

センサ・アクチュエータネットワークにおける 仮想ネットワークの構成と資源管理機構

金丸 達雄[†] 横田 裕介^{††} 大久保 英嗣^{††}

[†]立命館大学大学院理工学研究科 ^{††}立命館大学情報理工学部

スマートタウンやビルなどの大規模施設を運用する場合、同一空間内で複数のサービスを実行したいという要求がある。複数のサービスをセンサ・アクチュエータネットワークで実施する場合、ノードを共有することでセンサ・アクチュエータの量を削減できる。さらに、サービスを独立して管理することで管理コストが削減できる。本稿では、この要求を満たすセンサ・アクチュエータネットワークにおける仮想ネットワークの構成を提案する。また、提案する仮想ネットワークでは、複数のサービスを実行するために各ノード上で複数のプロセスが動作する。このとき、複数のプロセスの動作による資源競合や、プロセスごとに異なる目的で資源利用を行うことによる矛盾が発生する可能性がある。これらの問題を解決するためのノード上の資源管理機構についても述べる。

A Virtual Network Organization and Resource Management Mechanism in Sensor and Actuator Networks

TATSUO KANAMARU[†] YUSUKE YOKOTA^{††} EIJI OKUBO^{††}

[†]Graduate School of Science and Engineering, Ritsumeikan University

^{††}College of Information Science and Engineering, Ritsumeikan University

Operation of smart towns or large buildings requires execution of multiple services at the same area. The cost of services can be reduced by introducing a sensor and actuator network which executes multiple services, because shared use of nodes among services will decrease the number of nodes. The management cost of services will also be decreased if dependencies among services are removed. This paper proposes a virtual network mechanism on sensor and actuator networks that meets these requirements. Multiple processes will run on each node of the proposed virtual networks to provide multiple services. It may cause some problems, resource competition of multiple processes and inconsistent use of resources when multiple processes use the same resource by different purposes. We also introduce a resource management mechanism to solve these problems.

1 はじめに

スマートタウンやビルなどの大規模施設を運用する場合、同一空間内で複数のサービスを実行したいという要求がある。例えば、商業施設において施設全体で利用するサービスと特定の業務エリア向けのサービスを提供したいという場合が考えられる。施設全体でのサービスとしては、異常検知(火災検知, 建築物のモニタ)や環境管理(空調管理, 消灯点灯管理, 扉の開閉), また, 特定の業務エリア向けのサービスとしては, 業務用品管理(商品温度の

管理, 商品数量の管理)と業務用品の情報提供(商品の取り扱い履歴の提供, コマーシャル)などが挙げられる。

このような施設において効率的にサービスを実行するための手段の1つとして, 無線センサ・アクチュエータネットワークの利用が挙げられる。無線センサ・アクチュエータネットワークは, センサノードからの環境情報の取得, および環境情報に基づいたアクチュエータノードの制御といった機能を提供する。これにより, 環境の観測と遠隔からの機器の運用が容易に実現でき, 管理コストの削減につながる。また, 無線による通信を行うため, 配線コス

ト等の設備投資を削減できる。

従来の無線センサネットワークシステムの多くは単一の目的でシステムを運用することを想定しており、目的の異なる複数のサービスを同時に実行するための検討はあまりなされてきていない。しかし、サービスごとにネットワークを構成した場合、必要なノード数が増加し、設備および管理のコストが増加することになる。

本稿では、同一空間内で複数のサービスを実行するための、無線センサ・アクチュエータネットワーク上における仮想ネットワーク（以下、VN と記す）を提案する。この VN とは、単一の物理ネットワーク上で、複数のサービスがノードや通信経路を共有して構成するネットワークである。ノード上でサービスの実行に VN を用いた場合、サービスごとに独立したネットワークを用意する場合と比較して、センサ・アクチュエータノードの量を削減することができる。また、サービスごとに独立している場合と比較して、広域から多様な環境情報を取得することが容易になる。このため、効率的なセンサ・アクチュエータの制御が可能になる。

VN を用いることで、サービスの独立性を高めることができる。すなわち、サービスの管理者および利用者は、他のサービスとの依存関係や他のサービスに与える影響等を意識せず、単一のネットワークを占有的に利用する場合と同じように利用することが可能になる。結果として、全体の管理コストの増大を抑えることができる。

VN を用いて複数のサービスを実行する場合、ノード上では各サービスに対応した複数のプロセスが並行して動作し、それぞれの目的に応じた計算機資源の利用を行う。そのため、資源（センサ、アクチュエータ、無線通信装置）の競合や資源利用方針の矛盾が発生する可能性がある。この問題に対処するため、ノード上に資源管理機構が必要となる。

本稿では、センサ・アクチュエータを用いた VN の構成について述べ、VN 上で発生する問題を解決し、サービスの高信頼性を実現するための資源管理機構について述べる。以下、2 章で仮想ネットワークの概要、3 章で仮想ネットワークの利用例、4 章で仮想ネットワークを構成するノードに対する要求、5 章で仮想ネットワークを構成するノードのための資源管理機構、6 章で仮想ネットワークと資源管理機構の実装と評価に向けた検討について述べ、おわりに本稿をまとめる。

2 仮想ネットワークの概要

本稿で提案するネットワークは、global network と local network という 2 種類の VN から構成される（図 1）。各 VN は、1 つの基地局と複数のセンサ・アクチュエータノードを持つ。1 つのサービスは 1 つの VN に所属する。また、1 つの VN には複数種類のサービスが所属することができ

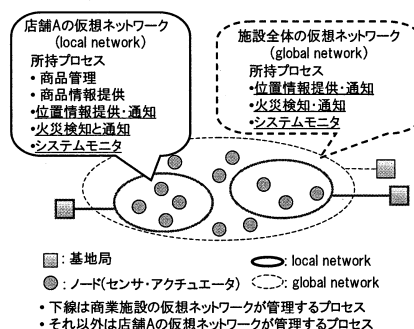


図1 仮想ネットワークの構成例

る。各ノード上では、サービスを実行するためのプロセスが動作する。1 つのサービスに対し、各ノード上で 1 つ以上のプロセスが対応する。

global network では、施設の運用、危機管理などネットワーク全体に提供するサービスを管理する。このネットワークは唯一存在し、すべてのノードが所属する。global network が管理するサービスのためのプロセスは、local network のものよりも優先度が高く設定される。これは、global network で扱うサービスは事故や故障の防止など緊急性の高いものが多いことによる。local network は、店舗やオフィスなど特定の場所に限定したサービスを管理する。local network はネットワーク上に複数存在し、対応するサービスを実行するノードのみが所属する。なお、複数の local network 間では、ノードを共有しないものとする。

次に、VN の構成要素である基地局、センサノード、アクチュエータノードについて述べる。

基地局 各 VN を管理する、計算能力の高いノード（PC など）である。各 VN に所属するセンサノードからセンシングデータを収集する。また、各 VN に所属するノードに対するプログラムの配布やアクチュエータの制御を行う。

センサノード 様々な環境情報を取得するためのセンサを搭載したノードである。指定されたサンプリング周期ごとにセンサからデータを取得し、隣接したノードもしくは基地局にセンシングデータを送信する。また、隣接ノードとの協調動作を行う。協調動作の例としては、基地局までの通信の負荷分散、アクチュエータの動作結果の観測とそれに基づいた制御などが挙げられる。

アクチュエータノード 実世界に作用する機器であるアクチュエータを搭載したノードである。アクチュエータは、基地局による制御に従って動作する場合と、周

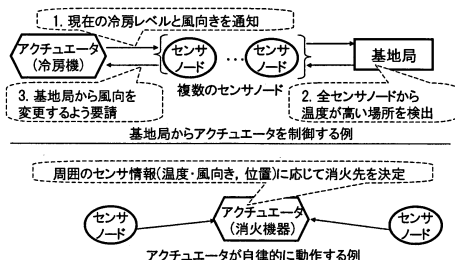


図2 アクチュエータの動作例

囲のノードから情報を取得し自動的に動作する場合がある。前者は、アクチュエータの動作を決定するための計算処理プロセスの負荷が高い場合や広域のセンサ情報が必要な場合に使用される。後者は、高速な応答が必要な場面に用いられる。この動作例を図2に示す。

3 仮想ネットワークの利用例

本章では、VNの利用例を示す。まず初めに、ある施設で新規にサービスを追加する例を挙げ、ノードの共用によるコスト削減が可能であることを示す。次にVN上でのアクチュエータ制御の例を挙げ、効率的な制御が可能であることを示す。

ある施設において新たなサービスを追加する場合、global network内でこのサービスに対応するlocal networkに参加するノードを決定し、対象ノードにプロセスを配布後、local networkを構築する(図3)。このプロセスはlocal network用のサービスを実現し、global networkのプロセスと同時に動作する。このとき、各VNの基地局は異なるため、VNごとに異なる方向へのデータフローが発生する。このように複数のサービスでノードを共有し、ノード上でそれぞれのサービスに対応するプロセスを並行して実行することで、センサ・アクチュエータノードの量を削減できる。

次に、図4に示す本VNを利用したアクチュエータの制御例について述べる。この例では、施設内に客足感知サービス(global network)と空調管理サービス(local network)が設置されている。客足感知サービスは、人感センサノードから構成されており、客足の情報を取得することができる。また、空調管理サービスは、温度センサノードから情報を収集し、基地局から空調機に指示を出すことで空調の制御を行う。このとき、空調管理サービスに所属する基地局は、客足感知サービスに所属する基地局へ人の流れについての問い合わせができる。これにより、自身の保持する温度情報とともに人の流れについても評価することが可能となり、効率的な空調制御が可能となる。このように、センサ・アクチュエータノード上にVNを実現することで、

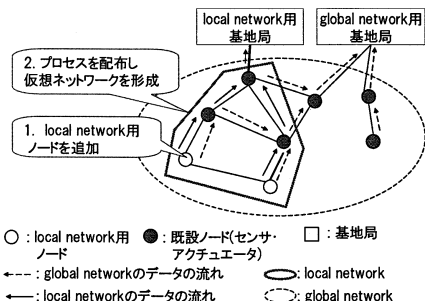


図3 商業施設において新規に業務を実施する例

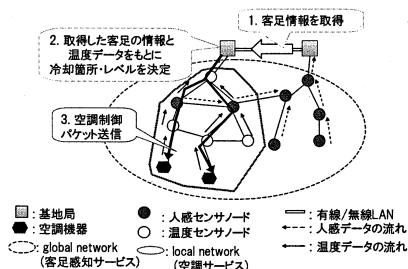


図4 仮想ネットワークを利用したアクチュエータの制御例

広域からの多様な情報を取得し、効率的な制御が実現可能となる。

4 仮想ネットワークを構成するノードに対する要求

本章では、VN上でサービスを実現するためのノードに対する要求について考察する。以下、サービスを単位とした処理の管理と資源利用の管理について示す。

4.1 サービスを単位とした処理の管理

本VNでは、ノードはサービスごとに処理の管理を行う。これは、各サービスの異なる要求を実現すると同時にサービス間での動作の衝突や矛盾を防ぐための管理を容易にするためである。

各サービスの要求は、観測したい事象・物体、応答時間、稼働期間、情報の粒度などによって定められ、その結果、データの集約方法やセンシング間隔などの処理方式が定まる。

例として、あるノードにおいて火災検知サービスと空調管理サービスの双方が実行されている場合を想定する。これらのサービスは、同じ温度センサを利用するが、独立してそれぞれ異なる処理を行う。これらのサービスにおける特徴を以下に示す。

- サービスごとに基地局へ送るデータの提供方法が異なる。火災検知サービスでは火災発生イベントを通知するが、空調管理サービスでは温度データそのものを送信する。火災検知サービスは、火災を検知した場合のみ基地局に通知すればよい。したがって、火災が発生した箇所のノードのみがデータを送信する。一方、空調管理サービスは、すべてのノードから温度データを定期的に送信する必要がある。これは、空調を行う施設全体の状況を把握し、その中から優先的に空調を制御する箇所を決定するためである。また、高い応答性は要求されないため、通信経路上でデータの集約を行い電力消費を抑えることが可能である。
- サービスごとにセンシング周期が異なる。空調管理サービスと比較した場合、火災検知サービスのセンシング間隔は短い。その理由として、火災検知サービスは火災をすぐに検出して拡散を防ぐ必要があることが挙げられる。このように、同一のセンサを用いる場合でも異なるセンシング間隔で情報を取得できるようにする必要がある。

以上より、本 VN 上でサービスを実行するためには、サービスごとに異なるプロセスを用いてノード上の動作を管理することが望ましい。サービスごとに要求される通信方法やセンシング方法などが異なるため、プロセスをサービス処理の管理単位として用いることによってこの要求を満たすことができると考えられる。

プロセスを利用する理由として、従来行われていた、収集したセンサデータを解析してサービスを提供する手法 [1, 2] では、独立した複数のサービス実現が困難であることが挙げられる。この手法は、ノードごとに一定の周期でセンシングを行い、その結果を基地局のデータベースに格納する。次に基地局のデータベースに蓄えられたデータを解析し、サービスごとに解析とその結果に応じた処理を行う。この方法では、1つのノードで複数の処理方式を持つことができず、サービスを独立して管理できない。

4.2 資源利用の管理

複数のプロセスを単一のノード上で稼働させる場合、(a) 資源割当の競合、および (b) プロセスごとに異なる目的で資源を利用することによる矛盾の発生、という 2 つの問題を解決する必要がある。

(a) が発生した場合、ノード上でプロセスが実行不能となる。これは、単一のプロセスが常に資源を保持している場合、他のプロセスがその資源を利用できなくなることに起因する。例えば、温度センサがあるプロセスによって常にロックされているために、空調管理サービスのプロセスが実行できなくなるような場合が考えられる。

また、(b) が発生した場合、プロセスの実行がサービス

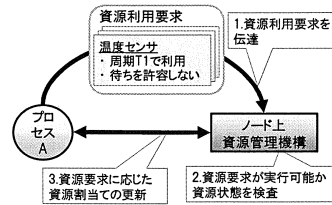


図5 資源管理機構の役割

の要求を満たさないという問題がある。例えば、プロセス A および B がともに移動機器の制御を行っている場合を想定する。プロセス A はサービス S1 に対応し、荷物 B1 を地点 P1 に運搬することを目的としている。また、プロセス B は、サービス S2 に対応し、荷物 B2 を受取りに地点 P2 へ移動することを目的としている。このとき、プロセス A と B それぞれが決定した機器の移動方向が逆である場合、移動が不能になる。

(a), (b) を防ぎサービスを目的と矛盾なく実行するには、ノード上の資源状態を検査しプロセスへ割当てを行う資源割当て機能と、競合を検知し資源割当ての状態を更新する競合検知機能が必要である。そのため、ノード上でこれらを実現する資源管理機構が必要である。

5 仮想ネットワークを構成するノードのための資源管理機構

本章では、4 章で述べた要求を実現するために必要となるノード上の資源管理機構について述べる。資源管理機構は、図 5 に示すように、ノード上のプロセスの資源利用の方法を検査し、資源利用が可能であれば、プロセスに資源を割当てる機構である。以下、資源管理機構の概要、資源割当て期間、資源管理機構の動作例について述べる。

5.1 資源管理機構の概要

4.2 節において、資源割当て機能と競合検知機能が必要であることを述べた。提案する資源管理機構は、資源の種類に応じた資源割当てを可能にすることでこれらを解決する。

提案する資源管理機構は、プロセスからの要求に基づき資源の種類に応じた、資源割当て期間と資源割当ての方針を決定する。提案する資源管理機構の構成を図 6 に示し、以下にその動作について述べる。

1. プロセスは、資源管理機構に対して資源予約 API を介して、資源利用要求を伝達する。この資源利用要求は、下記の項目を含む。

- 利用する資源の種類
- 資源割当て期間
- ブロックによる待ちを許容するか否か

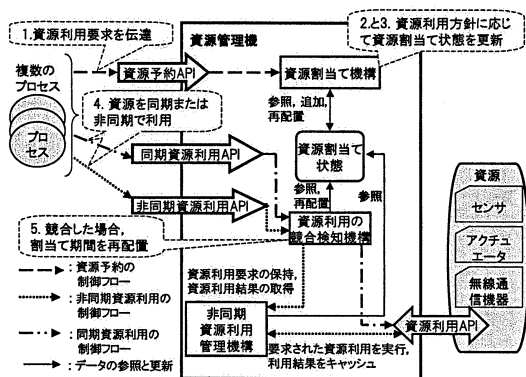


図6 資源管理機構の構成

- 資源割当て期間の変更は可能か否か

この利用要求は、プロセスが新たに資源を利用する前（ノードへの新規参入時、新たな資源が必要となった場合）にプロセスから発行される。

2. 資源割当て機構は、資源利用要求とノード上の資源割当て状態を比較する。資源利用が可能な期間があれば、資源割当て状態を更新し、プロセスに対して資源割当てを行う。この資源割当ての結果はプロセスに通知される。
3. 資源利用要求と資源割当て状態が競合した場合、資源割当て機構は、資源割当て状態を変更する。これは、資源割当て期間の再配置に対応する。資源割当て期間の再配置は、資源割当て状態を参照し、資源割当て期間の変更が可能か、また、待ちを許容するかを参照して決定する。次に例を挙げる。待ちを許可するプロセス A に対しては、競合するプロセス B の資源利用終了後に A の資源割当て期間を配置する。また、待ちを許可しないプロセス C に対しては、競合するプロセス D の実行前に C の資源割当て期間を配置する。
4. 資源を割当てられたプロセスは、図7のように資源管理機構のAPIを通して同期、または、非同期に資源を利用する。同期利用APIは、プロセスが資源割当て期間に資源を使用するためのAPIである。また、非同期利用APIは、資源割当て期間に要請した資源利用を行うAPIである。プロセスからの資源利用呼び出しと実際に資源利用を行うタイミングを異なるものにする事で、資源利用の自由度を高めることができる。非同期利用APIの資源利用の結果は、プロセスが任意のタイミングで非同期利用APIを再び呼ぶことで取得する。このとき、必要であれば再度

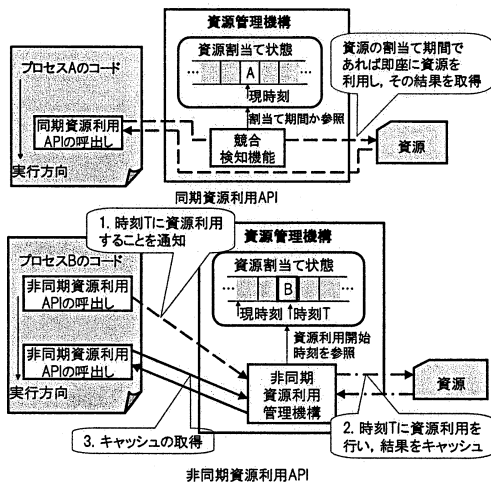


図7 同期的および非同期的な資源利用

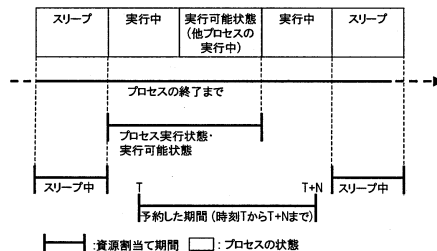


図8 資源割当て期間の種類

非同期の資源利用を要請できる。なお、非同期APIが取得した資源利用の結果は、資源利用の要求を行ったプロセスが次にスリープするまで保持される。

5. 以後、資源利用のAPIを介し、競合を検知した場合、資源割当て期間の再配置を行う。なお、再配置不能の場合、優先度の低いプロセスの資源割当てを中止し、プロセスの排除をスケジューラに要請する。

5.2 資源割当て期間

資源管理機構は、資源ごとに資源割当て期間を管理する。資源割当て期間は、資源割当て機構による資源割当て、および再配置を行う際の管理単位として利用される。また、競合検知機構の競合検知の単位、非同期資源利用管理機構の資源利用の開始時刻としても使用される。本資源管理機構が利用する資源割当ての期間は、プロセスがいつ資源を利用するかという観点から4種類に分けられる。以下、この4種類の資源割当て期間について説明する（図8参照）。

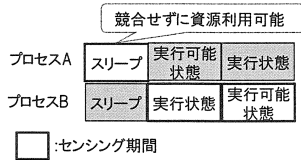


図9 スリープ中状態における資源利用

プロセスの終了まで プロセスの終了まで資源を占有し、他のプロセスが利用できない状態にする。主にアクチュエータの保護のために用いられる。これは、アクチュエータは、通常、単一の目的で利用することが望ましく、複数の異なる目的を持つプロセスで共用した場合、制御内容に矛盾が生じるためである。

プロセス実行状態・実行可能状態 スリープ以外の時間に資源を利用する。これは、資源利用の結果に応じて資源利用方法を変更するプロセスへの資源割当ての保護を実現する。例として、無線通信装置による連続した手続きが必要な場合への対処があげられる。また、他の例として、測距デバイスを用いた物体の動作確認が挙げられる。これは、測距した結果近傍に物体が存在することが分かった場合、連続して測距を行いたいという要求が発生するためである。

スリープ中 スリープ中に該当する資源を利用する。これにより、プロセスは、定期的な資源利用が保障される。例として、図9のような、周期的にセンシングを行うプロセスAと、実行状態および実行可能状態期間内の任意のタイミングでセンシングを行うプロセスBが並列に動作する場合を想定する。このとき、プロセスAは、自身のスリープ中にセンシングを行い、プロセス内で非同期にデータを取得することで、プロセスBによる任意のタイミングのセンシングを回避できる。

予約した期間 任意の期間における資源の利用を可能にする。プロセスが指定した時刻、指定した期間に資源の利用を行う。これにより、他プロセスは、その時間帯を回避した資源利用が可能となる。例として、同期通信を行う場合を想定する。プロセスは、同期通信が発生する時刻 T から $T+N$ まで、無線通信機器の受信状態の割当てを行う。これにより、他のプロセスの通信に妨害されることなく、他ノードからタイムスタンプが保持されたパケットを受け取ることができる。

5.3 資源管理機構の動作例

本節では、資源管理機構の動作例について述べる。想

定環境を図10に示し、以下に説明する。

- 2種類のプロセスA・Bが、センサを利用しセンシングを行っている。
- プロセスA・Bは、センサを同時に割当てできない。これは、センシングを行う際にプロセスA・Bが個別に精度(有効ビット数)を指定する必要があることによる。
- プロセスAは、時刻N1を基点として、周期Tでセンサの割当てが行われている。
- プロセスBは、実行状態・実行可能状態にセンサの割当てが行われている。

このような環境では、プロセスAのスケジューリングによっては、資源利用の競合が発生する。これは、Aの実行状態・実行可能状態中にBのセンシング期間が訪れる場合や、Bのセンシング中にAが実行される可能性があることに起因する。このような場合、資源管理機構は、次のように問題の発見と解決を行う。

1. 資源利用のAPIを介して、プロセスA・Bの資源の競合を発見する。
2. 資源割当て状態を見て、資源利用の期間が変更可能か、待ちが可能か否か調べる。競合しているプロセス間で待ちが可能なプロセスが発見されれば、そのプロセスの資源利用要求はキューイングして実行する。
3. 待ちが不可能な場合は、資源期間の変更が可能なプロセスを発見し、次の対処を行う。
4. スケジューラからプロセスBの開始時刻を取得し、プロセスAの資源割当ての基点時刻N1を競合が発生しないN2に変更する。N2は、競合するプロセスの開始時刻C1と前回のプロセスAのセンシング期間Lを用いて求める。N2の導出式は、 $N2 = C1 - L$ である(図10の下段の図を参照)。
5. 資源利用のAPIを介して、プロセスA・Bの資源の競合を調べる。競合が解決しない場合、優先度の低いプロセスを停止する。

6 仮想ネットワークと資源管理機構の実装と評価に向けた検討

本稿で提案した手法の有効性を確認するため、プロトタイプシステムとしてglobal networkに火災検知を模倣したサービス、local networkに空調管理を模倣したサービスを持つシステムを作成する予定である。また、この環境が正常に稼動するか確認するため、基地局において期待されるデータが取得できるか確認する。

また、ノード上の資源管理機構による競合や矛盾の解消が可能であることを検証する。これらは、資源の競合発生に関するログを取得することで行う。

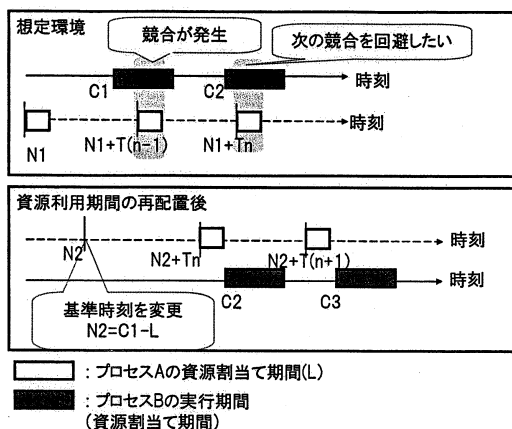


図 10 資源の再割当ての例

火災検知を模倣したサービスは、温度センサから取得した温度に基づき、基地局で火災箇所を把握するサービスである。このサービスは、温度センサを搭載したノード、温度センサを搭載しない通信専用ノードと基地局に相当する管理用 PC を用いて構築する。温度センサを搭載したノードは、定期的にセンシングを行い、センシング値が指定された閾値を超えた場合のみ、基地局に対しイベントの通知を行う。なお、センサノードには MICAz [3] を用いる。

空調管理を模倣したサービスは、温度センサから取得した温度に基づき、高温箇所を冷却するサービスである。使用する機器は、火災検知サービスで利用するノードに加え、仮想的な空調アクチュエータノードを用いて実現する。仮想的な空調ノードは、基地局からの指示に応じて、空調の制御を実行する代わりに LED を点滅させる。基地局からの指示は、温度センサを搭載したノードから定期的に取得する温度情報に基づき決定される。

ノード上の資源管理機構の評価では、ノードの環境に応じてプロセスが適切に稼働することを確認する。具体的には次の 2 点の確認を行う。

- ノードがその種類に応じた動作を行うこと
搭載しているセンサ・アクチュエータに応じたサービスが実施できることを確認する。
- プロセスの方針に沿った資源利用が行われていること
複数のプロセスが同一資源の利用を行う場合の資源割当て状況を確認する。割当てに失敗した場合、プロセスの方針に応じた例外処理を行うことができることを確認する。

ノード上の資源管理機構は、MICAz 上で動作する LiteOS [4] カーネルを改変し実装する予定である。LiteOS は、ミ

ネソタ大学で開発されたセンサノード用 OS であり、以下のような特徴を持つ。

- Unix のファイルシステムライクなネットワークの管理
- 基地局との対話的シェルを持つ
- ノード上のファイルシステムの実現
- 複数のスレッドを実行可能
- スレッド間におけるメモリ保護 (一部) を実現
- 無線を介したスレッドの配布が可能

7 関連研究

本 VN のような多目的の制御を目的としたネットワークとして、SANET [5]、LonWorks [6] がある。また、ノード上での資源確保のための調停が可能な Agilla [7]、危険な命令の実行を保護する機能をもつ t-kernel [8] がある。

SANET は、ユーザからの要求時にネットワークを形成することでサービスを提供する。これにより、複数のユーザ要求に対し柔軟な対処を実現する。特徴として、ノードが役割を持ち、それに応じて実行するアプリケーションのバイトコードを選択することや、基地局を前提としないため、耐障害性が高いことが挙げられる。

LonWorks は、ビル制御のためのシステムである。LonWorks 用のチップを搭載した機器 (センサ、アクチュエータ、制御機械) は、Lontalk というプロトコルを用いて相互に接続される。これにより、自律的に周囲の機器と協調動作してサービスを提供する。新規にサービスを実施する場合、既設ネットワークを利用できるため、必要な機器を追加するだけでよい。そのため、サービスごとにネットワークを構成する必要がなく、コストの削減を実現できる。また、基地局を前提としないため、耐障害性が高い。

Agilla は、センサノード用モバイルエージェントである。Agilla によるサービスの実行は、エージェントの評価関数に基づき複数のノードを移動することで達成される。また、エージェント間で交渉を行うことで資源の保護を実現する。

t-kernel は、危険な命令に対する保護と OS による制御の保証を行う機能を持つ。これは、ページングの際、ジャンプ命令をカーネルへのジャンプ命令に変更することで実現される。カーネルでは、ジャンプ先が安全であるか確認する。さらに、カーネルへジャンプしたとき、OS からの処理要求が存在すれば、アプリケーションを中断する。

SANET や LonWorks と比較した場合、本 VN の特徴としては管理が容易であることが挙げられる。本 VN は基地局を前提としてサービスを行うため、広範囲のサービスの実施状況を容易に把握することができる。また、SANET や LonWork と比較して堅牢性が高い。これは、基地局から配布したプロセスをもとにノード上で資源割当てを行うため、プロセス間の影響を予測しやすくなるためである。

Agilla や t-kernel と比較した場合、提案する資源管理機構は、統一的な資源管理を行うため、資源管理が容易である。Agilla は、プロセス同士で資源割当ての交渉を行う場合がある。この場合、資源利用を行うプロセスは、他のプロセスが利用するすべての交渉方法を持つ必要がある。これは、すべてのプロセスと交渉を行わない場合、資源の割当てが失敗する可能性があるためである。この複数の交渉方法を保持することは、プロセスの資源管理を困難にする。しかし、本機構ではプロセスごとに別の調停方法が発生せず管理が容易である。また、t-kernel と比較した場合、本機構はエラー原因の解決までを行う。このため、サービスを実行するプロセスがノードの状態を把握して、自身の処理を変更する必要がない。

8 おわりに

本稿では、センサ・アクチュエータネットワークにおける仮想ネットワークの構成と資源管理機構について述べた。ネットワーク上で複数のサービスを実施することにより、ノードの共用による効率化、無線通信の利用による配線コストの削減を実現し、大規模な施設などにおける効率的な環境観測や機器制御を可能にする。また、仮想ネットワークを用いることでサービスの独立性を高め、管理を容易にしている。そのため、サービスの管理者および利用者は並行して動作する他のサービスとの競合問題を意識する必要は無い。また、ノード上で動作する資源管理機構は、複数のプロセスに対する資源割当ての調整を行う。プロセスからの資源割当てに対する要求およびプロセスごとに定められた優先順位に基づき、資源の競合や資源の利用方針との矛盾を防ぐような資源割り当てを行い、サービスの目的と矛盾しない処理を実現する。今後は、資源割当て手法・再割当て手法、資源の種類ごとの割当て方針の詳細な検討、および資源の解放方針の検討を行った上で実装作業を進める予定である。

参考文献

- [1] Madden, S., Franklin, M., Hellerstein, J. and Hong, W.: TinyDB: an acquisitional query processing system for sensor networks, *ACM Transactions on Database Systems (TODS)*, Vol. 30, No. 1, pp. 122–173 (2005).
- [2] Chu, X.: Open Sensor Web Architecture: Core Services, PhD Thesis, The University of Melbourne, Australia (2005).
- [3] : Crossbow Technology: Wireless Sensor Networks: MICAz 2.4 GHz–Wireless Module, <http://www.xbow.com/Products/productdetails.aspx?sid=164>.
- [4] Cao, Q., Abdelzaher, T., Stankovic, J. and He, T.: The LiteOS Operating System: Towards Unix-Like Abstractions for Wireless Sensor Networks, *Information Processing in Sensor Networks*, 2008. *IPSN'08. International Conference on*, pp. 233–244 (2008).
- [5] Eltarras, R., Eltoweissy, M. and Youssef, M.: Towards evolving Sensor Actor NETWORKs, *Computer Communications Workshops*, 2008. *INFOCOM. IEEE Conference on*, pp. 1–6 (2008).
- [6] 高草 英博, , 磯井正義: LONWORKS 入門—ビル制御システムのオープン化, 日刊工業新聞社 (2001).
- [7] Fok, C., Roman, G. and Lu, C.: Mobile agent middleware for sensor networks: an application case study, *Information Processing in Sensor Networks*, 2005. *IPSN 2005. Fourth International Symposium on*, pp. 382–387 (2005).
- [8] Gu, L. and Stankovic, J.: t-kernel: providing reliable OS support to wireless sensor networks, *Proceedings of the 4th international conference on Embedded networked sensor systems*, ACM Press New York, NY, USA, pp. 1–14 (2006).
- [9] Finne, N., Eriksson, J., Dunkels, A. and Voigt, T.: Experiences from Two Sensor Network Deployments-Self-monitoring and Self-configuration Keys to Success, *LECTURE NOTES IN COMPUTER SCIENCE*, Vol. 5031, p. 189 (2008).
- [10] Hartung, C., Han, R., Seielstad, C. and Holbrook, S.: FireWxNet: a multi-tiered portable wireless system for monitoring weather conditions in wildland fire environments, *Proceedings of the 4th international conference on Mobile systems, applications and services*, ACM New York, NY, USA, pp. 28–41 (2006).
- [11] Cha, H., Choi, S., Jung, I., Kim, H., Shin, H., Yoo, J. and Yoon, C.: RETOS: resilient, expandable, and threaded operating system for wireless sensor networks, *Proceedings of the 6th international conference on Information processing in sensor networks*, ACM Press New York, NY, USA, pp. 148–157 (2007).
- [12] Wark, T., Crossman, C., Hu, W., Guo, Y., Valencia, P., Sikka, P., Corke, P., Lee, C., Henshall, J., Prayaga, K. et al.: The design and evaluation of a mobile sensor/actuator network for autonomous animal control, *Proceedings of the 6th international conference on Information processing in sensor networks*, ACM Press New York, NY, USA, pp. 206–215 (2007).
- [13] Yick, J., Mukherjee, B. and Ghosal, D.: Wireless sensor network survey, *Computer Networks*, Vol. 52, No. 12, pp. 2292–2330 (2008).