

無線可視領域通信を支援する相対位置情報を利用した 名前解決ミドルウェア

野田 厚志[†] 北須賀輝明^{††} 田頭 茂明[†] 中西 恒夫[†]
福田 晃[†]

A Name Resolution Middleware Using Relative Location Information
Supporting Wireless Visible Area Communication

Atsushi NODA[†], Teruaki KITASUKA^{††}, Shigeaki TAGASHIRA[†], Tsuneo NAKANISHI[†],
and Akira FUKUDA[†]

あらまし 本論文では、無線可視領域通信 (WVAC: Wireless Visible Area Communication) において通信相手の特定を支援する名前解決ミドルウェアを提案する。WVAC とは、近距離無線通信デバイスを用いて、ユーザの視界内に存在する端末と、一時的にネットワークを形成し、即座に情報交換を実現する無線通信のことである。提案ミドルウェアは、従来の名前を用いることに加えて、周辺端末との相対位置を補助的に提示することで、WVAC 環境における通信相手の特定を効果的に支援する。提案ミドルウェアは、(1) ミドルウェアの機能を、アプリケーションが汎用的に利用できるように、シンプルな API を提供する。(2) 事前に設定/配置が必要な専用サーバを必要としない。(3) キャリブレーションを必要とせずに、周辺端末の相対位置情報を取得できる測位手法を採用している。また、提案ミドルウェアのプロトタイプシステムを構築し、応用プログラムの作成と基礎的な評価を行った。評価の結果から、PDA 程度の処理能力で十分に提案ミドルウェアを稼働できることを示した。キーワード 無線通信, 可視領域, 名前解決, 相対位置

1. ま え が き

あらゆる機器や装置が通信デバイスを装備し、それがネットワークで有機的に結合されるユビキタス計算環境が現実のものになりつつある。特に、無線通信デバイスによる接続は、その利便性の高さから、ユビキタス計算環境の核となるデバイスとして、様々な分野からの研究開発が進められている。我々は、無線通信デバイスの中でも Bluetooth や無線 LAN のアドホックモードなどのインフラを必要としない近距離無線デバイスに着目している。また、このような近距離無線通信デバイスを用いて、ユーザの視界内に存在する端末と、一時的にネットワークを形成し、即座に情報

交換を実現する無線可視領域通信 (WVAC: Wireless Visible Area Communication) を構築し、WVAC を支援するネットワーク技術について研究している。

従来のネットワーク技術では、WVAC 環境が考慮されておらず、非常に煩わしい手順が必要である。特に、通信相手を特定する技術は、このような WVAC 環境の利便性を左右する重要な要素技術であるために、従来のネットワーク上の仕組みをそのまま流用することは適当でない。WVAC は、見えている端末との通信を実現することが目的である。すなわち、見えることが通信のトリガとなるために、その端末の物理的な配置をユーザの目で確認できることが WVAC の前提となっている。このことから、端末の配置情報 (すなわち端末との相対的な位置情報) が通信相手の特定を、より直感的に補助する重要なかぎとなると考えられる。

本論文では、WVAC 環境において通信相手の特定を支援する名前解決ミドルウェアを提案する。具体的には、提案ミドルウェアは、従来の名前を用いることに加えて、周辺端末との相対位置を補助的に提示し、

[†]九州大学大学院システム情報科学府・研究院, 福岡市
Graduate School/Faculty of Information Science and Electrical Engineering, Kyushu University, 744 Motooka, Nishi-ku, Fukuoka-shi, 819-0395 Japan

^{††}熊本大学大学院自然科学研究科, 熊本市
Graduate School of Science and Technology, Kumamoto University, 2-39-1 Kurokami, Kumamoto-shi, 860-8555 Japan

WVAC 環境における通信相手の特定を効果的に支援する。提案ミドルウェアを利用することにより、アプリケーションは周辺端末の名前とともにその物理的な配置を画面に表示することができる。このため、ユーザは周辺端末の配置と画面上の端末の配置を照らし合わせ、目的の端末を直感的に選択することが可能となる。提案ミドルウェアは、次の三つの特徴を有する。(1) 提案ミドルウェアの機能を、アプリケーションが汎用的に利用できるように、シンプルな API (Application Program Interface) を提供する。(2) WVAC 環境では専用サーバが配置されていることを必ずしも期待できないため、サーバ不在の環境でもミドルウェア自らがサーバの役割を担い、PDA 程度のサーバ処理能力でも小規模なサービスを提供する。(3) キャリブレーションを必要とせず、周囲の端末の相対位置を取得可能である WiPS (Wireless LAN indoor Positioning System)[1] をベースにした測位手法を採用している。また、提案ミドルウェアのプロトタイプシステムを構築し、応用プログラムの作成と基礎的な評価を行った。評価の結果から、PDA 程度の処理能力で十分に提案ミドルウェアを稼働できることを示した。

本論文の構成は以下のとおりである。2. で WVAC について述べ、3. で提案ミドルウェアの設計と各構成要素の詳細について述べる。4. で位置計算処理にかかる時間のシミュレーション結果や無線 LAN を利用した応用例を示し、提案ミドルウェアの実現性、有用性について評価する。5. で関連研究について述べ、最後に 6. で本研究をまとめる。

2. Wireless Visible Area Communication

本章では WVAC の概要を説明し、WVAC における通信相手の特定について議論する。

2.1 概要

WVAC とは、ユーザの視界内に存在する端末とネットワークを形成し、即座に情報交換を実現する無線通信のことである。WVAC では、アクセスポイントなどのバックボーンネットワークの存在しない環境においても、機器が局所的な無線ネットワークを構築し、端末同士がシングルホップ通信することを想定している。このようなネットワークを形成できる無線通信デバイスを、様々な装置や機器が装備しているユビキタス計算環境において、WVAC の有効な利用シナリオとして、例えば、以下のようなものを挙げることで

きる。

- 不特定多数が参加する会議において、目の前に座っている参加者の端末に会議の資料を渡したい。
- オフィスの中に複数のプリンタが存在している。このような環境で、ユーザ A の近くにあるプリンタから書類を印刷したい。

このような見えることが通信のかぎとなる WVAC 環境の構築が、様々な状況で今後必要とされることが予想される。しかしながら、現在のネットワーク技術では WVAC 環境が考慮されておらず、非常に煩わしい手順が必要である。具体的には、上記のような状況において、すぐ目の前にある端末と通信する場合でも、遠く離れたユーザと通信する場合と同じように通信相手の IP アドレスやメールアドレス等といった通信識別子を知る必要がある。近くの端末との通信の場合、周辺端末の通信識別子を自動的に取得する仕組みを用いることで、相手の通信識別子の入力の手間は省くことができるが、通信識別子と実際の端末との対応を確認する(通信相手を特定する)手間が必要であることに変わりはない。通信相手の特定は、このような WVAC 環境の利便性を左右する重要な要素技術であるために、従来のネットワーク上の仕組みをそのまま流用することは適当でない。このために、本論文では通信相手の特定方法に着目し、WVAC 環境における効果的な特定方法について探求する。

2.2 通信相手の特定

WVAC 環境において、通信相手を特定する方法の一つとして、名前による特定を挙げることができる。この方法は、ユニークな名前が各端末にあらかじめ付与されており、その名前を利用することで端末を一意に特定することができる。特に、インターネットにおける DNS (Domain Name System) では、この名前として、人が記憶しやすい単語が利用されており、所属するドメインが基準となり階層的に構造化されたものが採用されている。しかしながら、頻繁に利用する名前は、その名前と対応する端末との関連を記憶しておくことができるが、まれにしか利用しない端末などでは、それらを正確に記憶しておくことは困難である。また、一時的に構築されるネットワークを想定している WVAC では、その名前を周知することが困難である。

名前による特定方法を補助する目的で、LAN 環境では端末の名前リストを用いる方法が活用されている。すなわち、LAN 内に存在する端末の名前リストを自

動的に作成し、その名前リストから通信相手を特定する方法である。この方法では、内部では名前で端末を特定することには変わらないが、正確な名前を記憶しておく必要がない点で、LAN 環境における通信の利便性を高めている。しかしながら、名前リスト上での端末の表示は、物理的な端末の配置とは関係なしに、例えば名前を辞書式にソートしただけのものが多く、その名前リストの中から通信相手を探すのが非常に煩わしい操作となっている。

WVAC は、見えている端末との通信を実現することが目的である。すなわち、見えることが通信のトリガとなることから、その端末の物理的な配置をユーザの目で確認できることが WVAC の前提となっている。このことから、端末の配置（相対位置）が通信相手の特定を、より直感的に補助する重要なかぎとなると考えられる。また、通信相手の端末との相対位置だけでなく、他の周辺端末との位置関係もその端末の特定には非常に重要な情報であるといえる。したがって、名前リストの端末を一覧提示するだけでなく、図 1 のように、更に通信相手の端末との相対位置と、その端末と周辺端末との位置関係を提供することにより、WVAC 環境においてより自然な通信相手の特定方法を実現できる。また、DNS のような絶対的なドメインとは異なり、位置情報に関して相対的なドメインで構造化できれば、更なる利便性の向上が期待できる。すなわち、NEARBY（近隣を表すドメイン）や SPECIFIC（ある特定のエリアを指すドメイン）のような相対的なドメインを導入し、相対位置に関して端末をグループ化して提示することで、通信特定をより容易に実現できると考えている。

一方、上記の相対位置情報を利用した通信相手の特定方式は、通信における送信側だけでなく、受信側においても有効であると考えている。例えば、WVAC 環境である送信者からメッセージが届いた場合を考える。受信者は送信者の名前を見ただけでは、だれからのメッセージなのか特定できない。しかしながら、名前に加えて送信者の相対位置やその周辺端末との位置関係を把握できるならば、送信者を特定することが容易になる。また、WVAC では、これらの情報をアクセス制御へ応用することも考えられる。例えば、NEARBY ドメインに存在する端末からしか通信を受け付けられないなどがある。

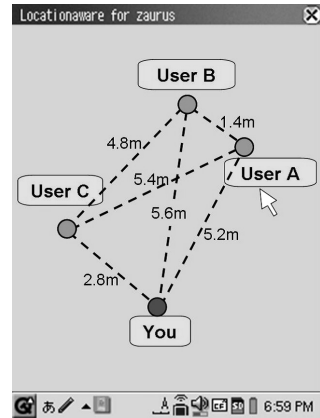


図 1 相対位置を用いた端末特定のイメージ図

Fig.1 Illustration for specifying a target host using relative location.

3. 提案ミドルウェア

本章では、2.2 で説明した相対位置情報を利用した名前解決ミドルウェアの設計とその詳細構成について説明する。本研究では、端末間通信を実現する無線ネットワークとしては IEEE802.11 規格に従う無線 LAN を想定している。無線 LAN は広く普及しており、利用が容易であるという利点をもつ。更に、既に多くの携帯端末に標準的に装備されている。

3.1 概要

提案ミドルウェアの主な機能は、WVAC 環境において、ネットワークに参加する端末の相対位置、名前、及び通信識別子（例えば IP アドレスや MAC アドレス）の組を管理し、相対位置情報と名前、及び名前と通信識別子との相互変換を可能にする。提案ミドルウェアでは、各端末には名前及び通信識別子があらかじめ付与されていることを想定し、相対位置に関しては、無線 LAN により電波強度から推定する。WVAC 環境で利用するために、提案ミドルウェアは以下の特徴を有する。

- (1) シンプルな API を提供する。
- (2) 専用サーバを必要としないシステム構成を採用する。
- (3) 事前設定を必要としない端末間の相対位置推定手法を採用する。

(1) について詳しく説明する。提案ミドルウェアの名前解決機能を、アプリケーションが柔軟に利用できるように、ミドルウェアの API を設計する必要があ

る．具体的には，名前と通信識別子との間，及び相対位置と名前との間の変換を提供する．端末の名前と通信識別子との間の相互変換は，従来の名前解決と同様の機能であるために，API は，伝統的な API を参考にすることができる．しかしながら，相対位置と名前との間の相互変換は，本提案システムの特徴的な部分であり，汎用的な API を考察する必要がある．

(2) については，2. で述べたように，WVAC 環境はインフラを必要としない環境を想定している．このため，名前解決に必要な専用サーバをあらかじめ配置することができない．提案ミドルウェアはこのような専用サーバを自動的に決定する仕組みを提供する必要がある．また，PDA 程度の処理能力でも動作が可能ないようにシステムを設計する必要がある．

(3) については，WVAC 環境は一時的に形成されるネットワークを想定している．このために，事前に設定が必要な要素技術は利用できない．特に，端末間の相対位置の推定に関して，既存の位置推定方式の多くは，事前のキャリブレーションや，基準点の存在が前提となっている．このために，事前設定が必要のない相対位置の推定方式を検討する必要がある．提案ミドルウェアでは，基準点を必ずしも必要としない WiPS の測位方式に着目している．WiPS がキャリブレーションを必要とせず，ある程度の精度を発揮することがシミュレーションにより確認されている [1]．また，計算負荷が低く，端末同士の位置の並びが正しく判定されやすいなどの特徴をもつため，WiPS の測位手法は，本ミドルウェアに必要な端末間の相対位置を推定するのに適しているといえる．しかしながら，WVAC 環境に WiPS 方式をそのまま適用することはできないために，WiPS 方式を改良する必要がある．

次節からは以上で述べた 3 点に関して，本ミドルウェアがどのように解決したのかを詳しく述べる．

3.2 アプリケーションインタフェース

提案ミドルウェアが提供する API は，名前と通信識別子との間の変換，及び相対位置情報と名前との間の変換に大きく分けることができる．それぞれについて詳しく説明する．図 2 に端末の通信識別子，名前，相対位置の変換を行う API の概略を示す．これら四つの API により，通信識別子，名前，相対位置の変換がシンプルに実装されている．名前と通信識別子との間の変換は，伝統的な C 言語の API である `gethostbyname` 及び `gethostbyaddr` を踏襲している．具体的には，以下のとおりである．

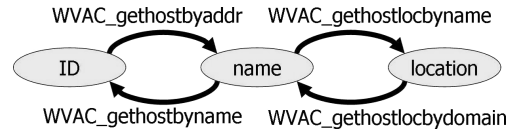


図 2 端末の通信識別子，名前，相対位置の変換
Fig. 2 Translation among identifier, name, and location.

`WVAC_gethostbyname`：端末の名前を引数にとり，値が代入された `hostent` 構造体が返される．`hostent` 構造体には，端末の名前とその通信識別子が含まれる．

`WVAC_gethostbyaddr`：通信識別子を引数にとり，値が代入された `hostent` 構造体が返される．

次に，相対位置と名前との間の変換について説明する．相対位置情報から名前の変換は，特定の座標から端末の名前に変換するのではなく，相対位置指定ドメインを導入し，そのドメインにマッチした端末の名前が返される．一方，名前から相対位置への変換は，単純に名前に対応する相対位置を返す API を提供する．相対位置指定ドメインとしては，以下の 4 種類を定義する．

ALL ドメイン：ある端末にとっての one-hop 以内の領域

NEARBY ドメイン：端末を中心にして周辺 $r(> 0)$ メートル内の領域．

SPECIFIC ドメイン：座標 (x, y) を中心にして周辺 r メートル内の領域．

PINPOINT ドメイン：座標 (x, y) をピンポイントに指定する領域．その座標に端末が存在しない場合は，最も近い端末が返される．

以下に，相対位置と名前との間を変換する API の詳細を記述する．

`WVAC_gethostlocbydomain`：相対位置指定ドメインを引数にとり，そのドメインにマッチする端末が `hostloc` 構造体の形で返される．`hostloc` 構造体には，端末の名前とその相対位置が含まれる．

`WVAC_gethostlocbyname`：端末の名前を引数にとり，その相対位置が `hostloc` 構造体の形で返される．

`WVAC_updatelocation`：提案ミドルウェアでは，端末間の相対位置情報は，ミドルウェアが保持するだけでなく，各アプリケーションでバッファリングする構成をとる．このバッファを提案ミドルウェアがもつ最新の情報に更新する．

`WVAC_rotateLocation`，`WVAC_mreverseLocation`：これらの API の説明は，3.4.2 で説明する．

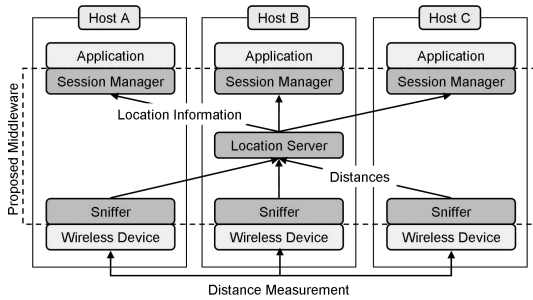


図 3 ミドルウェアを構成するサブシステム

Fig. 3 Subsystems in the proposed middleware.

3.3 システム構成

提案ミドルウェアは、役割別に三つのサブシステムから構成される。すなわち、位置計算サーバ、スニファ、セッションマネージャである。サブシステムの関係を図 3 に示す。このうち、スニファとセッションマネージャはすべての端末上で動作し、位置計算サーバは、位置推定サービスを提供する端末でのみ動作する。ホスト間の通信は無線通信によって行われる。我々は、見通しの効く範囲という極限られた範囲での通信を対象としており、すべての端末が直接通信可能であることを前提としている。このため、すべての端末で同じ計算をする無駄を避ける意味で代表となる端末が集中して位置計算を行う形式を採用している。また将来的には、計算資源や消費電力に余裕のある端末が位置計算を担う仕組みを備えることで、端末数が多い場合にも適切な負荷分散を図る予定である。位置計算サーバによって計算された位置情報は、各端末に配布され、アプリケーションから利用可能となる。各サブシステムの詳細について記述する。

スニファは、自端末と周囲の端末との距離を測定し、自端末の通信識別子、自端末の名前、及び周囲端末との距離を位置計算サーバへ周期的に通知する役割を担う。自端末は他の端末の通信を常に観測し、その通信の電波強度から距離を測定する。ここで、すべての端末間で距離を測定するためのハローメッセージを投げ合うと、通信トラフィックが増大してしまう。このため、無線 LAN デバイスのプロミスキャストモードを用いて他の端末あてのユニキャストパケットも常に受信し、距離測定に用いる。距離測定のためのハローメッセージは、一定期間通信の行われなかった端末のみが送信する。位置計算サーバは、ネットワークに参加している端末の名前と通信識別子を収集し管理する。新たな

端末の追加や削除により、この情報の更新があったときに、すべての端末に最新の端末情報を配布する。この端末情報は、各端末のスニファで利用される。つまり、端末情報に記載された端末との距離が測定されることになる。また、位置計算サーバの別の仕事として、各端末のスニファから端末間距離情報を周期的に収集し、それらの情報から各端末の相対位置を計算する。得られた相対位置は各端末のセッションマネージャへ配布されることになる。セッションマネージャは位置計算サーバから情報を受信し、それらの情報をキャッシュしておく。このキャッシュ情報は、バッファを通じてアプリケーションに提供する。位置計算の周期はアプリケーションの要求するリアルタイム性に従って適当な周期とする。また、スニファがハローメッセージや距離情報を送る周期もそれと同一の周期とする。例として、4.1 で述べる応用アプリケーションでは、1秒を周期として設定している。スニファは 1 周期の間に同じ端末からの距離情報が複数得られた場合、その平均値を位置計算サーバに通知する。

提案ミドルウェアでは、位置計算サーバをすべての端末の中から自動的に選出する必要がある。このため、位置計算サーバを最も早く起動した端末が選ばれるようにしている。以下に、位置計算サーバを選出する具体的な手順を示す。

(1) 各端末は、位置計算サーバを探索するために、ミドルウェア起動時に特定のポートに対してブロードキャストする。

(2) 位置計算サーバから応答がなかった場合、その端末は自身の端末に位置計算サーバを立ち上げ、手順 3 に進む。位置計算サーバからの応答があった場合は、何もせずにそのまま手順 3 に進む。

(3) 端末は、自分の名前と通信識別子を位置計算サーバに登録し、位置計算サーバから周囲の端末の通信識別子を受け取る。

(4) 位置計算サーバは既にサービスに登録されている他の端末らに対し、現在登録されている端末の名前と通信識別子を通知する。

複数の端末から位置計算を担う 1 台を選出するのは、リーダ選出問題の一つととらえることができ、従来の解決方法が利用できる。現状では、実装の簡便さから最初に起動した 1 台がサーバを担う仕組みとしている。

ネットワークへの端末の参加や離脱を考慮すると、サーバの欠落及び一つのネットワークにサーバが複数存在する状況が考えられる。サーバが欠落した場合、

各端末は位置情報が一定時間送られてこないことでサーバの欠落を検知することができる。この場合、各端末はランダムな時間待ち、再度サーバを探索する上記の手順を行う。ランダムな待ち時間を設けることで、同時に複数のサーバが立ち上がる可能性を低く抑えている。一つのネットワークにサーバが複数存在する場合、スニファはすべてのサーバに距離情報を通知し、セッションマネージャはどれか一つのサーバから位置情報を受け取る。サーバを担う複数の端末が同じ計算を行うため冗長ではあるが、ミドルウェアを利用するアプリケーションへの不都合は特に生じない。

3.4 WiPS をベースにした相対位置の推定方式

提案ミドルウェアにおける相対位置の推定方式について述べる。

3.4.1 WiPS の概要

WiPS は、端末の相対位置を計算し、いくつかの基準点を用いて絶対位置を推定する位置推定システムである [1]。WiPS は、すべてのホスト間の計測距離と推定距離との累積誤差を最急降下法を用いて最小化することで、端末間の相対位置を決定している。従来の方式が基準点と各端末との距離のみから位置を推定していたのに対し、WiPS ではすべての端末間の距離を利用することで精度の向上を図っている。WiPS の位置計算の主な特徴としては、端末密度が高いほど精度が高くなること、既存の他の方式と比べて短時間で位置を計算できることなどが挙げられる。基本的な位置計算の手順は以下のとおりである。

- (1) 任意の 2 端末間の距離を収集する。
- (2) 各端末位置の初期推定を行う。
- (3) 累積誤差の差が収束するまで、端末の位置の修正を繰り返す。
- (4) 最終的な結果を、端末の位置とする。

特に、位置計算の周期に対して、ノードの移動があまり激しくない場合には、最急降下法における繰返し計算の初期値として前回の計算結果を用いることで計算時間の短縮を図ることが可能である。

この WiPS の位置推定方式を WVAC 環境で用いる場合に問題が発生する。上述したように、WiPS は求めた相対位置に対して基準点を用いて絶対位置を計算し、提供している。しかしながら、WVAC 環境では基準点の存在を必ずしも仮定できないため、求めた相対位置には、絶対的な方位、ユーザの向き、及び左右の情報が欠落している可能性があるという課題がある。これらの方向情報の欠落は、位置情報利用アプリケー

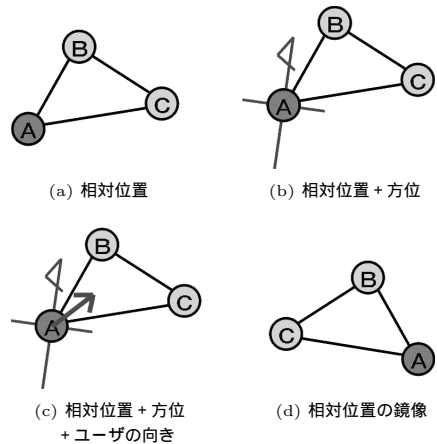


図 4 相対位置、絶対位置とユーザの向き

Fig.4 Relative location, absolute location, and user's direction.

ションの可用性を低下することにつながる。次項ではこの課題について詳しく述べ、その解決策について言及する。

3.4.2 相対位置情報の修正

WVAC 環境における WiPS で求めた相対位置情報の課題について説明する。例えば、端末間の相対位置が図 4(a) のような場合を考える。図 4(a) は端末間距離情報のみから得られる可能解の一つであるが、これには方向情報と左右情報が欠落しているため、ユーザが認識している端末の位置関係とは一致しない場合がある。方向情報には、(1) 北を基準とする方位と (2) ユーザ自身の向き (正面方向) の 2 種類がある。方向情報 (1) は周辺端末位置とともに方位を表示する場合 (図 4(b)) に必要である。方向情報 (2) は、ユーザが自身から見てある方向の端末と通信したいとアプリケーションに伝えたい場合 (図 4(c)) に必要になる。(1) の方位を知るためには三つ以上の絶対位置が既知である基準点の存在が必要である。更に (2) を求めるためには (1) に加えてユーザの正面の方位の情報が必要である。(2) は (1) と無関係に周辺端末がユーザ自身のどちらの向きにあるかという情報から決定することができる。また (1) を利用して (2) を求めることも可能で、その場合はユーザの正面方向の方位、つまりユーザの正面方向が北に対して時計回りに何度傾いているかという情報があればよい。

複数の基準点が存在しない場合、本ミドルウェアはこれら二つの方向情報を提供できない。

左右情報の欠落とは、例えば目の前に二人の携帯端末所有者がいる場合に、二人がそれぞれ左右どちらに居るのかをその情報から判断できないということである。鏡像になっていることが判断できないという言い方もできる(図4(d))。左右情報は位置測定平面と実際の平面が左右反転することで発生しているため、ユーザごとに異なるわけではなく、すべてのユーザに共通して発生する。この欠落は、三つ以上の方位が既知である基準点が存在しない場合に発生する。

本ミドルウェアでは、端末間距離を利用して位置測定を行う。三つ以上の基準点が測位に参加する場合は方向情報(1)や左右情報が得られるが、基準点数が不足する場合はこれらの情報を得られない。また、基準点数が満たされたとしても方向情報(2)は自ら判断できる情報をもたない。このような方向情報と左右情報の欠落によって位置情報利用アプリケーションの可用性が低下することになる。

絶対位置が既知である基準点が一定以上存在しない場合は、ユーザの補助なしでは方向情報(1)、(2)と左右情報のいずれも獲得できない。WVAC環境においてはこのうち方向情報(2)と左右情報が重要であると考え、提案ミドルウェアにユーザがこれらの情報を入力する機能を付加することで可用性の低下を抑える。左右情報は、一人のユーザの入力をミドルウェア全体に反映することで、複数のユーザが同様の入力をする負担を抑える。方向情報(2)と左右情報をユーザが入力する手段を述べる。

方向情報(2)に関しては、アプリケーション画面上の周辺端末の方向とユーザが目視した周辺端末の方向のずれを、ユーザ自身が把握できたときに、ユーザが入力する手段を用意する。具体的には、3.2で記述したWVAC_rotateLocationである。セッションマネージャはユーザが入力したずれ角 θ を記憶し、以降、位置計算サーバからの周辺端末の位置情報を受信するたびに、ずれ角 θ 分の回転を加えた情報を提示する。これによりユーザが向きを変えるなどでずれ角が変化するまでの間はユーザは正しい方向で画面上の端末位置を確認できる。

左右情報は、ユーザ個別の情報ではなく全ユーザに共通した情報である。そのため、あるユーザが左右反転してしまっている状況を見出し修正した場合は、全ユーザに通知する位置測定結果を反転するよう設計する。あるユーザがアプリケーション画面上で実際の左右と画面上の左右の反転(鏡像)に気づいた

場合、アプリケーション経由で本ミドルウェアのセッションマネージャに反転の指示を出す。具体的なAPIは、WVAC_mreverseLocationである。サーバから送信される相対位置情報には、あらかじめ1ビットの裏表情報が含まれている。セッションマネージャは正しい裏表情報を位置計算サーバに伝えることで、左右情報の修正内容を伝える。以降、位置計算サーバはすべての端末のセッションマネージャに対して裏表情報の修正された相対位置を通知する。

4. 評価

本章では、提案ミドルウェアの評価について述べる。評価のために提案ミドルウェアのプロトタイプシステムを構築した。プロトタイプシステムは、PDA上で動作を可能にするために、ARMプロセッサをターゲットにし、LinuxのデーモンとしてC言語で実装した。次節からは、そのプロトタイプシステムを用いて、応用ソフトウェアの構築及びシステムの性能評価を行う。

4.1 位置情報を使ったメッセージ通信ソフトウェア

本ミドルウェアの応用例としてチャット機能を有した通信アプリケーションを作成した。アプリケーションの実行画面を図5に示す。このアプリケーションでは、ミドルウェアが提供する周囲の端末の相対位置情報をグラフィカルに表示し、その画面上で端末の名前と位置情報を確認して、メッセージの送信対象となる端末を特定する。また、3.4.2で述べた相対位置情報の修正操作をGUIを介して(ドラッグにより)、実施

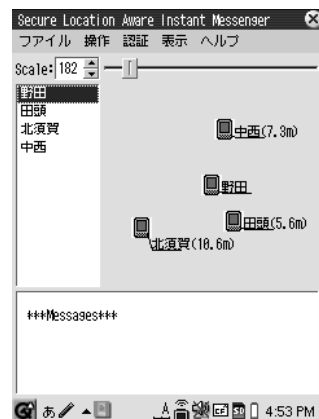


図5 位置情報を使ったメッセージ通信ソフトウェアの実行画面

Fig. 5 Screenshot for a messenger software using the proposed middleware.

```

if ( WVAC_updatelocation() ) { /* 相対位置情報を更新 */
/* WVAC_getnamesbydomainにより周囲の端末の名前を取得 */
for ( 周囲の端末数だけ繰り返す ) {
/* WVAC_getlocationbynameにより相対位置情報を取得 */
/* 端末の名前と相対位置情報を用いて画面に出力 */
}
}

```

図 6 位置情報を使ったメッセージ通信ソフトウェアの擬似コード

Fig. 6 Pseudo code for a messenger software using the proposed middleware.

することができる。

このアプリケーションでは、提案ミドルウェアが提供する API を図 6 のように利用している。このコードは、周囲端末の位置情報を取得し、表示する部分だけを抜粋している。

4.2 性能評価

提案ミドルウェアの性能評価を行った。ミドルウェアを動作させるのに必要な計算資源（メモリ、通信量）、相対位置を計算するのに必要な計算時間について評価した。これにより、携帯端末程度の処理能力で提案ミドルウェアを十分に稼働できることを示す。また、無線 LAN の距離測定誤差を軽減する手法を適用し、その効果を実測で確かめた。

4.2.1 必要な計算資源と通信量

本評価では、ミドルウェアを動作させるために必要な計算資源としてメモリ使用量を測定した。用いた機器は、SHARP Zaurus SL-C1000 であり、Intel® XScale™ (PXA270 416 MHz) RAM 64 MByte である。実験は位置計算サーバを担う端末として実機を 1 台だけ用い、他の端末は仮想的に作り出す。本来は他の端末から送られてくる距離情報は、位置計算サーバのソフトウェア内で乱数を使って擬似的に用意している。擬似的な端末の数を 5, 10, 20, 40 台と変化させて、位置計算サーバにおける計算資源を測定した。このため、擬似的に用意した距離情報用のメモリ使用量については、あらかじめ減算した値を測定値としている。擬似的に生成した距離情報は、次のような端末配置を想定して用意している。端末は 100m 四方の平面上にランダムに配置されており、各端末の無線 LAN デバイスの到達距離は十分に長いものとし、全領域で通信が可能とする。また、無線 LAN デバイスの 2 点間の距離の測定誤差は ±10% 以下になる確率が 68.26%、±20% 以下になる確率が 95.44% となるよう

表 1 位置計算サーバに必要な計算資源

Table 1 Resources required for calculating relative locations.

端末数	メモリ使用量 [kByte]
5	3700
10	3704
20	3716
40	3764

な正規分布に従うものとした。測定値は 10 秒間の平均をとった。

メモリ使用量の結果を表 1 に示す。まず注目する点は、端末数の増加に対して、メモリ使用量の増加が少ないことである。具体的には、端末数が 5 台から 40 台に増加してもメモリ使用量は 2% 程度増加しただけである。また、値の絶対値を見てみると、約 3.7 [MByte] 程度使用している。近年の多くの携帯端末は数十から数百 MByte 以上のメモリを搭載しており、これは実用的な数値といえる。

また、位置計算サーバの通信量を計算により求めた。ミドルウェアにおいて、各端末が位置計算サーバに送信するデータ量は、1 回当り他の端末数 × 8 [Byte] である。端末数が 10 台の場合、これが 1 秒間に 10 回送られてくるので、スニファ、位置計算サーバ間の通信量は、 $9 \times 8 \times 10 = 720$ [Byte/s] である。位置情報のリクエストメッセージサイズは 10 [Byte]、配布する位置情報は、端末数 × 76 [Byte] である。端末数が 10 台、各端末が 1 秒間に 1 回のリクエストを送ってくる場合、位置計算サーバ、セッションマネージャ間の通信量は、 $(10 + 10 \times 76) \times 10 = 7700$ [Byte/s] である。よって端末数 10 台の場合、総通信量は $720 + 7700 = 8420$ [Byte/s] である。通信量のオーダは端末数の二乗であり、端末数 40 台の場合でも総通信量は 135 [kByte/s] + プロトコルオーバーヘッド程度と少ない。また、情報を圧縮することで更に低減できであろう。

4.2.2 相対位置計算に必要な時間

位置計算サーバにおける位置計算処理にかかる時間を測定した。今回の測定では携帯端末として、B5 サイズのノート PC と PDA を用いて測定を行った。測定に使用したノート PC は、Panasonic Let's note CF-R4 であり、CPU は Intel® Pentium® M 1.20 GHz RAM 1 GByte である。また、PDA は、SHARP Zaurus SL-C1000 であり、Intel® XScale™ (PXA270 416 MHz) RAM 64 MByte を使用した。その他の端

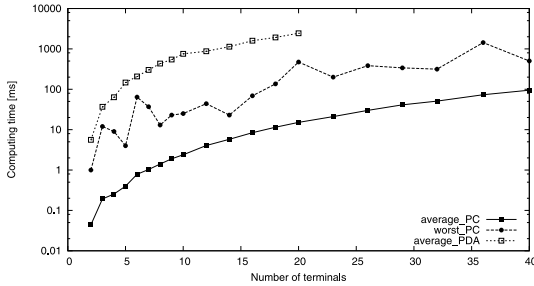


図 7 位置計算処理時間

Fig. 7 Computation time required for estimating relative locations.

表 2 携帯端末での位置計算処理時間 [ms]

Table 2 Computation time required for estimating relative locations [ms].

端末数	ノート PC 初回	ノート PC 2 回目	PDA 初回	PDA 2 回目
5	0.43	0.35	104	90
10	2.5	1.8	523	361
20	15	9	2088	1278
40	98	53	-	-

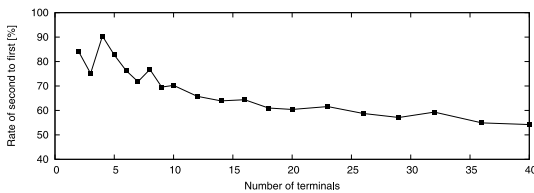


図 8 2 回目以降の計算時間の初回に対する割合

Fig. 8 Ratio of second computing time to first one.

末は、4.2.1 と同じようにダミーの情報を利用した。ノート PC では端末数を 2~40 の間で、PDA では端末数を 2~20 の間で変化させて、各端末において 1000 回ずつ実行時間を測定し、平均値を求める。ノート PC では併せて最悪実行時間も計測する。

測定結果を図 7、及び表 2 に示す。2 回目以降の位置計算では、初回の計算結果を繰返し計算の初期値とすることで、位置計算にかかる時間が初回よりも少ない。図 8、及び表 2 に初回計算にかかる時間と 2 回目の計算にかかる時間の比較を示す。2 回目の計算では端末の位置は変えずに距離測定誤差のみ再度付加し直している。

2 回目以降の計算にかかる時間は、ノード数が増えるにつれて初回計算にかかる時間よりも少ない割合であることが見て取れる。処理能力の低い PDA であっても、端末数が 10 台程度であれば初回の位置計算が 0.52 [s]、2 回目以降の計算が 0.36 [s] で行える。この

表 3 想定規模の比較

Table 3 Resources required for calculating relative locations.

位置提供ミドルウェア名	想定端末数
提案ミドルウェア	数十台程度
Ekahau Positioning Engine 4.2	20000
AirLocationII ステーションマネージャ	最大 10000

ため、位置計算にリアルタイム性がそれほど求められない 10 名程度の小規模な会合などでは、PDA を位置計算サーバに利用し 1 秒周期の測位サービスを提供できると考える。端末数が更に多い場合や、高いリアルタイム性が求められる場合には、PDA よりも高機能な端末を位置計算サーバとして用意する必要がある。ノート PC の処理能力があれば、数十台の携帯端末に対して 1 秒周期の測位サービスを提供するのに十分である。

本ミドルウェアは、インフラが存在しない状況ではその場の端末だけで互いの位置を計算する手法をとっている。専用の設備や特殊なデバイス、事前のキャリブレーションを必要としない利点は大きいと考えるが、すべての端末間の距離をもとに繰返し計算により位置を求める本手法のアプローチは、他の手法に比べるとオーバーヘッドが大きい。1 台の位置計算サーバでサポートできる端末数の他の手法との比較を表 3 に示す。ミドルウェアを運用する計算機は、CPU : Intel Pentium 4、メモリ : 1 GByte RAM 程度を想定している。我々の提案ミドルウェアでサポートできる端末数は、Ekahau [2] や AirLocation [3] のそれと比較すると 100 分の 1 以下である。Ekahau では専用のサーバ計算機を用いて 1 秒当り 600 端末の位置が推定できるとしている。本ミドルウェアでは、それと同程度の計算機でも 100 台程度であり、我々の想定している計算能力の低い携帯端末上では、より少ない数十台程度しかサポートできない。数十台の端末がサポートできれば、WVAC 環境のアプリケーションの一部は構築可能であると考えられるが、より大規模な WVAC 環境に対しては本ミドルウェアの改良が必要といえる。改良の一案として位置計算サーバの分散化があるが、改良手法は今後の課題とする。

4.2.3 端末間距離のノイズ軽減

無線 LAN の受信信号強度は、障害物のない理想の環境下においては距離の 2 乗に比例して減衰することが知られている。このため、他の端末から受信した無線の受信信号強度から端末間の距離を求めることがで

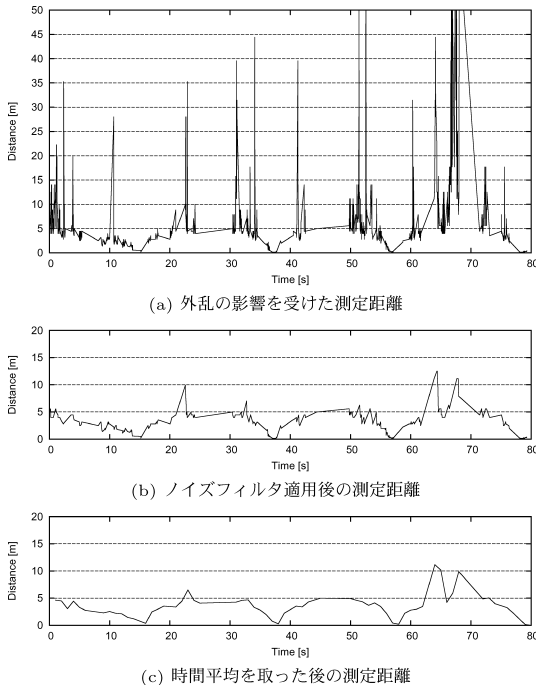


図 9 測定距離における外乱の影響の軽減

Fig.9 Noise reduction for measured distances.

きる．しかし実世界においては，障害物の影響により受信信号強度に大きなばらつきが発生する．

本ミドルウェアでは，外乱による誤差の影響を抑える手法として，人の動きに即さない急激な距離の変化をスニファでフィルタリングしている．図 9(a) に実際に測定した無線 LAN の受信信号強度から得られた距離を示す．この測定では，1 台の端末の位置を固定し，もう 1 台の端末を人間が持ち，2 台の端末間距離が 0~5[m] の範囲で移動することを繰り返して測定を行った．測定した値を見ると，実際の距離よりも大きく離れた値が頻りに計測されているのが分かる．これらは外乱の影響によるノイズである．このノイズを軽減するために，急激な距離の変化，特に端末間が離れる方向の変化に対してフィルタリングを行った．人の歩行速度は通常 1.0~1.5[m/s] 程度である．そこで，移動速度のしきい値を設け，直前の測定距離と新しい測定距離から求めた移動速度がこのしきい値を超えていた場合は，その情報を破棄する．今回はしきい値として人の歩行速度を参考に 5[m/s] とした．このフィルタリングを行った結果を図 9(b) に示す．外乱によるノイズが軽減されているのが分かる．

スニファでは，一定周期中に測定した距離情報の平均値をとることにより更なる誤差の軽減を行っている．平均値をとった後の測定距離を図 9(c) に示す．ここでは，1[s] ごとに平均値をとっている．一定期間中の平均値をとることは，測定した距離情報をサーバへ逐一通知することによる無駄な通信トラフィックの削減にも効果がある．実際に本ミドルウェアを運用する際には，平均値をとる周期はアプリケーションの要求に応じて，サーバでの位置計算周期とともに動的に変更することが望ましい．

測定距離のノイズを軽減したことにより，ミドルウェアが提供する位置情報上で，端末の位置関係が頻りに入れ替わるように見えたり，特定の端末が急に遠くへ移動してしまっただけのように見えることを軽減できる．

5. 関連研究

本章では，位置情報を用いて通信端末を同定する関連技術について紹介する．

センサネットワーク上のセンサまたはそれらが持つ情報を GoogleMap, GoogleEarth, Virtual Earth などのマップ上に配置し，センサネットワークのためのグラフィカルなユーザインタフェースを実現する研究が行われている [4], [5]．Geo-Coding [6] では，このようなユーザインタフェースを介してセンサネットワーク上の各センサとの双方向の情報交換を可能にし，ネットワークの運用コスト及び管理コストを削減している．しかしながら，事前に組織化されたネットワークが前提となっており（すなわち，各センサの位置を手動で設定する必要がある），加えてサーバ計算機が必要であることから，本研究とは異なる環境を対象としている．

端末の位置が既知でないような環境で，自律的に端末の位置を測定するシステムも存在する．位置情報を測定するシステムでは，端末に ID を割り当てた送信機を装備し，その送信機の位置を測定することで，ID と位置情報から端末を同定することができる．このようなシステムとしては，主に超音波を用いたもの，無線を用いたもの，及びそれらを複合して利用するもの [7] に分類できる．超音波を使ったシステムとして，Active Bat [8] や DOLPHIN [9] が挙げられる．これらのシステムは，特定の環境下において端末の位置を正確に測定することに主眼を置いている．特定の環境とは，あらかじめネットワークに接続された超音波受

信機を天井に複数配置した環境である。これらを利用して、IDをもつ超音波発信機を備えた端末の位置を測定する。測位精度は3~20cmと高精度であるが、専用の設備の導入にコストがかかる。

一方無線を用いたシステムとして、RADAR [10]とBradio [11]などが挙げられる。RADARは、無線LANの受信信号強度情報を用いた測位システムであり、端末の存在するエリア(部屋)を特定することを目的としている。複数の無線基地局及びデータベースを利用して測位対象となる無線端末の位置を推定する。このシステムでは、あらかじめ測位範囲内の各エリアで基地局の受信信号強度を測定しておき、データベースを作成しておく。実際に無線端末の位置を測定する際には、対象端末と基地局間の受信信号強度を観測し、データベースの情報と照らし合わせて、端末の存在するエリアを推定する。あらかじめデータベースを作成する必要があることから、その場で構成されるネットワーク上での利用には適さない。John KrummらのNearMe [12]は、近隣端末のリストとその位置をアプリケーションへ提供するという本研究と同じ目的をもっている。ただし、NearMeもRADARやBradioと同じく複数の無線基地局及びデータベースを利用して位置推定を行うアプローチをとっており、本研究と想定環境が異なる。

また、アドホックネットワークにおいて端末位置を測位する研究がなされている。文献[13],[14]では、SOM (Self-Organizing Maps) と呼ばれる手法をアドホックネットワークに適用し、端末が協調して位置を求める手法を提案している。固定の基準点を仮定している点、及びマルチホップを考慮している点で、本研究とは異なる環境を対象としている。また、ANMP [15]は、アドホックネットワークの管理プロトコルとして、IPネットワーク上で幅広く利用されているSNMPに着目し、アドホックネットワーク向けに改良している。ANMPでは、管理情報の中に位置情報を含むことが考慮されており、それらの情報を用いて、ネットワーク上のノードを管理することが考えられている。しかしながら、位置情報の取得方法に関して特に言及されておらず、ノードの相対関係を利用してより自然なインタフェースをユーザに提供するという本研究の目的とは異なる。佐藤ら [16]は、アドホックネットワークにおいて、端末の相互の接続関係から端末位置を統計的に推定する手法を提案している。この手法では、各端末が自端末から数ホップ以内の端末の位置をそれぞ

れ計算する。位置計算を各端末が行うことで、位置計算を担う端末の発見や選出を行う必要がないという利点がある。それに対して我々は、WVAC環境ですべての端末がシングルホップ接続していることを想定し、位置計算を担う端末を1台選出するアプローチを採用している。

商用として利用可能な位置推定ミドルウェアとしては、Ekahau [2]のEkahau Positioning EngineやAirLocation [3]のステーションマネージャが挙げられる。Ekahauは、あらかじめ各位置で測定した端末と基準点間の受信信号強度データベースをもとに、現在の端末位置を測定する手法をとっている。AirLocationは、端末からの電波を複数の基準点が受信し、それぞれの基準点への電波到達時間差 (Time Difference of Arrival, TDoA) によって距離を求め、三辺測量によって位置を推定する手法をとっている。

6. む す び

近距離無線通信デバイスを用いて、ユーザの視界内に存在する端末と、一時的にネットワークを形成し、即座に情報交換を実現する無線可視領域通信 (WVAC: Wireless Visible Area Communication) では、通信相手特定する技術が、利便性を左右する重要な要素技術である。提案ミドルウェアは、従来の名前を用いることに加えて、周辺端末との相対位置を補助的に提示し、WVAC環境における通信相手の特定を効果的に支援する。本ミドルウェアによりアプリケーションは周囲の端末の相対位置情報が参照できるほか、それらの端末との通信が即座に可能となる。これらを可能にするミドルウェアの設計を示した。提案ミドルウェアが既存の携帯端末の処理能力程度で十分に機能することを確認した。また、応用アプリケーションの例により本ミドルウェアの有用性を示した。

今後の課題として、位置計算を担う端末の適切な選出方法が挙げられる。計算資源や消費電力に余裕のある端末が位置計算を担う仕組みが求められる。また、多くの端末をサポートするための位置計算の分散化も挙げられる。その他、セキュリティの問題が挙げられる。本ミドルウェアでは不特定多数の端末が即座に通信ができ、かつ位置情報を互いに参照できる。このため、プライバシー保護やセキュリティについて考慮する必要があると考える。また、向きの問題も課題として挙げられる。本システムが測定するのは端末の相対位置であり、向きの情報はユーザの入力にゆだねてい

る．向きの情報が利用可能になれば，アプリケーションの可能性がより向上する．

謝辞 本研究の一部は，独立行政法人情報処理推進機構（IPA）未踏ソフトウェア創成事業（未踏ユース），科研費（基盤研究（B）18300021，若手研究（B）20700067）による助成を受けている．

文 献

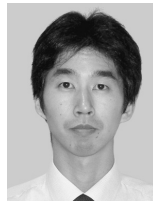
- [1] 北須賀輝明, 中西恒夫, 福田 晃, “無線通信網を用いた屋内向け測位方式,” 情処学論: コンピューティングシステム, vol.44, no.SIG10(ACS2), pp.131-140, 2003.
- [2] Ekahau Positioning Engine:
<http://www.ekahau.com/>
- [3] Hitachi AirLocation™:
<http://www.hitachi.co.jp/wirelessinfo/airlocation/>
- [4] G. Werner-Allen, P. Swieskowski, and M. Welsh, “MoteLab: A wireless sensor network testbed,” Proc. Fourth International Conference on Information Processing in Sensor Networks, pp.483-488, 2005.
- [5] S. Andre, N. Suman, L. Jie, P. Bodhi, and Z. Feng, “SenseWeb: Browsing the physical world in real time,” Demo Abstract, Proc. Fifth International Conference on Information Processing in Sensor Networks, 2006.
- [6] C. Decker, T. Riedel, P. Scholl, A. Krohn, and M. Beigl, “Graphically geo-coding of sensor system information,” Proc. Fourth International Conference on Networked Sensing Systems (INSS'07), pp.138-141, 2007.
- [7] P. Bellavista, A. Corradi, and C. Giannelli, “Coupling transparency and visibility: A translucent middleware approach for positioning system integration and management (PoSIM),” Proc. 3rd International Symposium on Wireless Communication Systems (ISWCS'06), pp.179-184, 2006.
- [8] A. Harter, A. Hopper, P. Steggle, A. Ward, and P. Webster, “The anatomy of a context-aware application,” Computer, vol.34, no.8, pp.50-56, 2001.
- [9] M. Minami, Y. Fukuju, K. Hirasawa, S. Yokoyama, M. Mizumachi, H. Morikawa, and T. Aoyama, “DOLPHIN: A practical approach for implementing a fully distributed indoor ultrasonic positioning system,” Proc. International Conference on Ubiquitous Computing (UbiComp) 2004, LNCS 3205, pp.347-365, 2004.
- [10] P. Bahl and V.N. Padmanabhan, “RADAR: An in-building RF-based user location and tracking system,” Proc. Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM) 2000, vol.2, pp.775-784, 2000.
- [11] B. Yoshimi, G.B. Bolam, N. Sukaviriya, J. Elliott, B. Carmeli, J. Morgan, and H. Derby, “Bradiao: A wireless infrastructure for pervasive computing environments,” Proc. 21st IEEE International Conference on Performance, Computing, and Communications, pp.309-316, 2002.
- [12] J. Krumm and K. Hinckley, “The NearMe wireless proximity server,” Proc. International Conference on Ubiquitous Computing (UbiComp) 2004, pp.283-300, 2004.
- [13] 朝倉 茂, 梅原大祐, 川合 誠, “SOM アルゴリズムを用いた移動端末の分散型位置推定法,” 情処学論 (B), vol.J85-B, no.7, pp.1042-1050, July 2002.
- [14] 滝沢泰久, デイビス ビーター, 岩井誠人, 川合 誠, 小花貞夫, “無線アドホックネットワークによる自律的端末位置推定方式とその特性,” 情処学論, vol.46, no.12, pp.2903-2914, 2005.
- [15] W. Chen, N. Jain, and S. Singh, “ANMP: Ad hoc network management protocol,” IEEE J. Sel. Areas Commun., vol.17, no.8, pp.1506-1531, 1999.
- [16] 佐藤雅幸, 松尾啓志, “統計的近似とばねモデルを用いたアドホックネットワークにおける端末位置決定手法,” 情処学論, vol.46, no.12, pp.2892-2902, 2005.
(平成 20 年 7 月 23 日受付, 11 月 7 日再受付)

野田 厚志



2004 九大・工・電気情報卒．2006 同大学院システム情報科学府博士前期課程了．現在，同大学院システム情報科学府博士後期課程在学中．主として組込みシステム，モバイルコンピューティングに関する研究に従事．

北須賀輝明 (正員)



1993 京大・工・情報工学卒．1995 奈良先端科学技術大学院大学情報科学研究科博士前期課程了．同年シャープ(株)入社．パーソナルコンピュータの開発に従事．2001 九州大学大学院システム情報科学研究院助手．2006 九州大学博士(工学)．2007 年 3 月九州大学大学院システム情報科学研究院助教．2007 年 10 月熊本大学大学院自然科学研究科准教授．モバイルコンピューティング，組込みシステム，並列/分散処理，コンパイラ，計算機アーキテクチャに研究的関心をもつ．情報処理学会会員．



田頭 茂明 (正員)

1996 龍谷大・理工卒。1998 奈良先端科学技術大学院大学情報科学研究科博士前期課程了。2000 同大情報科学研究科博士後期課程了。2000 広島大学工学部第二類助手。2001 広島大学大学院工学研究科助手。2007 広島大学大学院工学研究科情報工学専攻助教。2007 九州大学高等研究機構若手研究者養成部門 SSP 学術研究員，九州大学大学院システム情報科学研究院特任准教授。システムソフトウェアの研究に従事。IEEE Computer Society，情報処理学会各会員。



中西 恒夫 (正員)

1993 阪大・工・通信卒。1998 奈良先端科学技術大学院大学情報科学研究科博士後期課程了。博士(工学)。1996 日本学術振興会特別研究員。1998 奈良先端科学技術大学院大学情報科学研究科助手。2002 九州大学大学院システム情報科学研究院助教授，また同大学システム LSI 研究センターほかを併任。2007 年 3 月九州大学大学院システム情報科学研究院准教授。コンパイラ，計算機アーキテクチャ，組込みシステム，並列/分散処理に研究的関心をもつ。ACM，IEEE Computer Society，情報処理学会各会員。



福田 晃 (正員)

1977 九大・工・情報工学卒。1979 同大大学院修士課程了。同年 NTT 研究所入所。1983 九州大学大学院総合理工学研究科助手。1989 同大学助教授。1994 奈良先端科学技術大学院大学情報科学研究科教授。2001 より九州大学大学院システム情報科学研究院教授。工博。2008 九州大学システム LSI 研究センターセンター長，教授(兼任)。オペレーティングシステム，コンパイラ，組込みシステム，モバイルコンピューティング，計算機アーキテクチャ，並列/分散処理，性能評価等の研究に従事。本会平成 2 年度研究賞，平成 5 年度 Best Author 賞受賞。著書『並列オペレーティングシステム』(コロナ社)，訳書『オペレーティングシステム』(共訳，培風館)。ACM，IEEE Computer Society，情報処理学会，日本 OR 学会各会員。