

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

Ad Hoc Network Construction using Mobile Agents

河口 信夫 外山 勝彦 稲垣 康善

Nobuo KAWAGUCHI Katsuhiko TOYAMA Yasuyoshi INAGAKI

名古屋大学 大学院工学研究科 計算理工学専攻

Department of Computational Science and Engineering

Graduate School of Engineering, Nagoya University

概要

携帯端末が急速に普及し、モバイル環境下において端末間で直接情報を交換・共有することが求められている。このような場面で、必要に応じて構築されるネットワークは、アドホックネットワークと呼ばれ、さまざまな構築手法が提案されている。しかし、一般にネットワークは多様な状況で用いられ、単一のルーティングプロトコルが全ての状況において最適となることは考えられない。本稿では、様々な状況に応じたアドホックネットワークを構築するために、モバイルエージェント技術を用いる手法を提案する。通信を行う端末がその目的に応じて適切なエージェントを利用することにより、状況を認識し、適応した動作をするネットワークを構築することが可能となる。また、従来のルーティングプロトコルでは、中継する端末の接続が切れていると通信を行うことができなかったが、接続を待つエージェントを利用することにより、断続的なネットワーク接続においても通信が可能となる。

1 はじめに

小型で高性能な携帯情報端末を自由に持ち運び、高度な無線技術によりいつでもネットワーク接続を可能にする環境が整った。しかし、モバイル環境下において端末間で情報の交換や共有を直接行いたいという要求は十分には満たされていない。いつでもどこでも自由に端末間で情報の交換・共有が可能になれば、名刺交換やスケジュール調整、議事録やメモの共有、資料の配布等の、これまで人が行ってきた協同作業の支援が可能になることが期待される(図1)。このように、必要に応じ一時的に構築するネットワークはアドホックネットワークと呼ばれ、近年盛んに研究が行なわれている [10] [11] [12] [16]。アドホックネットワークでは、ネットワークを構成する端末は事前に互いの存在を知らず、集中管理を行うサーバは存在しない。そのため、従来のネットワークとは異なった概念や管理・利用手法が必要となる。

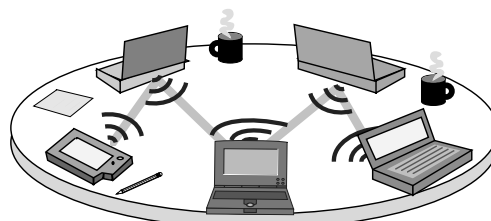


図1 携帯端末によるネットワーク

我々はすでに、出会ったその場で、いつでもどこでも誰とでも何台とでも手軽に通信を可能にするアドホックネットワークの構築手法を提案した [5] [7]。この手法では、各端末は自律的に周囲の端末を認識し、情報交換を行うことによりアドホックネットワークを構築する。直接通信が不可能な端末同士は、自律的に最適な通信路を選択し、中継により通信を行う。しかし、この手法は接続端末数の少ない小規模なネットワークを対象としており、大規模な通信には向かない。

一方、アドホックネットワークをTCP/IP上で構築するために、IETF [20]内でMANETというワーキンググループが研究を行っている。ここでは、AODV [12], DSR [16]といったさまざまなルーティングプロトコルが提案されており、性能の比較実験 [17]が行なわれている。

しかし、ネットワークに接続する端末の数、集中・分散の度合、データ量、通信速度、リンクの双方向性等が多様であるため、単一のルーティングプロトコルが全ての状況において最適となることは考えられない。すなわち、任意の状況で適切な通信を行なうためには、一つの端末上に様々なプロトコルを準備し、さらに、ネットワーク上のすべての端末がそのプロトコルを理解する必要があり、現実的には困難である。

本稿では、様々な状況に応じたアドホックネットワークを構築するために、モバイルエージェント技術を用いる手法を提案する。各データパケットをモバイルエージェントと捉え、全ての通信をエージェントの移動として考える。通信を行う端末がその目的に応じて適切なエージェントを利用することにより、状況を認識し、適応した動作をするネットワークを構築することが可能となる。また、従来のルーティングプロトコルでは、中継する端末の接続が切れていると通信を行うことができなかったが、接続を待つエージェントを利用することにより、人づてに伝言を伝えるように、断続的なネットワーク接続においても通信が可能となる。我々は、この枠組に基づいたプロトタイプシステムとしてモバイルエージェントネットワーク MAGNET を構築した。MAGNET は携帯端末等の赤外線通信上で用いることを主な目標としており、現在は赤外線エミュレータ上で動作している。

以下では、アドホックネットワークにおけるモバイルエージェントの利用について述べる。また、MAGNET において、複数の単純なエージェントの組合せにより、アドホックネットワークの代表的なルーティングプロトコルである DSR [16]が実現できることを示し、さらにその断続的なネットワークへの拡張を示す。

2 アドホックネットワークの構築

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。

- ネットワーク構築前に設定がない (他の端末の存在を知らない)。
- 各端末が自律的に構築する。
- ネットワークを集中管理する端末がない。
- 任意の端末間で直接通信できるとは限らない (隠れ端末が存在する)。
- 新たな端末の参加や退出など、ネットワークトポロジの動的な変化がある。

アドホックネットワークを構築するためには、一般に複数の端末上で動作する分散アルゴリズムの開発が必要となる。通信エラー等が頻発し、動的にネットワークトポロジが変化する場合には、各端末の状態の整合性を保つために、非常に複雑な処理を必要とする。我々が開発した赤外線通信プロトコル [7]では、端末間での通信量や回数を減らすために、通信相手の端末の状態を利用しており、動的な変化やエラーが状態の整合性を狂わせることが問題となった。そのため、頑健性を実現のためには、様々な状態に対応可能なイベントテーブルを作成する必要があった。

このような経験から、動的な環境やエラーが発生する状況では、端末間の通信を順序良く同期的に行う手法は複雑になるという知見を得た。一方、モバイルエージェントは、状態を持ったまま移動するため、プログラム上で状態を予測する必要がなく、特別な同期の仕組みを必要としないネットワーク構築が可能になることが期待される。

ところが、従来のモバイルエージェント技術は、ネットワークの存在や、通信相手の存在を仮定しており、ネットワークの動的な変更が起こるアドホックネットワーク上ではそのまま利用できない。つまり、ネットワークの状況の動的な変化に対応するための手法が必要となる。そこで、我々はネットワークのリンクを監視し、その変更をイベントとしてエージェントに伝えることにより、動的な適応を可能にする枠組を提案する。次章では、我々が開発したモバイルエージェントによるネットワークについて述べる。

3 モバイルエージェントネットワーク

MAGNET (Mobile AGent NETwork) は、モバイルエージェントにより実現される通信ネットワークシステムである。MAGNET は、将来的には赤外線通信 (IrDA [19]) 上での実装を目指している。現在は、Java(JDK1.1) 上におけるモバイルエージェントシステムとして実装されており、通信はソフトウェアで実現された赤外線エミュレータ上で行う。MAGNET では、シンプルなモバイルエージェントを組み合せ、様々なネットワークプロトコルと同様の動作を実現することができる。

MAGNET の特徴を以下に示す。

- 重複するエージェント ID の存在
エージェントは生成時には固有の ID を持つが、エージェントをデータパケットとして用いる場合のために、同じ ID を持ったまま他のホストへコピーする機能を持つ。すでに存在するエージェントと同じ ID を持つエージェントがホストへ移動してきた場合、存在するエージェントに後から来たエージェントの情報が渡され、情報の統合や同期が可能になる。
- ネットワークリンクの監視
アドホックネットワークを構築するためには、近傍のホストの存在を確認する。リンクの存在を適宜確認し、変更があった場合、適切なエージェントに伝達する。
- エージェント階層の概念
エージェントに階層の概念を導入し、単純なエージェントの組合せで複雑なタスクの実現を目指す。エージェントの階層はシステムとしては管理せず、各々のエージェントがその登録を管理する必要がある。

従来のエージェントシステム [13] では、エージェントのコピーにおいては、必ず新しい ID を持つ方法論が用いられてきた。一方、ネットワーク上のデータパケットは一般に同じパケットは同じ ID を持つ。そこで、MAGNET では、同じ ID を持ったまま他のホストへコピーする機能を導入した。ただし、単一のホスト上には同じ ID を持つエージェントは複数存在しない。これにより、ループが存在するネットワークにおいても、エージェントがループをおこさない仕組みを容易に提供できる。これは DSDV [10] において、パケットに固有のシーケンス番号をつける手法と同様の方法である。

リンクの監視により、エージェントの適応動作が可能になる。例えば、携帯端末を利用しているユーザの場合、リンクの新たな存在は、ユーザが何らかの通信を行おうとする意図を表していると考えられる。そこで、適切なエージェントにリンクの存在を伝えることにより、ユーザの意図を理解した動作が可能になる。また、美術館や会議室といった特定の場所に存在するホストにおいては、新たなリンクの存在は新たなユーザの登場であるため、必要なエージェントを移動させることができる。

また、MAGNET では、MobileSpaces [8] で用いられたエージェント階層の概念を導入している。単純な働きをするエージェントを適切に組み合わせることにより、必要なネットワークエージェントが構築できる。エージェント階層はネットワークエージェントのコンポーネントプログラミングを可能にする。

MAGNET の各端末は以下のステーションナリエージェント (マネージャ) を基本構成として持つ。

1. エージェントマネージャ:
ローカルノードに登録されるすべてのマネージャ、エージェントの管理を行う。また、エージェントと他のマネージャとの通信を担当する。
2. リンクマネージャ:
近傍のノードの探索を行い、通信を希望するエージェントに通知する。エージェントの配送もここで行う。リンクは赤外線や TCP/IP 等を利用できる。
3. ルーティングマネージャ:
ネットワーク上のノードを辿って通信を行うルーティングエージェントを管理する。ノード間の通信は、ルーティングマネージャが、各ルーティングエージェントに依頼して行う。

MAGNET では全てのプログラムコードをモバイルエージェントとして捉える。一般に、マネージャは動的に変更されないが、システムのバージョンアップの際には他のエージェントと同様に配送される。

3.1 MAGNET エージェント

図 2 に MAGNET におけるモバイルエージェントの基本クラスを示す。他のエージェントシステムに比べイベントハンドラが少ないが、これは、システムの目的をネットワークの構築に絞ったためであ

```

public class MagAgent implements Serializable{
    MagID id = MagID.getNewID();
    transient MagManager magMgr;
    void onRegister(){
        // エージェントマネージャに登録後
        // 呼び出される
    }
    void onDispose(){
        // エージェントマネージャから消された後
        // 呼び出される
    }
    void onSame(MagAgent m){
        // 同じ ID のエージェントが同じ
        // ホスト上に来た時に呼び出される
    }
}
// リンクモニタインタフェース
interface LinkMonitor {
    // 新しいリンクが現れた時呼び出される
    public void onNew(Vector ns);
    // リンクが消滅した時呼び出される
    public void onBye(Vector ns);
}

```

図2 MAGNET エージェントとリンクモニタ

り、永続化等の機能は現在のところ持たない。

onRegister はエージェントがシステムに登録された時に呼び出され、最初の場所と移動先での区別はない。ただし、コンストラクタ等で変数を初期化することが可能である。

onSame は同じ ID を持つエージェントがシステムに登録された場合に、そのエージェントを引数として呼び出される。これにより、ネットワーク上でエージェントがループ状に動作することを防ぐとともに、複数のエージェントの知識を共有することが可能である。後に示す BroadCaster はこの機能を用いている。

また、LinkMonitor はリンクマネージャが管理するエージェントが持つべきインタフェースを定義しており、新たなホストの出現により onNew が呼び出される。

3.2 プロトコル構築例 (DSR)

MAGNET により、ネットワークプロトコルが構築できることを示すために、アドホックネットワーク上で有効なルーティングプロトコルである DSR(Dynamic Source Routing protocol) [16]の実現を示す。DSR は、主に Route Discovery と Route Maintenance の 2 つの手法から構成される。Route Discovery は、ホスト S が宛先ホスト D と

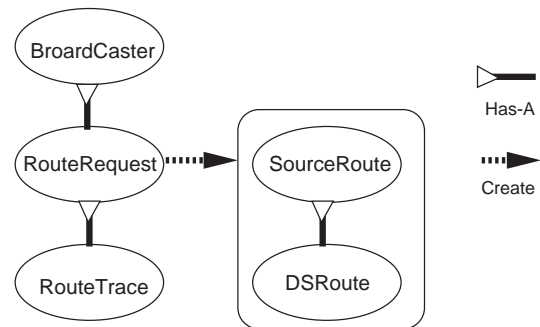


図3 DSR プロトコルのデザイン

通信を行う場合に、D までの経路を必要とする時に行われる。まず、ホスト S は RREQ(Route Request) パケットを隣接している全ホストに送信する。RREQ パケットがホスト D に到達するか、D までの経路を知っているホストに到達した場合、そのホストは RREP(Route Reply) パケットを S に対して返送する。このパケットには D までの経路情報が保存されている。D までの経路を知らないホストは、RREQ パケットを中継していく。これにより、ホスト S は D までの経路を知ることができ、通信が可能になる。また、ルーティングパケットによりネットワークが混雑するのを防ぐために、各ホストはできるかぎり知り得たルーティング情報をキャッシングしている。Route Maintenance は経路情報が間違っている場合に、それを訂正する手法であり、Route Error パケットにより通知される。経路が使えなくなった場合、他の経路を利用するか、Route Discovery を再度行えば良い。以上の手続きにより、DSR はアドホックな状況での通信を可能にしている。

本稿の実現では Route Discovery のみを対象とする。DSR を実現するプロトコルは複数のモバイルエージェントとステーションリエージェントによって構成することができる。図 5 にルーティング探索を行うクラス、及び直接接続しているリンクにエージェントを送信する補助ステーションリを示す。

図 4 に構築に用いるネットワークエージェント群を示す。これが Route Discovery に必要な全エージェントのソースコードであり、これだけでプロトコルが簡潔に実現できるのは、MAGNET の記述力の高さを示している。この例では説明を簡単にするため、エージェントの生成等は直接オブジェクト

```

// ブロードキャストエージェント
class BroadCaster extends MagAgent {
    Vector nodes = new Vector(); // ノード記録用
    MagAgent mes;
    BroadCaster(MagAgent m){ mes = m; }
    void onRegister(){
        Vector newNodes = magMgr.getNeighbor();
        newNodes.removeAll(nodes);
        nodes.addAll(newNodes);
        nodes.addElement(magMgr.currentNode());
        magMgr.seqRegister(mes); // メッセージの登録
        new DirectMulticast(this, newNodes, magMgr);
    }
    void onSame(MagAgent bc){ // 同じエージェントの到着
        nodes.addAll(((BroadCaster)bc).nodes);
    } // とりあえず情報を共有するのみ
}
// ソースルートエージェント
class SourceRoute extends MagAgent{
    Vector route;
    MagAgent message;
    SourceRoute(MagAgent m, Vector rt){
        message = m; route = rt; }
    void onRegister(){
        if(route.size() == 0) {
            magMgr.register(message); unRegister();
        } else { NodeID nd = (NodeID)route.firstElement();
            route.removeElementAt(0); moveTo(nd);
        }
    }
}
// ルートトレースエージェント
class RouteTrace extends MagAgent{
    Vector ns = new Vector(); // パスの記憶先
    void onRegister(){
        ns.addElement(magMgr.currentNode());
        unRegister(); }
    Vector getTrace(){ return ns ;}
}
// ルートリクエストエージェント
class RouteRequest extends MagAgent{
    NodeID dest; // 目的ノード
    RouteTrace rt = new RouteTrace(); // パスの記憶
    RouteRequest(NodeID dt){ dest = dt;}
    void onRegister(){
        if(dest.equals(magMgr.currentNode())){
            Vector ns = rt.getTrace(); // 目的地到着
            magMgr.register(new SourceRoute(
                new DSRouter(dest, ns), reverse(ns)));
        } else { magMgr.seqRegister(rt);
            unRegister();
        }
    }
}
// DSR ルータエージェント (DSR パケット生成)
class DSRouter extends MagAgent implements Router{
    NodeID dest;
    Vector route;
    DSRouter(NodeID dt, Vector rt){
        dest = dt; route = rt;
    }
    void onRegister(){
        magMgr.registerRouter(dest, this);
    }
    public void sendData(MagAgent m){
        magMgr.register(new SourceRoute(m, route));
    }
}

```

図4 DSR プロトコルエージェント

```

// ルートディスカバリ スタータ
class RouteDiscovery {
    RouteDiscovery(NodeID dest, MagManager magMgr){
        magMgr.register(
            new BroadCaster(new RouteRequest(dest)));
    }
}
// 直接マルチキャスト
class DirectMulticast {
    DirectMulticast(MagAgent msg, Vector dest
        , MagManager magMgr){
        Enumeration e = dest.elements();
        while(e.hasMoreElements())
            magMgr.copyTo(msg, (NodeID)e.nextElement());
    }
}

```

図5 ステーションナリクラス

を生成しているが、一般のエージェントシステム同様、ランタイムシステムを利用した生成手法を用いても良い。

- BroadCaster
まだ通信していない端末へメッセージを転送 (LoopFree)
 - RouteTrace
移動した端末の系列を保存
 - RouteRequest
目的の端末の判定
 - SourceRoute
中継端末の情報を持つルーティングパケット
 - DSRouter
ルーティングパケットの生成エージェント
これらのエージェントの構成を図3に示す。初期状態では、BroadCaster, RouteRequest, RouteTrace の3エージェントが階層構造を持ったまま、ネットワーク上を移動する。目的ホストに到着すると、RouteRequest が SourceRoute と DSRouter を生成し、この2エージェントが元ホストまで逆に経路をたどっていく。最終的には、元ホスト上に DSRouter が到着し、データ送信時に SourceRoute を生成する。動作を以下に順に示す。
1. RouteDiscovery により BroadCaster 以下のエージェントが生成される
 2. BroadCaster は自分自身を直接接続の未知のホストへ次々とコピーする
 3. RouteRequest は現在のホストが目的ホストかを確認する
 4. RouteTrace は現在のホストを通過記録に保持する
 5. 目的ホストへ到着するまで2までの繰り返し
 6. RouteTrace の通過記録をそのまま持つ DSRouter

```

// 動的なブロードキャストエージェント
class BroadCasterAdhoc extends MagAgent
implements LinkMonitor{// リンクモニタを実装
Vector nodes =new Vector();
MagAgent message;
BroadCasterAdhoc(MagAgent m){ message = m; }
void onRegister(){
Vector newNodes = magMgr.getNeighbor();
newNodes.removeAll(nodes);
nodes.addAll(newNodes);
nodes.addElement(magMgr.currentNode());
magMgr.seqRegister(message);
// リンクモニタとしての登録
magMgr.registerLinkMonitor(this);
new DirectMulticast(this,newNodes,magMgr);
}
// 新規ホスト発見
public void onNew(Vector newNodes){
newNodes.removeAll(nodes);
nodes.addAll(newNodes);
new DirectMulticast(this,newNodes,magMgr);
}
void onSame(MagAgent bc){// 同じエージェントの到着
nodes.addAll(((BroadCasterAdhoc)bc).nodes);
} // 配送情報を共有する
}

```

図6 アドホック環境で利用できる Broadcaster

を生成する

7. 通過記録を逆にしてSourceRoute を生成する
8. 元のホストに到着するまでSourceRoute が移動する
9. 元のホストでDSRouter がネットワークマネージャに登録される

このように、本来なら複数のパケットを理解する仕組みが各ホストに必要な DSR を、MAGNET を用いればモバイルエージェントのみで実現できる。

3.3 アドホックネットワークへの拡張

DSR はアドホックネットワーク上のプロトコルであるが、ネットワーク上をパケットが伝搬している際のネットワークの断絶には、エラーを返すのみである。MAGNET では、各エージェントがリンクをモニターすることにより、動的なネットワーク環境でもブロードキャストやルーティングを可能にする。

DSR プロトコルで用いたエージェントを動的な環境に適應できるように変更したものを図6に示す。このエージェントはリンクモニタインタフェースを実現し、onRegister 時にリンクモニタとしての登

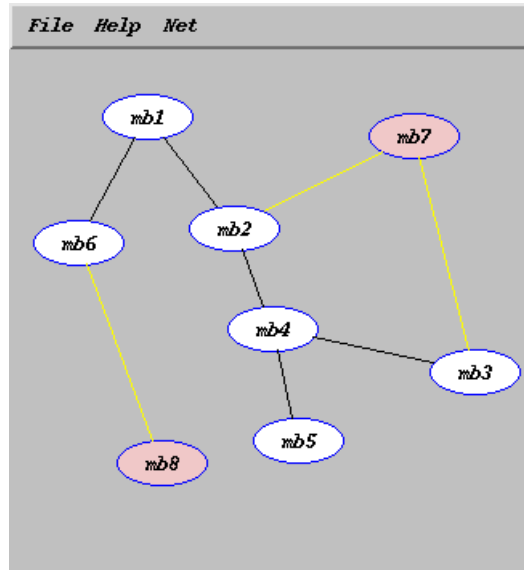


図7 MAGNET 環境の動作画面

録を行っている。onNew により、新たなホストが発見されると、改めてブロードキャストを再会する。

同様な拡張をSourceRoute に行うことにより、DSR プロトコル全体を断続的なネットワークでも動作可能にできる。

4 MAGNET エミュレーション環境

図7に MAGNET の動作画面の一部を示す。グラフ上のノードが各端末に相当しており、ノードやリンクの状態はポインティングデバイスを用いて直接操作できる。また、他のプログラムやスクリプトによる動的な変更も可能である。

各ノードは、ホスト上のエージェントマネージャにしており、リンクマネージャにより、リンクの状態が監視されている。各エージェントマネージャは実際の Java スレッドであり、異なる計算機上からスタブクラスを用いて TCP/IP で接続することができる。将来的には、スタブクラスが赤外線等の物理インタフェースを直接利用するように変更することにより、同じプログラムで、実際の通信を行うことが可能である。

現在は、赤外線エミュレーションを行なっているため、単一のホストが通信している間は他のホストは通信ができず、行った場合はエラーとなる仕組みが組み込まれている。この制限をはずすことにより、IEEE802.11 のような無線ネットワークのエ

ミュレータへの拡張も可能である。

5 関連研究

5.1 アドホックネットワーク

IETF [20]のアドホックネットワークワーキンググループ (MANET [11]) は、主にルーティングに関する研究を行っており、既存の TCP/IP を用いたアプリケーションをモバイルコンピューティングでも利用しようという発想である。しかし、IP ネットワーク上のアプリケーションは、一般に動的な端末の参加・退出に対応しておらず、ユーザの意図を通信の成立により判断するという我々の手法は利用できない。

5.2 アドホックネットワークアプリケーション

ParcTab [9]は小型の端末であり、構内に設置された赤外線送受信機を通じてサーバにアクセスし、天気予報を問い合わせたり、メールチェック等のアプリケーションを実現している。しかし、端末間の直接通信を実現しているわけではなく、自律的なアドホックネットワークを構築しているわけではない。C-MAP システム [3]は無線ネットワークを持つ携帯端末とアクティブバッジを組み合わせたガイドシステムであり、ユーザの位置情報を利用して、状況に適応したガイドを可能にしている。しかし、サーバを必要とし、アドホックなネットワーク上での利用は考えられていない。一方「なかよし」[2]や WirelessDAN [1]は PHS パケット通信や赤外線拡散通信によりアドホックネットワークを構築し、グループウェアを実現している。しかし、通信のために特別な機器を必要とし、ルーティングの機能は実現されていない。

5.3 アクティブネットワーク

インターネットの世界では、IPv4 から IPv6 への移行が実験されているが、ネットワーク機器の変更コストが問題視されるようになってきた。この問題を解決するために、柔軟にプログラムが可能なルータやネットワーク端末の構築手法が研究されており、これらを総称してアクティブネットワークと呼ばれている。

BBN の SmartPacket [15]は、Sprocket と呼ばれ

る C 言語に似た言語を Spanner という仮想マシンコードへ変換し、仮想機械上で動作させるシステムであり、1k バイトのパケット中にルーティングに関するプログラムを入れることに成功している。

MIT では ANTS [14]と呼ばれる Java ベースのアクティブネットワークシステムが研究されている。ANTS によってルータに特別な仕組みを必要とせずに MobileIP や Multicast が実現できることが示されている。

しかし、これらの研究は既存の固定ネットワークを中心に考えられており、いわゆるアドホックネットワーク上での運用は考慮されていない。

5.4 モバイルエージェントシステム

様々なモバイルエージェントシステムが開発されている [13] [18]。しかし、アドホックネットワーク上で動作することを前提としたシステムはこれまでのところ存在しない。携帯端末等への適応は考えられているが、主に固定ネットワークへの接続が主であり、端末間で直接エージェントをやりとりすることを考慮したシステムは無い。

6 まとめ

本稿では、モバイルエージェント技術を用い、アドホックネットワーク構築を行う手法を提案した。エージェントをネットワークパケットとみなし、階層的にエージェントを利用することにより、単純なエージェントの組合せにより様々なプロトコルを実現できる。

本手法に基づいたプロトタイプとして、モバイルエージェントネットワークシステム MAGNET を実現し、その上でアドホックネットワークにおいて代表的な DSR プロトコルを実現した。複数のパケットを用いる DSR を、単純なエージェントの組合せで実現できることから、MAGNET の記述性の高さが示された。また、少量のコードの追加によりアドホックな環境に対応する DSR プロトコルも容易に実現できた。

MAGNET は赤外線通信エミュレータ上に実装されており、視覚的な表示により、モバイルエージェントの動作の様子を用意に理解することができる。MAGNET は現在も開発中であり、プロトコルデザ

インのための様々なエージェントのデザインパターンの開発、状況に適応したプロトコルの選択手法、性能評価、セキュリティ等は今後の課題である。特にエージェントが移動した先の環境を認識する手法については、現在はネットワークの状況のみの認識であるが、携帯端末のバッテリー、処理能力などのハードウェアの含めて様々なものが考えられる。これらの一般化を行うことは困難であるため、認識手法そのものを動的に変更するメタな手法を用いることを考えている。

参考文献

- [1] 多鹿陽介, 岩村和昭, 池上史彦, 中村誠: 携帯情報機器の通信に適した自律無線ネットワーク Wireless-DAN の提案, 信学技報, IN 94-161(1995).
- [2] 倉島顕尚, 市村重博, 田頭繁, 前野和俊, 武次将徳, 永田善紀: 集まったその場での協同作業を支援するモバイルグループウェアシステム「なかよし」, 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 (DiCoMo97), pp.233-238(1997).
- [3] 角康之, 江谷為之, 間瀬健二: context-aware なモバイル・アシスタント, 第 55 回情報処理学会全国大会講演論文集, pp.443-444(1997).
- [4] 西部喜康, 和気弘明, 森原一郎, 服部文夫: モバイル環境下でのユーザの振舞いの解析とエージェント通信への適用法の検討 - Experiments of ICMAS'96 Mobile Assistant Project -, 信学論, Vol.J81-D-I, No.5, pp.523-531(1998).
- [5] 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 (DiCoMo98), 情報処理学会, pp.619-626(1998).
- [6] 片桐秀樹, 河口信夫, 外山勝彦, 稲垣康善: モバイルアドホックネットワーク構築のための分散アルゴリズムの提案とその実現, 情報処理学会第 57 回全国大会講演論文集, Vol.3, pp.558-559(1998).
- [7] 片桐秀樹, 河口信夫, 外山勝彦, 稲垣康善: 赤外線通信を用いた頑健なモバイルアドホックネットワーク構築手法, 情処研報, 98-MBL-7-9, pp.63-70(1998).
- [8] 佐藤一郎, 高橋美奈子, 棚橋杏子, 吉野裕子: モバイルエージェントの階層的な構成と移動, 日本ソフトウェア科学会第 15 回大会, pp.249-252(1998).
- [9] Bill N. Schilit, Norman I. Adams, and Roy Want: Context-Aware Computing Applications, *In Proceedings of the Workshop on Mobile Computing Systems and Applications*, Santa Cruz, CA, pp.85-90(1994).
- [10] Charles E. Perkins and Pravin Bhagwat: Highly Dynamic Destination-Sequenced Distance-Vector Routing (DSDV) for Mobile Computers, SIGCOMM'94 (1994).
- [11] S. Corson and J. Macker: Mobile Ad hoc Networking (MANET): Routing Protocol Performance Issues and Evaluation Considerations, Internet Draft, draft-ietf-manet-issues-00.txt (1997). work in progress
- [12] Charles E. Perkins: Ad Hoc On Demand Distance Vector (AODV) Routing, Internet Draft, draft-ietf-manet-aodv-02.txt (1998). work in progress.
- [13] Danny B. Lange, Mitsuru Ohshima: Programming and Deploying Java Mobile Agents with Aglets, Addison Wesley(1998).
- [14] David J. Wetherall, John V. Guttag and David L. Tennenhouse: ANTS: A Toolkit for Building and Dynamically Deploying Network Protocols, IEEE OPENARCH'98,(1998).
- [15] Beverly Schwartz, Wenyi Zhou, Alden W. Jackson, W. Timothy Strayer, Dennis Rockwell, Craig Partridge: Smart Packets for Active Networks, BBN Technologies (1998). (<http://www.net-tech.bbn.com/smtpkts/smtpkts-index.html>)
- [16] David B. Johnson and David A. Maltz: Dynamic Source Routing in Ad Hoc Wireless Networks, *In Mobile Computing*, Tomasz Imielinski and Hank Korth eds., Chapter 5, Kluwer Academic Publishers, pp.153-181, 1996.
- [17] Hosh Broch, David A. Maltz, David B. Johnson, Yih-Chun Hu, and Jorjeta Jetcheva: A Performance Comparison of Multi-Hop Wireless Ad Hoc Network Routing Protocols, Mobicom98, 1998.
- [18] General Magic: Odyssey (<http://www.genmagic.com/technology/odyssey.html>)
- [19] Infrared Data Association: Serial Infrared Link Access Protocol, Version 1.1 (1996). (<http://www.irda.org>)
- [20] Internet Engineering Task Force: (<http://www.ietf.org>)
- [21] BlueTooth: (<http://www.bluetooth.com>)
- [22] HomeRF: (<http://www.homerf.org>)
- [23] IrBus: (<http://www.irbus.org>)