

漸増的に端末を認識するアドホックネットワーク構築手法

戸崎貴資[†] 河口信夫[†] 外山勝彦^{††} 稲垣康善[†]

携帯端末の急速な普及により、モバイル環境下において端末間で情報を直接交換・共有することが求められている。この要求を満たすために、片桐らは赤外線通信を用いたアドホックネットワーク構築手法を提案した。この手法では、ネットワーク構築のため端末には状態があるとしているが、用いる状態が複雑であるため、実装や拡張が難しいといった問題点が存在した。また、全端末は、ネットワークを完全に把握した後に通信を開始するため、ネットワークの構築に時間がかかる場合にはデータ通信の開始が遅延した。本稿では、ネットワーク構築の際の通信に端末の状態を用いないアドホックネットワーク構築手法を提案する。本手法において、各端末は、探索により発見した近傍の端末とは即座に通信可能であり、ネットワークを完全に構築するまでデータ通信を待つ必要はない。また、各ノードは既知端末とノード情報を交換することにより、近傍以外の端末を漸増的に認識する。各端末は、他の端末の状態を考慮することなく、探索とノード情報の交換のみにより、ネットワークを構築できる。

Incremental Construction of Ad-Hoc Network

Takashi Tosaki[†] Nobuo Kawaguchi[†] Katsuhiko Toyama^{††} Yasuyoshi Inagaki[†]

In this paper, we propose an incremental construction method for ad-hoc networks based on infrared communication. Each mobile host periodically probes other hosts while exchanging messages between known hosts. When a host discovers new neighbor hosts, it exchanges the updated node information with all neighbor hosts. Node information contains two sequential numbers which avoid the looping and the updating problems, and it is piggybacked onto messages to decrease the time costs required for each infrared connection.

1 はじめに

ノートパソコンやハンドベルドPCに代表される小型で高性能な携帯端末が普及し、いつでもどこでも必要な時に様々な情報の利用が可能になりつつある。より多くの人々が携帯端末を持つようになると、会議など多くの人々が集まる場合は、同時に多数の携帯端末が集まる場となる。そのような場で、携帯端末間でネットワークを構築し、端末に保存されている情報の交換・共有を直接行うことが可能となれば、これまで人が行ってきた共同作業の支援が可能になることが期待される。このように、必要に応じて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われている [7][8][9][10]。

アドホックネットワーク構築手法の一つとして片桐らは、赤外線通信を用いたアドホックネットワーク構築手法を提案している [1][3]。この手法では、端末には状態があり、他の端末の状態を考慮しながら通信を行いネットワークを構築する。しかし、ネットワークの動的変化や通信エラーに対応するため、用いる状態が複雑になり、その結果、実装や拡張が難しいといった問題が生じた。また、全ての端末が、ネットワークを完全に把握した後にデータ通信を開始するため、端末数が多くネットワーク構築に時間がかかるときにはデータ通信の開始が遅れるということも問題となった。

本稿では、ネットワークの把握に完全さを求めないアドホックネットワーク構築手法として、漸増的に端末

を認識するアドホックネットワーク構築手法を提案する。各端末は探索により近傍の端末を発見し、既知の端末とノード情報を交換することによりネットワークを漸増的に把握する。本手法はネットワーク構築に端末の状態を用いない。また、認識した端末とは即座に通信が可能であり、ネットワークを完全に構築するまで通信の開始を待つ必要はない。本手法では通信可能端末は既知端末のみであるが、ブロードキャスト通信を行うことにより、全端末へのデータ送信が可能である。

本稿では提案する手法の実現についても報告する。具体的な通信手段としては任意の通信媒体の利用が可能であるが、本研究では、多くの携帯端末に実装されている赤外線通信 (IrDA [4]) を用いる。赤外線通信は指向性が高く通信到達距離が短いため、隠蔽性が高く、名刺交換等のプライベートな情報の通信に適している。

以下、本稿ではまず2章において、片桐らのアドホックネットワーク構築手法について説明し、その問題点を述べる。3章で今回提案する漸増的に端末を認識するアドホックネットワーク構築手法について述べる。また、提案した手法の実現については4章で述べる。5章では関連研究との比較について述べる。

2 片桐らのアドホックネットワーク構築手法

以下本稿では、携帯端末をノードと呼び、固有な識別子 n_i を持つ円で表す。ノード間の通信路をリンクと呼び、各ノードが認識しているリンクを実線で表す。あるノードに対して直接通信可能なノードを隣接ノードと呼ぶ。また、ネットワーク構築に必要な情報をノード情報、アプリケーションからの送信データをメッセージとそれぞれ呼ぶ。

片桐らの手法は、アドホックネットワークを構築する際の問題を単純化するため、最初は、ネットワーク構

[†]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

築中のトポロジの動的な変化や、通信エラーはないという仮定(仮定1)のもとでネットワーク構築アルゴリズムを作成し[1]、次にそれをトポロジの動的変化や通信エラーを考慮するように拡張した[3]。

仮定1のもとで作成したアルゴリズムは以下の3つのステップから成り立つ。ただし、アルゴリズム中の探索開始ノードとはネットワーク構築手続きを開始するノードのことである。

Step1 構築するネットワークを木構造、探索開始ノードを木構造の根とみなす。探索開始ノードから木構造の葉に相当する末端のノードへ向かって、各ノードが隣接ノードの発見手続きを行い、発見したノードにネットワーク情報を送信するという動作を繰り返し行う。

Step2 Step1で用いた経路を逆にたどり、末端ノードから探索開始ノードへ向かってネットワーク情報を送信する。これにより、探索開始ノードは全てのネットワーク情報を獲得する。

Step3 全てのノードが同じネットワーク情報を持つように、探索開始ノードから末端ノードまでStep1で用いた経路をたどり、各ノードが隣接ノードへネットワーク情報を送信するという動作を繰り返し行う。

上記の3つのステップを経て、各ノードがネットワーク内の全ノードを完全に把握した後、メッセージ通信可能となる。

次に片桐ら[3]は、仮定1を取り払い、実際の動的なネットワークに対応するようアルゴリズムを拡張した。しかし、トポロジの動的変化や、通信エラーが各ノードの状態の間の整合性を乱すことが問題となった。そのため、ネットワーク構築の頑健性を実現するために、様々な状態に対応可能なイベントテーブルが用いられた。しかし、イベントテーブルには、各状態毎にノードの参加や退出といった全てのイベントに対する動作を記述しなければならないため、イベントテーブルが複雑となり、拡張や実装が難しいという問題点が生じた。また、ネットワーク構築後でないとメッセージ通信ができないため、端末数が多くなりネットワーク構築に長時間必要な場合、メッセージ通信の開始が遅れるという問題点もあった。

3 漸増的に端末を認識するアドホックネットワーク構築手法

本章では、完全にネットワークを構築することを目的とせず、出来る限り短時間で近傍のノードとの通信を可能とし、他のノードは漸増的に把握するアドホックネットワーク構築手法を提案する。各ノードは探索により、直接通信可能なノードを発見する。また、各ノードは、保持しているノード情報を相互に交換することにより、時間の経過と共にネットワーク全体を漸増的に把握する。

片桐らの手法では、各ノードには、ネットワーク構築アルゴリズムの各段階に対応する状態を与えなければ

ならなかった。本手法では各ノードは、ネットワーク構築に段階がなく、探索とノード情報の送受信を繰り返すのみでネットワークを構築できるため、各ノードには状態の概念が不要である。

本手法は具体的な通信手段として多くの携帯端末上に実装されている赤外線通信を対象にする。赤外線通信は指向性が高く、通信到達距離も短いため、プライベートな情報の通信に適している。

3.1 解決すべき問題点

前述したアドホックネットワーク構築手法を考える場合、片桐らの手法では考慮する必要のなかった新たな問題点が生じる。それらの問題点を以下に挙げる。

(1) 未知ノードへの送信

片桐らの手法ではネットワークを完全に構築した後にメッセージ通信を行うため、通信可能ノードはネットワーク内の全ノードである。本手法では漸増的にネットワークを把握するため、通信可能ノードは既知ノードのみとなり、未知ノードへのメッセージ送信方法が必要となる。

(2) ノード情報の整合性

本手法では、各ノードは探索とノード情報の送受信によりネットワークを構築するが、送信するノード情報について、古い情報と新しい情報が入り乱れる可能性がある。そのため、ノード情報に整合性を持たせる方法が必要となる。

(3) 赤外線通信のコスト

赤外線通信は、通信速度に比べ、コネクション確立の時間的コストが大きい。赤外線通信規格IrDA[4]では、最大4Mbpsの通信速度が可能と定めているが、コネクション確立には最低で500ms必要である。そのため一回の通信で、できる限り多くの情報を送信し、通信回数をできる限り少なくする方法が必要となる。

各問題に対する解決策は、問題(1)に対しては3.7節で述べる。問題(2)の解決策については3.3節で述べる。また、3.4節で問題(3)に対する解決策について述べる。

3.2 準備

本手法において各ノード n_i は、ネットワーク構築のために以下の情報を保持する。

- ルーティングテーブル
認識したノードへの経路が記述されているテーブルで、メッセージ送信時に参照される。詳細は3.6節で述べる。
- 獲得したノード情報集合
 n_i が作成したノード情報と、他ノードから受信したノード情報からなる集合。
- シーケンシャル番号 SN_i

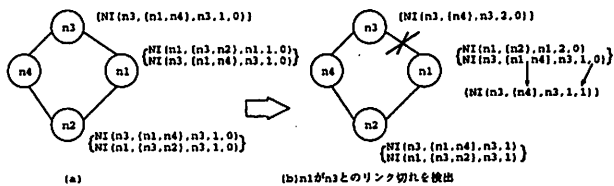


図 1: トポロジの変化により NI に矛盾が生じる例

探索の結果が異なるたびに n_i により更新される。 SN_i はノード情報の新しさを表すために用いられる。詳しくは 3.3 節で述べる。

- DTimer
探索のタイミングの決定に用いられる。3.5 節で述べる。
- NTimer(n_j)
隣接ノード n_j へのノード情報送信のタイミングの決定に用いられる。3.5 節で述べる。
- ブロードキャストシーケンシャル番号 $BCSN_i$
ブロードキャストメッセージのループを防ぐため用いるシーケンシャル番号。3.7.2 節で述べる。
- 他ノードのブロードキャストシーケンシャル番号 $BCSN_i$ とともにブロードキャストメッセージのループを防ぐために用いられる。3.7.2 節で述べる。

3.3 ノード情報

各ノード n_i は、探索の結果に基づき以下の形式のノード情報を作成する。

$$NI(n_i, N_i, n_k, SN_i, SNO_i)$$

ここで、 n_i は自ノード ID、 N_i は探索により発見した隣接ノード ID の集合である。 n_k はこの NI の受信元ノードである。NI を受信したノードは受信時に NI の第 3 項を受信元ノード ID に更新する。あるノード n_j へのノード情報の送信は、第 3 項が n_j である NI は送信しない。また、第 1 項が n_j である NI は、その NI を作成したノードであるため、確実にその NI を保持しているはずであり、それを送信することは省略する。

NI には DSDV[7] の手法を参考に、3.2 節で述べた各ノードが更新するシーケンシャル番号 SN_i が付加されており、これが、ノード n_i のリンクに関する情報の新しさを表している。すなわち、 SN_i の値が最も大きい NI をノード n_i について最新の情報として扱う。また、 n_i 以外のノードが、受信した NI の内容に矛盾を発見する場合がある。例を図 1 に示す。図 1(b) では、ノード n_1 の保持する情報 $NI(n_3, n_1, n_4, n_3, 1, 0)$ は、実際 n_1 と n_3 の間のリンクが切れているにもかかわらず、 n_1 と n_3 が隣接していることを示している。そのような場合、矛盾を発見したノードは矛盾のある NI の内容を変更しなければならないが、変更後の情報の方が変更前より新しい情報として扱わなければならない。そのため、NI には他ノードが更新するシーケンシャル番号として SNO_i が付加されており、矛盾内容を修正する際、 n_i 以外のノードにより更新される。各ノードは SN_i の値が同じなら SNO_i の値が大きい NI を n_i について最新

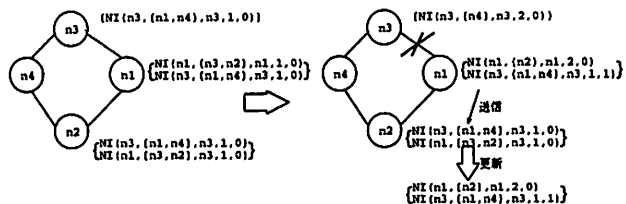


図 2: シーケンシャル番号の利用例

の情報として扱う。図 1 では、 n_1 は n_3 が作成した NI の第 2 項の矛盾を修正し、第 5 項の値を更新している。

図 2 にシーケンシャル番号の利用例を示す。ノード n_3 は n_1 とのリンク切れを認識するとシーケンシャル番号を変更する。 n_4 は n_3 から NI の集合を受信し、以前保持していた NI の集合のシーケンシャル番号と比較して、受信した情報の方が新しい情報と認識する。従って、上記の 2 つのシーケンシャル番号を用いることにより、3.1 節で述べた、ノード情報の整合性の問題を解決できる。

3.4 通信回数の削減

本手法では、メッセージ送信の際、必要であればノード情報を付加して送信することにより通信回数を減らすことにする。これにより 3.1 で述べた赤外線通信のコストの問題に対処できる。

3.5 ネットワーク構築アルゴリズム

本手法における各ノードの動作の概要を以下に示す。ただし、動作の順序は特に決められていない。

- 適当な時間間隔 α が経過した後、他のノードを探索し、隣接ノードとして登録する。探索結果に変更があればルーティング計算を行う。
- 探索の結果に変更があれば、適当な時間間隔 β だけ待機し、ノード情報をまだ送信していない隣接ノードへ、ノード情報のみ送信する。
- 他ノードからノード情報を受信すると、最新のノード情報を用いてルーティングテーブルを更新する。
- ノード情報の受信によるトポロジ変化を認識した後、適当な時間間隔 γ だけ待機し、ノード情報をまだ送信していない隣接ノードへノード情報のみ送信する。
- メッセージを送信する場合、メッセージにノード情報を付加して送信する。
- メッセージ中継の際、メッセージにノード情報を付加して送信する。

以上により、各ノードは直接通信可能なノードは α 時間で認識し、他のノードは漸増的に認識する。トポロジ変化の認識後即座にノード情報を送信すると、ノード情報の送信のみでネットワークが混雑する可能性があるため、ノード情報を送信するまでに待機時間 β, γ を設けている。

β と γ の関係は、 $\beta \leq \gamma$ である。探索によるノード情報は自らが作成したノード情報であるので、他ノードから受信したノード情報よりも信頼できる。そのため、

表 1: イベントテーブル

Event	Action(s)
expire DTimer	Discovery(); setDTimer(); if isChangeTopology() then $\forall n_j \in N$ setNTimer(β, n_j);
expire NTimer(n_j)	sendNI(n_j , getNI(n_j));
requestSend(msg)	sendUnion(nextHop(msg), msg, getNI(n_j));
rcvNI(n_j , NI)	analyzeNI(NI); if isChangeTopology() then $\forall n_k \in N \wedge n_k \neq n_j$ setNTimer(γ, n_k);
rcvMsg(n_j , msg)	if isNeedRelay(msg) then sendUnion(nextHop(msg), msg, getNI(n_k));
rcvUnion(n_j , msg, NI)	analyzeNI(NI); if isChangeTopology() then $\forall n_k \in N \wedge n_k \neq n_j$ setNTimer(γ, n_k); if isNeedRelay(msg) then sendUnion(nextHop(msg), msg, getNI(n_k));

探索の結果による情報をより積極的に早く送信すべきである。故に $\beta \leq \gamma$ となる。

なお、待機時間の測定のために、各ノードは次の2種類のタイマーを保持する。

- DTimer
探索のタイミングの決定に用いる。各ノードはタイマーを初期状態で α に設定し、タイマーが0になったら探索を行い、再び α にセットする。
- NTimer(n_j)
登録した隣接ノードごとに用意されるタイマー。引数の n_j は隣接ノードである。探索によりトポロジ変化を認識するとタイマーは β にセットされる。また、ノード情報を受信しトポロジ変化を認識すると、タイマーは γ にセットされる。ただし、タイマー作動中にトポロジ変化が起きても β や γ にリセットされない。タイマーが0になると引数の隣接ノードへノード情報を送信する。ただし、NTimer(n_j) が作動中に、ノード n_j へメッセージを送信する際、メッセージにノード情報を付加して送信し、タイマーは0にセットされる。

各ノードの動作の詳細は表1のイベントテーブルに示す。表1中のイベントの説明は表2に示し、イベントテーブル中の関数の説明は表3に示す。ただし、表中の N は隣接ノード集合である。

3.6 ルーティングテーブル

各ノードは、探索やノード情報の受信により得た NI の集合を基に、全域木の最小パスを求めるアルゴリズムによりルーティングテーブルを作成する。ルーティングテーブルには既知の宛先ノードとその宛先ノードへの最短経路上の次の中継先が示されている。

3.7 メッセージ

本手法では、漸増的にネットワークを把握するため、通信可能なノードは既知ノードのみである。ただし、ブロードキャスト通信を行うことで、未知ノードも含むネットワーク内の全ノードへのメッセージ送信が可能となる。これにより3.1で述べた未知ノードへの送信の問題に対処できる。以下、既知ノードへの送信に用いる通常メッセージとブロードキャストに用いるブロードキャストメッ

セージの2種類のメッセージの形式と配送手順について説明する。

3.7.1 通常メッセージ

通常メッセージは以下の形式である。

$RM(n_i, F, data)$

ここで n_i は送信元ノードID、 F は送信先ノードのIDのリスト、 $data$ はアプリケーションからの送信要求データである。

メッセージの配送は、ルーティングテーブルを参照し、最短経路となる中継先にメッセージを送信し、目的のノードへメッセージが到達するまで中継を繰り返す。

3.7.2 ブロードキャストメッセージ

ブロードキャストメッセージは以下の形式である。

$BM(n_i, BCSN_i, data)$

n_i は送信元ノードID、 $data$ はアプリケーションからの送信要求データである。 $BCSN_i$ は3.2節で述べたブロードキャストシーケンシャル番号で、 n_i がブロードキャストを行うたび n_i により更新され、メッセージのループを防ぐために用いられる。

3.2節で述べたように、各ノードは、他ノードのブロードキャストシーケンシャル番号を保持しており、保持している番号より新しい番号を持つメッセージを受信したときのみ、受信元ノード以外の全隣接ノードへメッセージを送信する。このとき、以前に保持していた番号は新しい番号に書き換えられる。また、ブロードキャストの際ノード情報を付加することにより、全てのノードに自分を認識させることもできる。

3.8 片桐の手法との比較

表4に本手法と片桐の手法との比較を示す。片桐の手法ではネットワーク構築後に通信を開始するが、本手法では最初の探索後は常に既知ノードと通信が可能である。また、片桐の手法では、ネットワークに参加している全てのノードとネットワーク構築後に通信可能であるが、本手法では既知ノードとの通信のみ可能である。ただし、ブロードキャスト通信を行うことにより、全ノードとの通信が可能である。また、片桐らの手法ではネッ

表 2: イベントの説明

Event 名	意味
expire DTimer	DTimer がタイムアウトする。
expire NTimer(n_j)	隣接ノード n_j のタイマー NTimer(n_j) がタイムアウトする。
requestSendMsg(msg)	アプリケーションからメッセージ msg の送信要求を受ける。
rcvNI(n_j ,NI)	ノード n_j からノード情報 NI を受信する。
rcvMsg(n_j ,msg)	ノード n_j からメッセージ msg のみを受信する。
rcvUnion(n_j ,msg,NI)	ノード n_j からメッセージ msg とノード情報 NI を受信する。

表 3: 関数の説明

関数名	処理内容
Discovery()	探索を行い、結果が以前と異なればルーティングテーブルを更新する。
analyzeNI(NI)	受信したノード情報 NI を用いてルーティングテーブルを更新する。
isChangeTopology()	トポロジが変更されたら true を返す。
nextHop(msg)	メッセージ msg の宛先ノードの nextHop をルーティングテーブルから参照し、その ID を返す
setDTimer()	DTimer を α にセットする。
setNTimer(const, n_j)	隣接ノード n_j の NTimer が作動していなければ const 値にセットする。
sendNI(n_j ,NI)	隣接ノード n_j へノード情報 NI を送信する。
sendUnion(n_j , msg,NI)	NTimer(n_j) が作動していれば、ノード情報 NI とメッセージ msg を同時に送信し、NTimer(n_j) をリセットする。そうでないなら msg のみを送信する。
isNeedRelay(msg)	メッセージ msg を中継する必要があるら true を返す。
getNI(n_j)	n_j へ送信するノード情報を返す。

トワーク構築のため、各ノードに複雑な状態を与えているため拡張や実装が難しいが、本手法では状態という概念がないため、拡張や実装は容易である。また、片桐らの手法では、ノード情報とメッセージを同時に送信することは考えられていないため、赤外線通信におけるコネクション設定のコストの問題が解決できていない。

4 手法の実現

提案したアドホックネットワーク構築手法を Java 言語 (JDK1.1) を用いて実装した。また、実機での実装におけるプログラミングのデバッグや動作の評価実験を容易にするため、赤外線通信をエミュレートする環境を構築した。図 3 にエミュレーション環境上での動作の様子を示す。左上のウィンドウがエミュレーションサーバであり、各ノード間の通信を管理する。ウィンドウ上の円が各ノードを表し、ノード間のリンクが直接通信リンクを表す。サーバ上でノード間の接続状況を変更することにより、動的なネットワークトポロジの変更をエミュレートすることができる。右側の複数の小さなウィンドウは各ノードを実現するプログラムであり、通信可能ノードや保持しているノード情報、および通信の様子が表示されている。

エミュレーションサーバとの通信ライブラリと赤外線通信ライブラリとは互換性があり、各ノードの動作を実現しているプログラムは変更することなく、実際の赤外線通信上でアドホックネットワークの構築ができる。

5 関連研究

RIP[5] は LAN に広く利用されている distance vector 型のルーティングプロトコルである。RIP ではループの回避ができないため、トポロジの動的変化が頻繁に起こるアドホックネットワークへの適用には向いていない。

一方、IETF[6] 内では MANET というアドホックネットワークのワーキンググループにより、AODV[8] や DSR[9]、DSDV[7] などのルーティングプロトコルやデータの転送などが検討されており、それらの手法の比較実験も行われている [10]。

DSDV[7] は distance vector 型のルーティングプロトコルであり、シーケンシャル番号をルーティング情報に付加することにより、ループの回避を保証している。DSDV は定期的にルーティング情報の送信が行われ、また、トポロジ変化が生じた時にもルーティング情報を送信しなければならない。本手法はトポロジが変化した時のみノード情報を送信するため、DSDV より少ない通信量でルーティングができる。

DSR[9] はメッセージ送信のたび、必要に応じて destination へのルーティング情報を探索する手法である。destination、もしくは destination までのルーティング情報を持つノードをブロードキャストにより探索し、発見した場合に、ルーティング情報を source に返し、source 側からパケットに全ての中継経路情報を含めて送信する。パケット送信中に経路の変更があればパケットは source に返信され、再び探索を行い再送される。そのため、DSR では、メッセージ送信要求から実際にパケットが届くまでに大きな遅延が生じる可能性がある。

表 4: 片桐の手法と本手法の比較表

項目	片桐の手法	本手法
通信開始	ネットワーク構築後	最初の探索終了後
通信可能ノード	全ノード	既知ノードのみ
通信の種類	マルチキャスト	ブロードキャスト、マルチキャスト
実装、拡張の容易性	状態が複雑であり難しい	状態がないため容易
コネクションのコストの問題	考えられていない	メッセージにノード情報を付加し通信回数を減らす

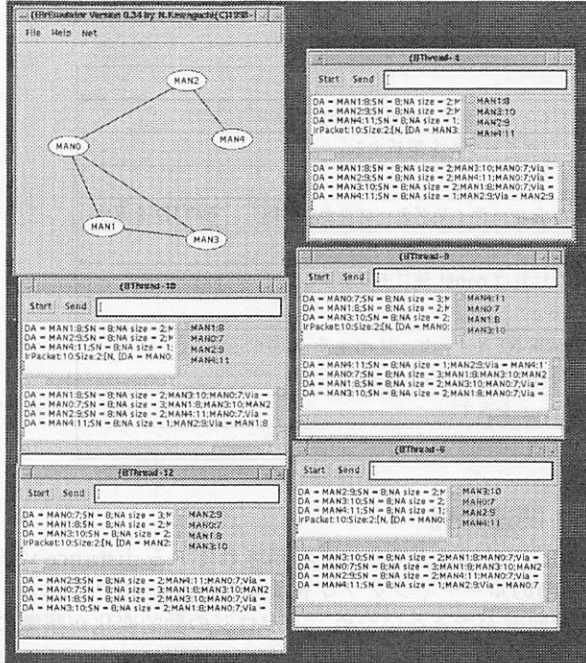


図 3: エミュレーション環境上での実行画面

本手法では既知端末へは即座に通信でき、中継ノードを指定していないため、経路に変更があっても中継先で経路を変更できるため待機することはない。また、本手法ではブロードキャスト通信を行うことにより、探索を行うことなく未知ノードへの送信ができる。

AODV[8]はDSRとDSDVを組み合わせたルーティングプロトコルである。DSRのようにメッセージ送信のたび、必要に応じてdestinationへのルーティング情報を探索する。用いられるルーティング情報は、DSDVのように、宛先へのホップ数とnext hop、シーケンシャル番号が付加されたdistance vector型の情報である。DSRと同じく、メッセージの配送の遅延という点が本手法とは異なる。

なお、本節で述べた全ての手法は主にIP上での実装を考慮しており、赤外線通信上での実装は考えられていない。そのため本稿で提案する手法のように、ノード情報とメッセージを同時に送信し通信回数を減らすことは考えられていない。

6 まとめ

本稿では新しいアドホックネットワーク構築手法として、漸増的に端末を認識するアドホックネットワーク

構築手法を提案した。各端末は、探索により発見した近傍の端末とは即座に通信可能であり、相手の端末の状態を考慮せず、ノード情報を交換することにより漸増的にネットワーク全体を把握する。この手法を考える上で生じる問題として、未知ノードへの送信方法、ノード情報の整合性、赤外線通信のコストの問題があったが、それぞれ、ブロードキャスト通信、ノード情報に付加する2種類のシーケンシャル番号、メッセージにノード情報を付加することによる通信回数の削減といった方法で問題を解決した。また、Java言語を用いて実装し、赤外線通信エミュレーション環境上で動作確認を行った。

今後の課題として、本手法の評価や他の手法との性能の比較などが挙げられる。また、端末数の増加に伴うスケーラビリティも問題となる。

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, 情報処理学会, 分散, 協調とモバイルワークショップ論文集 (DiCoMo98), pp619-626(1998).
- [2] 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, vol.3, pp.558-559 (1998).
- [3] 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, 情報研報, 98-MBL-7-9, pp.63-70(1998).
- [4] Infrared Data Association: Serial Infrared Link Access Protocol, version 1.1 (1996).
- [5] C. Hedrick: Routing Information Protocol, Internet Request For Comments RFC1058, (1988).
- [6] Internet Engineering Task Force: (<http://www.ietf.org>)
- [7] Charles E. Perkins, Pravin Bhagwat: Highly Dynamic Destination-Sequenced Distance-Vector Routing (DSDV) for Mobile Computers, SIGCOMM'94(1994).
- [8] Charles E. Perkins: Ad Hoc On Demand Distance Vector (AODV) Routing, Internet Draft, draft-ietf-manet-aodv-02.txt(1997) work in progress.
- [9] David B. Johnson, David A. Maltz: Dynamic Source Routing in Ad Hoc Wireless Networks, In Mobile Computing, edited by Tomasz Imielinski and Hank Korth, Chapter 5, Kluwer Academic Publishers, pp.153-181(1996).
- [10] Josh Broch, David A. Maltz, David B. Johnson, Yuh-Chun Hu, Jorjeta Jetcheva: A Performance Comparison of Multi-Hop Wireless Ad Hoc Network Routing Protocols, Mobicom98(1998).