妥当な前提であると考えており、LonWorksも以前我々が実装したセンサノード<sup>2)</sup>も、ともにこの条件を満たしている。

- センサネットワークに接続されるセンサノードは通信機構と演算装置を持ち、接続されたセンサのデータ処理を行うことが可能である。
- 通信メディアおよび通信プロトコルが規定されて

\* ノード数がある決まった範囲内で変動する場合も含む。

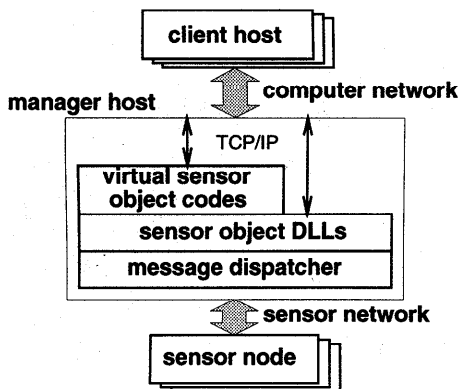


図2 センサノード管理システムの構成

Fig. 2 The framework of sensor node management system and its applications.

いて、センサノードに与えられる唯一の識別子を用いて、特定のセンサノードと通信が可能である。

- 計算機とセンサネットワークが接続可能であり、相互に通信することが可能である。
- センサノード上で実行されるプログラムはそのセンサノードのメモリ上にあり、センサネットワークに接続すればすぐにセンサの機能を利用できる状態にある。

### 3.1 センサノードの管理とセンサオブジェクト

まず、管理システムを図1のルータ上に配置することで、計算機ネットワークとセンサネットワーク双方に接続する。管理システムとそのアプリケーションにはクライアント・サーバモデルを適用し、既存の計算機ネットワーク上からの利用を容易にする。クライアントがセンサノードにアクセスする際には、管理システムに対して要求を出し、管理システムが処理を行いリプライを返す。通信にはTCPソケットを用いる。

この方式の問題点は、両ネットワーク間にまたがる処理がすべて管理システムに集中するため、データのトラフィックの集中や耐故障性の低下が生じることである。トラフィックの集中に関しては、近年のネットワークバンド幅の拡大と計算機の処理能力の向上を考慮して、深刻な影響をもたらさないと考えた。また耐故障性の問題は管理システムを複数持つことで解消できると考えるが、本論文では管理システム導入による柔軟性や拡張性の獲得に重点を置いているため、枠組みには含めなかった。

プログラミング・インタフェースにはオブジェクト指向モデルを適用する。センサノードの種類ごとにそれぞれクラスを用意し、管理システム内に、各々のセンサノードに対応するクラスのインスタンスを生成・

保持するようにする。それぞれのクラス定義を記述したコードは動的ロード可能なライブラリの形でシステムに登録することで、管理システムを停止させることなく新しいセンサノードのクラスを追加可能にする。一方、クライアントにはクラスライブラリを提供することで、クライアントと管理システム間の通信や、センサノードに対するメッセージの送受信を隠蔽することができる。

オブジェクト指向の導入の問題点は、管理システムもアプリケーションも、クラス定義を持たないセンサノードに関する処理が不可能なことである。これはコンピュータの周辺機器とデバイスドライバの関係に似ており、たとえば新しい種類のセンサノードが追加されても、対応するソフトウェアがなければ無意味である。したがって、真にセンサノードの種類に関して開放系のセンサネットワークを構築するためには、対応するソフトウェアを管理システムが自動ダウンロードできなければならない。今回はこれは考慮に入れず、必要なソフトウェアはすべて管理者がシステムに登録するものとする。

### 3.2 センサノードの Plug & Play

ここでいう Plug & Play とは、センサノードが追加されたときに、自動的に管理システム内に対応するインスタンスが生成されるという意味である。Plug & Play を実現するうえで必要な条件は、センサノードが追加されたときに、追加されたというイベントが管理システムに通知されることである。これは各センサノード上のソフトウェアが持つべき機能で、すべてのセンサノードにはこの機能が搭載されていると仮定する。

センサノードの追加や除去によってセンサネットワークが変化する場合、クライアントが利用できるセンサオブジェクトが動的に変化する。このため管理システムは利用可能なセンサオブジェクトの情報をクライアントに提供できるようにする。これは要求に応じて返答する場合と、イベントとして非同期に通知する場合の両者に対応する。提供する情報の内容は、センサノードの識別子、センサの種類、通信状態、何個のクライアントが接続しているか、対応するライブラリのバージョン番号などである。

### 3.3 仮想センサオブジェクト

計算機上のアプリケーションは1つのセンサノードから得られるセンサデータだけではなく、センサフュージョン等の処理を行うことによってノイズを除去し、かつ抽象度の上がったセンサデータも要求すると考えられる。各センサノードはそれぞれプロセッサ

を持つので、ある程度のセンサデータの処理を行うことができるが、複数のセンサノードから得られたデータのフュージョンは、センサノードよりも計算能力の高いホストで行うことが現実的な場合が多いと考える。このため、計算機上でセンサフュージョンを行うプログラムに対しても、センサノードに対してと同様にアクセスできることが望ましい。そこで、センサノードにマップされるセンサオブジェクトとは別に、センサフュージョン用のオブジェクト（仮想センサオブジェクト）を提供する。クライアントはこれらのセンサオブジェクトの違いを意識せずに利用できるようにする。

仮想センサオブジェクトは、そのオブジェクトが利用するセンサオブジェクトからは、クライアントと等価に扱われる。利用される側のセンサオブジェクトは、クライアントと同様に仮想センサオブジェクトからのメソッドコールを処理し、イベントを伝達する。また、仮想センサオブジェクトが利用可能なセンサオブジェクトは、センサノードにマップされたセンサオブジェクトだけではなく、他の仮想センサオブジェクトも利用できるようにする。

この機構を用いると、センサネットワーク特有の問題である出力値の物理的位置への依存性を吸収することができる。これは、クライアントから仮想センサオブジェクトの空間的な存在位置を与えられることで、実際にその座標にセンサが存在するか否かに関係なくデータを返答することができるという意味である。このためには、クライアントには仮想センサオブジェクトのみを提供し、各仮想オブジェクトが実センサオブジェクトへのマッピングを動的に変更可能に実装されていけばよい。

このような空間的な位置座標をキーにしたサービス要求形態は、あるセンサの出力値ではなく、ある座標のデータを得ることが本質的な要求であるクライアントにとって有効である。データの推測は、近傍に存在する同種のセンサオブジェクトの出力値や過去の履歴を用いて行われる。さらに、センサノードの故障や一時的な通信不能状態が発生した場合も、近傍のセンサオブジェクトのデータから推測して返答するような処理も可能になる。

#### 4. センサネットワークの構築

前章で述べた枠組みに基づき、センサネットワーク管理システムを実装した。LonWorksのニューロンチップを用いたノードを実装し（図3）、より対線で結合して78.1 Kbpsのバンド幅のセンサネットワークを構築した。各センサノードには後述するIndoor Positioning

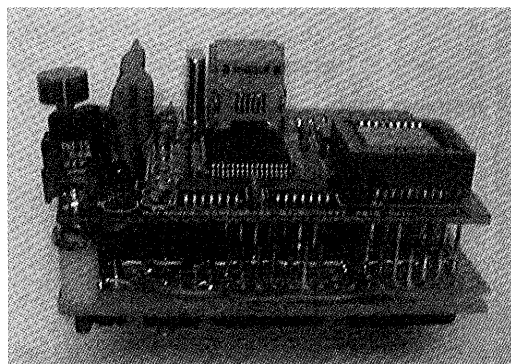


図3 センサノード

Fig. 3 A photograph of a sensor node.

System (IPS)用の回路を付加し、センサノードの空間的位置を把握可能にした。このセンサネットワークをPC/AT互換機（PentiumPro 266 MHz, 96 MB, WindowsNT4.0）に接続し、このマシン上に管理システムプログラムを実装した。以後、実装した管理システムの詳細に関して述べる。なお、実装の際のセンサノードへのアクセス時間の目標仕様は以下のとおりである。

- 1回のアクセス時間が200 ms以内であること
- センサノード数の変化（32まで）に対する依存性が少ないこと
- クライアント数の増加（16まで）に対する依存性が少ないこと

アクセス時間が200 ms以内というのは、現在すでに使用している焦電型赤外線センサによる人体検出の周期（5 Hz）から決定した。

##### 4.1 センサオブジェクト

管理システムプロセスは、各センサノードに対応するセンサオブジェクトを自プロセスのアドレス空間内に生成する。このオブジェクトには管理システム内で唯一の識別子（Object ID）を与える。クライアントとの通信時にはこのObject IDを用いてセンサオブジェクトを特定する。

システム稼働中の新たなセンサノードの追加や、故障あるいは不要なセンサノードの除去に対応するため、センサオブジェクトのコードをDLL（Dynamic Link Library）で記述した。管理システムは、DLLに対応するセンサノードがセンサネットワークに接続されたときにオブジェクトを生成する。この際オペレーティングシステムが自動的にDLLをメモリにロードする。これにより、システムを停止することなく新たなセンサオブジェクトのクラスのロードを可能にした。アンロードに関しては後述する。ただし、DLLの登録とセ

ンサノードとの対応付けは、新たな種類のセンサノードを導入する際にシステム管理者が行うものとする。

#### 4.2 センサノードの Plug & Play

センサノードの Plug & Play の機構は、PCMCIA カードのカードサービスの機構を参考にした。センサノードの種類は、センサノードプログラムの種類を表す **Program ID** によって指定される。センサノードがセンサネットワークに接続されたときに、センサノードは **Node Attach Packet** という特殊なパケットをブロードキャストする。Node Attach Packet は、センサノードのアドレスと Program ID を含んでおり、これを受信した管理システムは新しいセンサノードがセンサネットワークに接続されたことを認識する。Node Attach Packet 中の Program ID をキーにして対応するセンサオブジェクト DLL をインポートし、オブジェクトを生成する。

センサノードがセンサネットワークから取り外された場合や故障した場合には、対応するセンサオブジェクトを利用しているすべてのクライアントに通知し、センサオブジェクトを利用不可状態にする。センサノードが取り外されたことを認識するために、一定時間ごとに各センサノードに対して確認のパケット (**Beacon Packet**) を送信するようにした。Beacon Packet を受信したセンサノードは **Node Alive Packet** を送信する。Node Alive Packet が返ってこなかったセンサノードは、取り外されたか故障しているものとして対応するセンサオブジェクトを利用不可状態に変更するようにした。

#### 4.3 仮想センサオブジェクト

仮想センサオブジェクトは、複数のセンサオブジェクトからのデータを計算機上で統合することが可能なセンサオブジェクトで、対応するセンサノードを持たない。仮想センサオブジェクトにも管理システム内で唯一の Object ID を与え、クライアントからはセンサオブジェクトと同様に扱えるようにクラスを実装した。

さらに、仮想センサオブジェクトはクライアントと同様に、管理システム内の他のセンサオブジェクトを利用できなければならない。そのため、仮想センサオブジェクトには Object ID だけでなく Client ID も与え、一般のクライアントと同様の管理も行うようにした。ただし、仮想センサオブジェクトは他のセンサオブジェクトに対するメソッドコールをソケット経由で行う必要がないため、他のセンサオブジェクトのメソッドを直接呼び出すようにした。仮想センサオブジェクトとセンサノードにマップされるセンサオブジェクトの違いを 図 4 に示す。

#### 4.4 Indoor Positioning System (IPS)

空間的位置に基づくデータ要求に対応するためには、各センサノードの物理的な位置を把握する必要がある。「空間的位置」には様々な階層の概念が存在する。地球規模で考えた場合には緯度と経度の 2 次元ベクトルで位置を表す。また論理的な位置表現として住所による表記がある。緯度と経度による位置指定の計算機ネットワークでの扱い方に関しては Tomasz らによって提案されており<sup>1)</sup>、住所は現在 DNS で広く利用されている Fully Qualified Domain Name (FQDN) が近い。どちらも重要であるが、本研究で注目するのは、1 つの部屋や廊下などの物理的かつ論理的に区切られた領域内での空間的位置である。このためのシステムとして以前 IPS という機構を実装したが<sup>5)</sup>、LonWorks のノードに対応するために新規に実装した。IPS では座標の測定に超音波の Time of Flight (TOF) を用いており、各ノードは受信器を持っている。発信器は天井に設置されている。このため TOF を測定するには発信器と受信器のカウンタを同期させる必要がある。しかし LonWorks の定めるパケットハンドリングを経由しては、精度の良い同期は難しい。そこで、1 組のより対線同期用に割り当て、ハードウェア的にカウンタの動作を同期させるようにした。

### 5. 評価

実装したシステムが目標を満足しているかを検証するため、以下の項目に関して測定した。時間の測定には Windows の GetSystemTime() システムコールを使用した。どの測定も 10 万回測定し、その平均値を 1 回の測定値とした。なお、IPS の性能評価は別の論文<sup>3)</sup>で行っている。

#### 5.1 クライアントによるセンサデータの取得

管理システムを実行しているホスト上のアプリケーションが、管理システムにクライアントとしてアクセスし、センサデータを取得する際の所用時間を測定した。この測定値と、管理システムを用いないでセンサネットワークの接続しているホストからセンサデータを取得する時間と比較した。これにより、実装した管理システムとのソケット通信によるオーバーヘッドと、管理システム内の処理のオーバーヘッドを測定する。また、管理システムと同一サブネットに接続されている同じ性能のワークステーションから、センサノードにアクセスする時間を測定した。この時間はネットワークのバンド幅や利用率に依存するために、単純に比較することは難しい。しかし、アプリケーションの実行ホストに依存しないセンサノードからのデータの取得

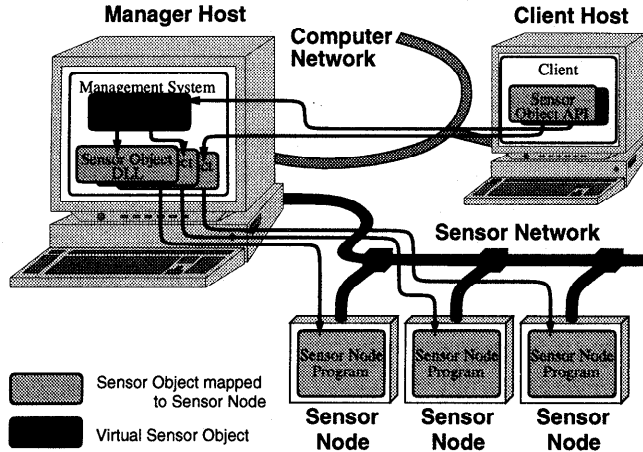


図4 センサノードとセンサオブジェクト

Fig. 4 Relationship between sensor nodes and sensor objects.

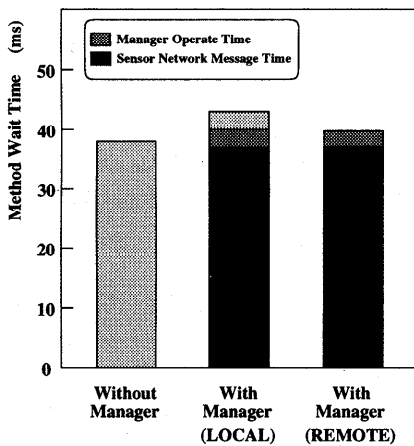


図5 センサデータ取得に要する時間の比較

Fig. 5 The result of sensor data access time.

が、どの程度のオーバーヘッドの増加で可能になったかを知ることができる。結果を図5に示す。

左から、管理システムを使用しない場合、管理システムと同一ホスト上のクライアント、他ホスト上のクライアントでの結果である。どの結果も目標の200msより低いアクセス時間であり、目標は達成されている。この結果から、以下のことが分かる。まず、管理システムを利用することにより、14.6%のオーバーヘッドが発生した。これにはソケット処理のオーバーヘッドの他、管理システム内でのリスト処理やOS内部のライブラリ結合のオーバーヘッドが含まれると考える。次に、他ホスト上のクライアントで測定した時間的コストは、同一ホストでのコストより13%少なく、管理システムを利用しない場合とほぼ同じであった(1%の増

加)。同じ性能の別ホスト上で実行したこと以外の条件は同一であるため、これは以下のように説明できると考える。パケットの送受信の際には、デバイスドライバ内での送信後の後処理や、受信時のハンドリングのオーバーヘッドがかかる。また受信待ち状態に入るまでのユーザレベルの処理も存在する。クライアントとサーバが別ノード上に存在する場合は、これらの処理が並列に実行されるが、同一ノード上の場合には逐次的に実行しなければならない。この並列化の効果がネットワークの伝搬遅延を上回ったため、時間的コストの違いが現れたのではないかと考える。

## 5.2 スケーラビリティ

次に、実装した管理システムの、(1) クライアント数の増加に対する待ち時間の変化と、(2) 1クライアントが使用するセンサノード数の増加に対する待ち時間の変化を評価するための測定を行った。結果をそれぞれ図6、図7に示す。

(1)の実験では、実際には1つのアプリケーションで $n$ 個のセンサオブジェクトを使用し、乱数でアクセスを発生させて各アクセスごとの待ち時間を測定した。図6から、クライアントの待ち時間は、クライアント数が4まではほぼ線形に増加するが、その後約50msで一定値を維持することが分かった。この現象は、同じセンサへの同時アクセスによるウェイトに原因があると考えられる。つまり、クライアント数が少ない間は、同時アクセスによる影響が顕著に待ち時間に現れるが、ある程度以上になると、一定時間内に同時アクセスの起きる確率がほぼ一定になり、待ち時間が線形に悪化していくことにはならないと考える。以上よ

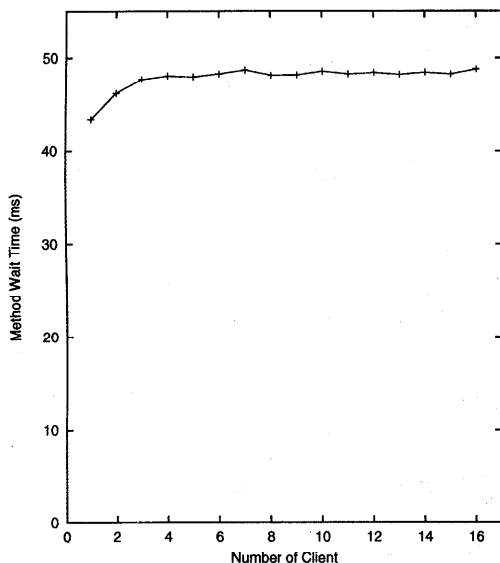


図6 クライアント数に対する平均待ち時間の変化

Fig. 6 Average wait time in regard to the number of clients.

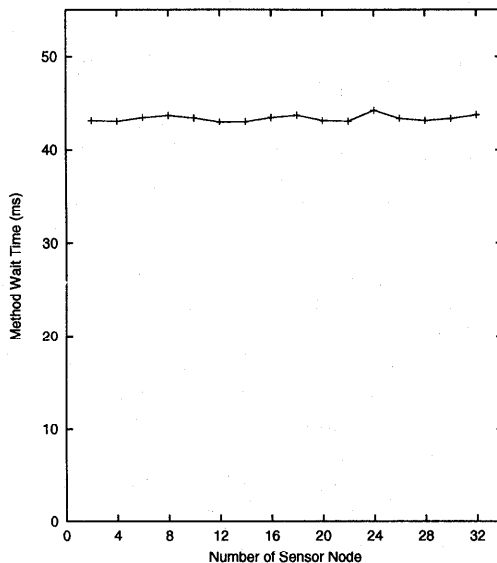


図7 センサノード数に対する平均待ち時間の変化

Fig. 7 Average wait time in regard to the number of sensor nodes.

り、クライアント数の変化とアクセス時間に関する目標を達成できたことが分かった。

図7から、クライアントの平均待ち時間が、センサネットワークに接続しているセンサノード数の変化によらずほぼ一定であることが分かり、目標を達成していることが分かった。これは内部のクライアント管理等に要する計算オーバーヘッドがきわめて微量で、センサノード数に対して依存しないことを示しており、実際のシステムに適用するためにはきわめて重要な性質であると考えられる。

## 6. 結 論

生活環境内のセンサネットワークが開放系であることを示し、センサノード数や位置が動的に変化するセンサネットワークにおけるセンサノード管理システムの枠組みを提案し、実装した。この際仮想センサの概念を実装し、IPSと組み合わせることでセンサネットワークの特徴である空間的位置の依存性を吸収した。このシステムのスケーラビリティに関して評価し、目標仕様を満足する性能を持っていることを示した。

実験結果から、時間的コストの大部分がセンサノードの処理やセンサネットワークの伝送時間が占めており、管理システム自体は良好な性能を示していることが分かった。実験で得られた特徴が一般的なものか否かは、これだけの実験では判断できない。しかし、管理システムの実装はセンサネットワークの媒体にほと

んど依存していないため、より高い処理能力のセンサノードと広いバンド幅を持つセンサネットワークを利用すれば、その分の性能向上を得ることはできる。

残った問題点としては、本システムではまだ完全に開放系センサネットワークには対応しきれないということがあげられる。特に、新規に追加されたセンサノードに対応するソフトウェアの問題は、将来の拡張性を得るためには必ず解決しなければならない。今後はこの問題点を解決しつつ、また無線化したセンサノードを作成して利便性を高め、アプリケーションを多数作成していきたいと考えている。

謝辞 本研究は、日本学術振興会による未来開拓学術研究推進事業の補助を受けて行われた。またLonWorksは、ECHELON社の登録商標である。

## 参 考 文 献

- 1) Imielinski, T. and Navas, J. C.: GPS-Based Addressing and Routing, RFC 2009 (1996).
- 2) Kohno, M. and Anzai, Y.: Intelligent Sensor Network as Nervous System of Artifacts, *Proc. International Symposium on System Life*, pp.109-115 (1997).
- 3) Kohno, M. and Anzai, Y.: An Adaptive Sensor Network System for Complex Environments, *Proc. 5th International Conference on Intelligent Autonomous Systems*, pp.21-28 (1998).
- 4) Koshizuka, N.: The current status and future

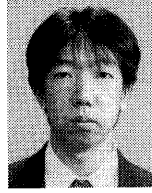


of the BTRON and Human-Machine Interface Subprojects, *Proc. 13th TRON Project International Symposium*, pp. 22-27 (1996).

- 5) 松永昌浩, 河野通宗, 安西祐一郎: 超音波による室内オブジェクトの3次元位置検出システムの設計と実装, 第15回日本ロボット学会学術講演会予稿集, Vol.1, pp.183-184 (1997).
- 6) 小笠原司, 築根秀男, 平井成興, 富田文明, 國吉康男: ユービキタスオートノミーロボット遍在技術の確立を目指して, 日本ロボット学会学術講演会予稿集, Vol.13, No.3, pp.1197-1198 (1995).
- 7) Sato, T., Nishida, Y. and Mizoguchi, H.: Robotic room: symbiosis with human through behaviour media, *Robotics and Autonomous Systems*, Vol.18, No.1-2, pp.185-194 (1996).
- 8) Smith, S.: An Approach to Intelligent Distributed Control for Autonomous Underwater Vehicles, *Proc. 1994 Symposium on Autonomous Underwater Vehicle Technology*, pp.105-111 (1994).
- 9) 徳田英幸: 分散リアルタイムシステムのためのOSアーキテクチャ, *情報処理*, Vol.35, No.1, pp.18-25 (1994).
- 10) Vinyes, J., Vazquez, E. and Miguel, T.: Throughput analysis of p-CSMA based LonTalk protocols for building management systems, *8th Mediterranean Electrotechnical Conference*, Vol.3, pp.1741-1744 (1996).
- 11) Weiser, M.: The Computer for the Twenty-First Century, *Scientific American*, Vol.263, No.3, pp.94-104 (1991).
- 12) Williams, B. and Nayek, P.: Immobile Robots: AI in the New Millennium, *AI Magazine*, Vol.17, No.3, pp.16-35 (1996).

(平成10年4月8日受付)

(平成10年11月9日採録)



河野 通宗 (学生会員)

昭和46年生。平成6年慶應義塾大学理工学部電気工学科卒業。平成8年同大学大学院計算機科学専攻修士課程修了。現在同大学院理工学研究科計算機科学専攻博士課程在学中。



太田 充彦

昭和48年生。平成8年慶應義塾大学理工学部電気工学科卒業。平成10年同大学大学院計算機科学専攻修士課程修了。同年富士ゼロックス株式会社入社。



白石 陽 (学生会員)

昭和46年生。平成6年慶應義塾大学理工学部電気工学科卒業。平成8年同大学大学院計算機科学専攻修士課程修了。現在同大学院理工学研究科計算機科学専攻博士課程在学中。



安西祐一郎 (正会員)

昭和21年生。昭和49年慶應義塾大学大学院博士課程修了。昭和63年より慶應義塾大学理工学部教授。平成元年より同大学大学院計算機科学専攻教授兼任。この間昭和56～57年カーネギーメロン大学客員助教授。計算機科学、認知の情報処理過程の研究に従事。工学博士。日本ロボット学会、電子情報通信学会、日本神経回路学会、日本心理学会、ACM、IEEE等会員。