

モバイルアドホックネットワークにおける コミュニケーション支援手法

河 口 信 夫[†] 清 水 邦 彦[†]
外 山 勝 彦^{†,††} 稲 垣 康 善[†]

本稿では、携帯端末間において構築されるアドホックネットワークにおけるコミュニケーションの特徴を示し、その上で情報の共有や交換を支援する手法を提案する。本手法では、端末を持ち運び、アドホックネットワーク上での情報の同期を繰り返すことにより、単一のネットワークに接続されない複数のホスト間で情報の同期が可能になる。また、本手法を用いて実現したコミュニケーション支援のプロトタイプシステムを示す。

Communication Supporting Methods for Mobile Ad-Hoc Networks

NOBUO KAWAGUCHI,[†] KUNIHICO SHIMIZU,[†] KATSUHIKO TOYAMA^{†,††}
and YASUYOSHI INAGAKI[†]

This paper clarifies the characteristics of ad-hoc communication between the portable information devices under the ad-hoc network. This paper proposes supporting methods for information sharing over the ad-hoc network. By using these methods, one can synchronize the information between the hosts which aren't connected to the same network simultaneously. This paper also shows the prototype systems for communication support based on the method.

1. はじめに

急速な携帯情報端末の小型・低価格化と共に、情報端末を常に持ち運び、日常的に利用する人々が増加しつつある。携帯端末普及の要因としては、携帯電話やPHSの利用により、いつでも必要な時に必要な情報を利用できることが挙げられる。一方、多くの人が情報端末を持ち運ぶことによって、移動先で情報端末を携帯している他人に出会う機会も増えつつある。しかしながら、会議等において各参加者が利用している情報端末間での情報の交換・共有はほとんど行われていない。また、移動先には、自宅やオフィスのデスクトップPCや、コンビニや駅などに設置された情報キオスク端末が存在するが、携帯している端末との情報交換・共有を行う仕組みは十分ではない。

我々は、このような移動先で一時的に行われる情報交換・共有を特にアドホックコミュニケーションと呼び、情報端末による支援の実現を目指して研究を進めている²⁾。アドホックコミュニケーションを支援するためには、各情報端末間で端末間通信が必要となるが、これを実現するのが必要に応じて一時的に構築されるアドホックネットワーク⁴⁾である。

本稿では、アドホックネットワークにおけるコミュニケーションの特徴を明らかにし、携帯端末間や固定端末間での情報共有を支援する手法を提案する。また、本手法に基づいて構築したオークションシステムとマルチプレーヤ通信ゲームについて述べる。

2. アドホックネットワークにおけるコミュニケーションの特徴

アドホックネットワークは必要に応じ、一時的に作られるネットワークであり、以下の特徴を持つ。

- ネットワークを管理する端末が存在しない(常に存在するとは限らない)
- 動的な端末の出入りがある

[†] 名古屋大学大学院工学研究科
Graduate School of Engineering, Nagoya University

^{††} 名古屋大学統合音響情報研究拠点
Center for Integrated Acoustic Information Research(CIAIR), Nagoya University

表 1 各ホストの保持する同期情報

ID		自ホストの ID
同期履歴リスト		過去に同期したホスト ID n_i と最終同期時刻 t_i (n_i, t_i) のリスト
ユニット変更時刻		すべてのユニットの最終更新時刻
各ユニット毎 の同期情報	ID	ユニット ID
	変更時刻	ユニットの最終変更時刻
	シーケンス番号 リスト	ユニットの新しいさを示す値。過去にそのユニットを扱った端末の ID n_j と、その端末がユニットを変更する毎に増やす値 s_j (n_j, s_j) のリスト

- ・ 事前に設定がない
- ・ 各端末が自律的に構築する
- ・ 任意の端末間で直接通信できるとは限らない

我々は、すでに動的にアドホックネットワークを構築する手法を提案しており¹⁾³⁾⁴⁾、またその上で動作する様々なアプリケーションをモバイルエージェント技術を用いて検討を進めてきた²⁾。

しかし、実際にアドホックネットワークが使われる場面を考えてみると、ネットワーク上で共有・交換された情報は、その場での利用に加え、他の異なるアドホックネットワークに参加して情報交換を行うことによって有用となる場合が多く存在する。

例えば、個人の携帯端末を自宅の PC やオフィスの PC との間で持ち歩く場合を考える。携帯端末には個人のスケジュールが入っており、自宅の PC には家族の予定、オフィスの PC には仕事関係のスケジュールが入っているとすると。この場合、各端末のスケジュール情報は独立にアップデートされるが、端末を PC に接続すれば、スケジュールの同期が行えるとする。自宅やオフィスで行われる各々のスケジュール同期は、一種のアドホックコミュニケーションと考えることができる。なぜなら、この通信は常時可能なわけではなく、必要に応じて行われるためである。しかし、1 回のアドホックコミュニケーションでは、自宅やオフィスのどちらかの PC としかスケジュールを共有することができないため、自宅とオフィス間を移動して、複数回の同期することによって、はじめて正しいスケジュール情報を得ることができる。つまり、単なるアドホックコミュニケーションではなく、その繰返しによって必要な情報を取得するのである。

このような場合、各々のアドホックコミュニケーションは、単なる端末間の情報同期ではなく、直接接続していない自宅の PC とオフィスの PC 間での情報同期を実現していると考えることができる。つまり、独立したネットワーク間でも、各ネットワーク上で移動端末がアドホックコミュニケーションを行い、ネットワーク間を移動すれば、ネットワークを超えて情報の共有・交換が行えることを示している。

本稿では、アドホックコミュニケーションにおいて本質的に重要な技術は、上記に示したように、同一ネットワーク上に存在しない複数端末間で、情報の整合性を保ちつつ情報の同期・共有を行う手法であると考え、その手法を提案する。

3. アドホックネットワークにおける情報の同期手法

以下では、同期を行う情報端末をすべてホストと呼び、同期する情報の単位をユニットと呼ぶ。ユニットは、単なるファイルであっても、データベース中の各々の値でも良い。本稿では、各ホストが固有の ID を持ち、独立に動作する時計を保持していることを前提とする。また、アドホックネットワークにおいては、通信可能なホストの ID を得られるものとする。

本手法では、ホスト間で同期を行う場合に以下の点を考慮している。

- ・ 最新のユニットを優先する
複数のホストが、同一のユニット ID であるが、内容の異なるユニットを保持している場合、新しいユニットを優先して残す。
- ・ 同期に必要な通信量を削減する
できる限り通信量を減らす。これにより通信時間の削減も可能になる。

以下では、これらの点を実現するための手法、および同期情報について述べ、本手法における同期の手順を示す。

3.1 ユニットの最新性の判断

一般に、ファイル名が同じだが、内容の異なるファイルにおいて、最新のファイルの判断するためにはタイムスタンプを用いる。しかし、アドホックネットワークの環境では、各ホストの時計が同期しているとは限らないため、各ホスト上のタイムスタンプを比較材料としては利用できない。そこで本手法では、各ユニット毎に付加情報として、ユニットを変更したホスト ID と、そのホストによる変更回数を表すシーケンス番号を対にしてリストとして保持し、さらにユニットの最終変更時間を保持することとした。

これにより、自分や通信相手のホストが変更していないユニットでもそのユニットが最新であるかの判断が可能になった。

3.2 通信量の削減

アドホックネットワークで、情報の同期を行う場合、新しいホストを発見する毎に、そのホストとの同期の要求が生じる。新しいホストが見つかる毎に全てのユニットを送ることは、通信量が多く無駄であるため、同期の必要性を確認し、最小の通信量とする必要がある。本手法では、各ホストは過去に同期したことがあるホストと最終同期時刻の対のリストである同期歴リストを保持しており、新たなホストの最終同期時刻が、最後にユニットを変更した時刻(ユニット変更時刻)よりも前である場合に同期が必要であることがわかる。

この場合、時刻はホスト上のローカルタイムであるので、比較することが可能である。さらに、各ユニット毎にホストとの最終同期時刻と変更時刻を比較することによって、そのユニットを送るべきかが判断できる。

各ホストが保持する同期情報をまとめて表1に示す。

3.3 同期手順

各ホストは、以下の手順を繰り返すことによって、アドホックネットワーク上の情報同期を実現する。

- (1) 出会ったホスト間でアドホックネットワークを構築し、ネットワークから現在通信可能なホストIDを受け取る。
- (2) 自分の同期歴リストを用い、通信可能でユニットの変更時刻後に同期していないホストがあった場合、同期端末リストに記録する。
- (3) 同期端末リストに記録してある各ホストに対して、ホストIDと変更したユニットを送信する。また、その他の通信可能なホストに対しては、既同期メッセージを送信する。
- (4) 変更されたユニットを受け取ったホストは、付加された各ホストに対するシーケンス番号が1つでも大きいユニットを優先し、自分の保持するユニットを更新する。全てのシーケンス番号が同じ場合は、ユニットの更新を行う必要はない。シーケンス番号リストの大小が比較できない場合は各ユニット毎に特別な同期処理を行う[☆]。
- (5) 通信可能な全てのホストから、変更されたユニット、もしくは、既同期メッセージを受信すれば、同期終了。同期歴リストに、通信した全てのホストIDと同期時刻を記録する。

[☆] 複数のホストによって独立に変更されたユニットの同期は、単純なルールで行うことはできない。詳細については5章にて議論する。

上記動作中に新しいホストが参加した場合は、そのホストに対して(1)からの動作を繰り返す。

4. 情報同期手法に基づくプロトタイプシステム

本手法に基づき、Java(JDK1.1以降)上で情報同期ライブラリを実現した。実現には、アドホックネットワークを構築するモバイルエージェントシステムMAGNET⁷⁾を利用した。アドホックネットワークを構築する機構がすでにMAGNET上に存在するため、ライブラリを簡潔に実現することができた。本ライブラリは、アドホックネットワーク上の様々なコミュニケーション支援システムを実現するのに有用である。

ライブラリの有用性を示すために、ライブラリを用いて、アドホックコミュニケーションを支援するアプリケーションとして、オークションシステムや、マルチプレーヤ通信ゲームを実現した。以下では、同期ライブラリと、実現した各システムについて述べる。

4.1 同期ライブラリ Sync

同期ライブラリは、同期を実現するオブジェクト、各ユニットを管理するオブジェクト、基本的なユニットを表すクラス、および各オブジェクトを結び付ける統合オブジェクトから構成される。同期オブジェクトには、ユニットを格納するための `void setUnit(Vector vec)` や、同期されたユニットを取り出すための `Unit getUnit(UnitID uid)` などのメソッドが存在する。通常のオブジェクトの代入に代えて、これらのメソッドを利用すればネットワーク上の通信や情報の同期のことを考慮せずに情報共有のプログラミングが可能である。

また、他のホストでユニットが更新された場合に呼び出されるメソッド `void updateNotify(Unit un)` によって、同期が行われた場合の処理も容易に記述できる。

基本的なクラスが実現されているので、プログラマはユニットクラスを実現するだけで、本ライブラリを利用することができる。

4.2 オークション

同期ライブラリを用い、シンプルなプロトタイプ情報同期システムとしてオークションシステムを実現した。図1に、動作中の画面を示す。左上の接続状況ウィンドウは、MAGNETのアドホックネットワークエミュレータである。この上で各ホスト間のリンクを繋げたり切ったりすることによって、アドホックネットワークの構築をエミュレートできる。

このシステムでは、ユーザがいつでもオークションを

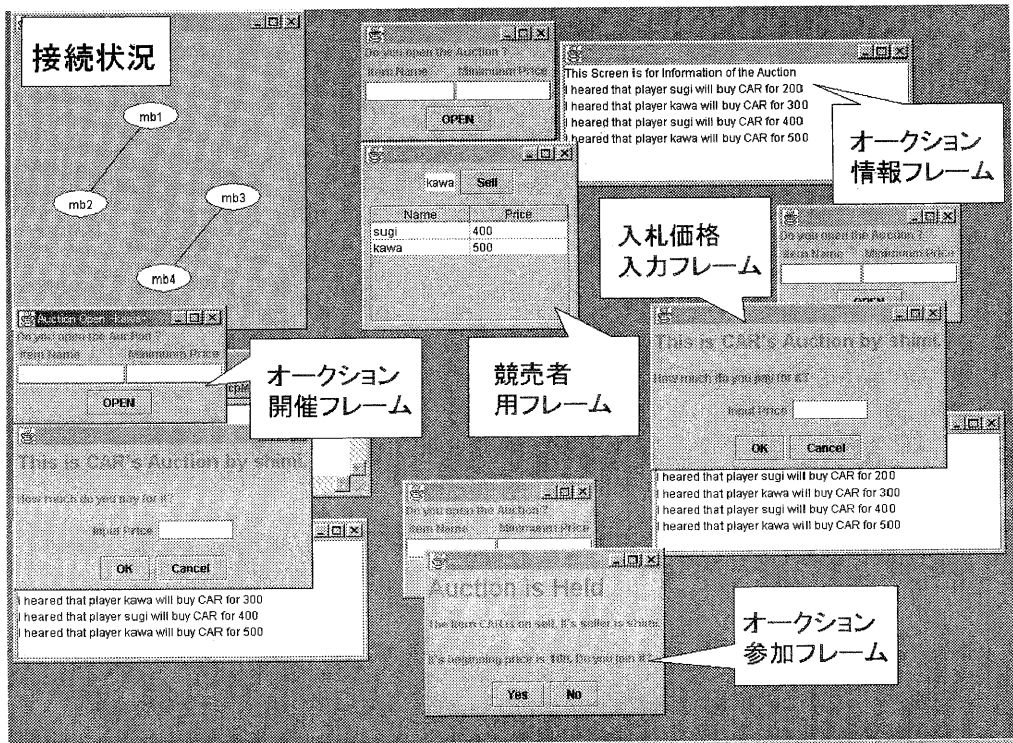


図1 オークション システム

開くことができる。図1左のオークション開催フレームでは、売り出す物と最低値段をつけてオークションに出品することができる。あるユーザがオークションを開いた情報は、アドホックネットワークを通して、ネットワーク上の他のホストに伝えられる。

一般のネットオークションと本システムでは、動作が大きく異なる点がある。ネットオークションでは、サーバがすべての情報の管理をしており、オークションへのビッドの結果も即座に反映される。一方、このシステムでは、オークション参加者は、現在のオークション参加者にアドホックネットワーク上で出会って情報共有を行なった人であり、必ずしもオークション開催者のホストと直接接続している必要はない。つまり、どこでもオークションを開催することが可能であり、さらにサーバを必要としない。各参加者は、オークションにいつでもビッドすることができる。

また、オークション参加者は、他の参加者とアドホックネットワークを構築することによって、その時点での入札相場を知ることができ、また、オークション開催者と出会えば落札の状況を確認することができる。

本システムはユーザとオークション価格をユニットと

して実現している。同期ライブラリの利用により、短時間で簡潔なシステムの実現が可能となった。

4.3 マルチプレーヤ通信ゲーム

本手法により複雑なコミュニケーションの支援が可能であるかを確認するために、同期ライブラリを用いてマルチプレーヤ通信ゲームを構築した。このゲームは、モノポリーに似たゲームであり、各ユーザは、共有された盤面上で、盤面上のコマを動かし、土地を購入することができる。動作中の画面を図2に示す。この画面では、3名のプレーヤが土地の争奪戦を繰り広げている。

一般のマルチプレーヤ通信ゲームでは、途中退出や途中参加は難しいが、このゲームでは、常に参加・退出が自由である。また、途中退出したプレーヤが他のホストとゲームの続きを進めることや、別々に開始された複数のゲームを途中で合流させることも可能である。

これらの情報同期の仕組みはすべて同期ライブラリを用いて実現している。さらに、土地については、前節のオークションシステムを用いて競売にかけることができる。このゲームでは、コマの位置、土地、得点等について同期ライブラリを利用しており、本ライブラリにより、簡潔な実現が可能になった。

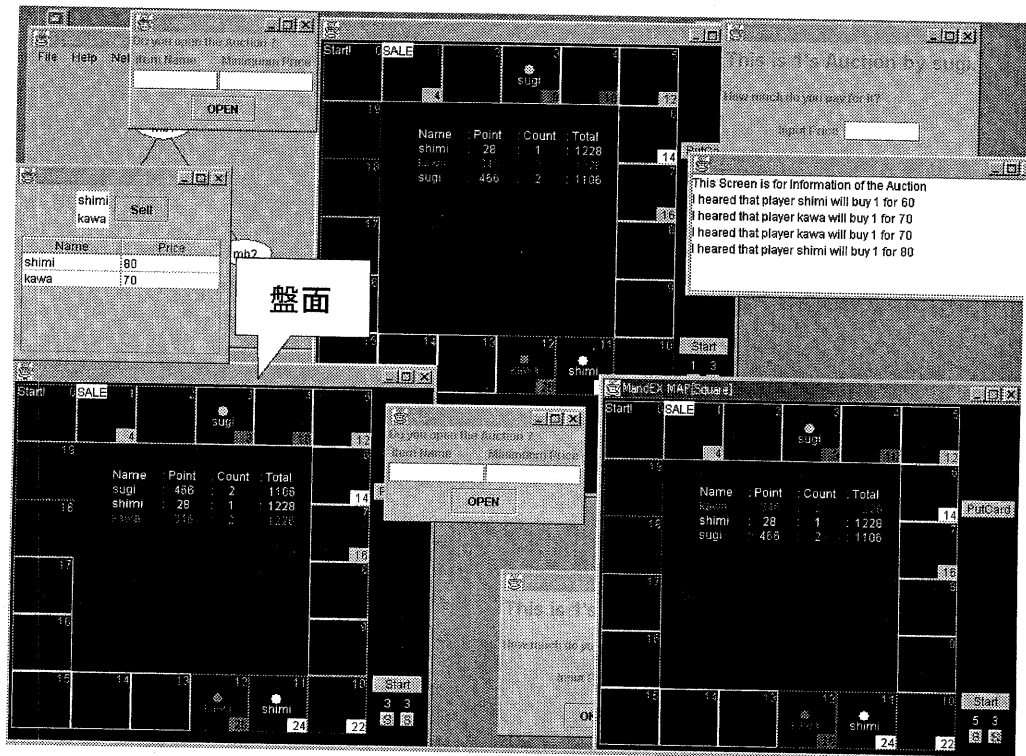


図2 マルチプレーヤ土地売買ゲーム

5. 議 論

5.1 ユニットの同期が困難な場合

ファイルシステムや、データベースでは、データの整合性が重要であり、その維持のための多くのコストを費やしている。本手法においても、できる限り最新のユニットを優先させることによって、整合性を保とうとしている。

しかし、複数のホストで独立に変更されたユニットにおいては、シーケンス番号リスト間で優先順位をつけることができない。この場合、ユニットの中身を見て、どのように同期を行うかの判断を行う必要がある。しかし、ユニットの中身はアプリケーションによって異なるため、一般的なルールを決めることはできず、各アプリケーション毎に適切なルールを定める必要がある。

また、できる限りこのような問題を減らすためには、ユニットの単位を小さくすることが考えられる。例えば、テキストファイルを共有データとする場合、ファイル1つをユニットとせず、各行をユニットとすれば、優先順位の衝突も少なくなることが期待できる。

我々が実現したゲームにおいても、盤面全体をユニッ

トとせず、各コマや、購入した土地毎にユニットを生成して、できる限り優先順位の衝突が起こらないようにしている。また、このゲームでは、優先順位の衝突が起こった場合においても、例えば、あるプレーヤが土地を売った後に、他の土地が売られていない盤面と同期した場合には、所有者のデータを優先するルールを導入している。

5.2 セキュリティ

情報共有においては、データの整合性を保つことに加え、安全性や機密性を保つことも重要である。本稿では、情報の配布の範囲(アクセス権)やセキュリティについては述べていないが、実用上は避けては通れない問題である。

アドホックネットワークにおける認証は、認証機関がネットワーク上に存在しない場合もあり得るため、完全に行えるわけではない。そこで、重要な情報は、とりあえず暗号化して共有し、認証機関を通じた後に、鍵を得ることができるような仕組みが必要である。また、データや同期情報の改竄に対し頑健な仕組みを検討する必要がある。

6. 関連研究

6.1 分散情報共有システム

固定ネットワークにおいて、分散情報共有を行う仕組みは、分散データベースや、CORBA や HORB 等の分散オブジェクトなどにより活用されている。しかし、これらのシステムは、サーバを中心として動作しており、サーバが存在しないアドホックネットワーク上では動作しない。また、本手法のように接続の繰り返しを用いて情報共有を行う手法はこれまで提案されていない。

6.2 分散ファイルシステム

一方、接続されていない状況で情報を共有しつづける仕組みとしては、Coda⁵⁾ や PFS⁶⁾ 等の断絶時モードを持つファイルシステムが実現されている。しかし、これらのシステムは共にサーバ・クライアント型であり、本手法が可能にしている任意のホスト間での情報共有の実現はなされていない。すなわち、これらのファイルシステムでは、上記のマルチプレーヤ通信ゲームやオークションシステムの実現は困難である。

また、情報を同期する仕組みとしては、Windows95/98 にに附属するブリーフケースが存在するが、これはピアツーピア型であり、2 台のホスト間での情報共有が可能になるだけで、複数のホスト間での共有は行えない。

しかし、これらのシステムは、ファイルを用いることによって、既存のアプリケーションの応用が可能である点が利点である。

7. ま と め

アドホックコミュニケーションにおいて、同じネットワークに接続していない端末間で、情報の同期・共有を支援する手法を提案した。また、本手法を実現する同期ライブラリを Java 上で実装した。本ライブラリは、アドホックネットワークを実現する MAGNET 上で動作する。ライブラリを用いて、オークションシステムや、マルチプレーヤ通信ゲームを実現し、本手法とライブラリの有用性を確認した。

今後の課題としては、本ライブラリを用いた同期パフォーマンスの測定と、スケーラビリティの調査が挙げられる。また、アドホックネットワークにおけるコミュニケーション支援には、周囲の状況を判断し、適応して動作するアプリケーションが不可欠である。本手法と、そのような適応システムが協調して動作することによって、より柔軟で高度なコミュニケーション支援システムの構築が期待される。

参 考 文 献

- 1) 河口信夫, 片桐秀樹, 内柴道浩, 外山勝彦, 稲垣康善: モバイル環境下の自律分散通信の実現とその応用, マルチメディア, 分散, 協調とモバイルワークショップ論文集 (DICOMO'98), pp.619-626 (1999).
- 2) 河口信夫, 外山勝彦, 稲垣康善: 携帯情報端末間の情報共有を支援するモバイルエージェントシステム, 情報処理学会第 59 回全国大会特別セッション (1) 講演論文集, pp.23-30(1999).
- 3) 片桐秀樹, 河口信夫, 外山勝彦, 稲垣康善: モバイル環境下の自律分散通信の実現とその応用, 情報処理学会研究報告, 99-MBL-7 pp.63-70 (1998).
- 4) 河口信夫, 外山勝彦, 稲垣康善: モバイルエージェントによるアドホックネットワークの構築, ソフトウェア科学会 SPA'99, (1999).
- 5) Kister, J.J. and Satyanarayanan, M.: Disconnected Operation in the Coda File System., Proc.13th Symposium on Operating Systems Principles, pp.233-238 (1991).
- 6) 楯岡孝道, 植原啓介, 砂原秀樹, 寺岡文男: PFS: 通信環境に動的に適応するファイルシステム, コンピュータソフトウェア, Vol.15, No.2, pp.62-81 (1998).
- 7) Nobuo Kawaguchi, Katsuhiko Toyama, Yasuyoshi Inagaki: MAGNET: Ad-Hoc Network System based on Mobile Agents, Mobile Agents for Telecommunication Applications (MATA'99), pp.127-142(1999).
- 8) 清水邦彦, 河口信夫, 外山勝彦, 稲垣康善: アドホックネットワークにおける情報分散共有手法, 電気関係学会東海支部連合大会講演論文集, pp.292 (1999).