

アドホックネットワークにおいて冗長経路を維持する経路制御方式

Routing algorithm maintaining redundant paths for ad-hoc networks

立命館大学情報理工学部 西村 俊和

概要: 本研究では音声や動画のストリーミングを行うことを想定し, アドホックネットワークにおいて再経路探索時の遅延を減らすため, データ通信中定期的にルート探索を行うことによって, 常に信頼性の高い予備経路を取得する手法を提案した. また, 人間が携帯端末を持ち歩いて構成されるアドホックネットワークに着目し, ネットワークシミュレータ上で特性評価を行った.

1. はじめに

従来の移動体通信ネットワークでは固定された基地局が必要であり, また電波帯域有効利用のためその送信電力には制限があるため, そのサービスエリアは通常基地局の近隣に制限されている. また, 災害等で基地局が使用不可能の場合は通信を行うことができない. これに対し, アドホックネットワークでは, 基地局に依存せず, 端末同士が直接, あるいは端末を中継しながら通信を行い, 自律分散的ネットワークを構成することが可能である. よって, 災害時や軍事利用, イベント会場などで一時的にネットワークが必要とされる状況での利用が期待されており, 将来的に非常に有用である技術であるため, 現在, 様々な研究が行われている.

アドホックネットワークは, 携帯端末のみで構成されており, 携帯端末は常に同じ場所に存在しているとは限らないため, そのネットワークポロジは常に変化していく. よって, その経路も常に変化していくため, 安定な通信を行うためには工夫が必要である. アドホックネットワークでは, 通信経路が変化した場合, 再び宛先ノードまでの経路を探索しなければならなくなり, この探索している間は, データ通信は行われず, 遅延が発生する. この遅延は音声や動画のストリーミングなどリアルタイムアプリケーションを使用した際に大きなボトルネックになるであろうと予想される.

よって, 本研究ではこの再経路探索の時間を短縮するために, 信頼性の高い予備経路をデータ通信中に取得し, 再経路探索時間を短縮する方法を提案する.

2. アドホックネットワーク

アドホックネットワークとコミュニケーション

アドホックネットワークは基地局によらず, 無線通信の行える端末により構成されるネットワークである. 通信のインフラストラクチャが必要でないため, 限られた範

囲内で安価にネットワークを構成することができる. 近年の無線通信技術の発達と計算機の小型軽量化に伴い, 個人が携帯電話や携帯端末を持ち歩くことが日常化してきた. 現在広く用いられている携帯電話等の移動体通信は, 電話やインターネット上のサーバへの接続手段であるので, 広域を覆って遠隔地の人々が距離を越えて通信を行うことができる. 一方, 我々の生命活動や社会活動は通常局所的であり, また運輸や物流の現技術は情報通信と比べて低速で高コストである. 従って, 通常は居住地域や勤務地域等における近くの商業, サービスや隣人等, 地理的な局所性を持った人々と共に生活を行っていると思われる. このような日常生活でのコミュニケーションを支援するためには, 必ずしも広域を覆う通信手段だけでは十分でなく, その局所性を有効利用できる手法が必要であると考え.

そこでここでは, アドホックネットワークによって近隣の人々が結ばれる状況を想定する. これまで対面で人と人が情報をやりとりするには, 聞き間違いや勘違いの発生し易い音声か, あるいは印字した資料, 受け取り手がその場で内容を関知することのできない記録メディアの授受を行う方法が主流であった. しかし携帯端末間のアドホックネットワークが発達することによって安価に対面でデータ交換ができ, しかも内容をその場で確認して利用することが可能となる. 電話番号や住所といった名刺などで授受可能な短い情報のみならず, 複数のURLや鮮明な写真, 大量のデータを送受信することも容易である. 記録メディアを用いれば, 予め内容の決まっているデータについては同様のことは可能であるが, テレビ動画のストリーミングのようにデータ量が記録メディアを超越して原理上記録困難なもの, あるいは電話の音声のように会話終了時までデータが蓄積されているは本来の機能が損なわれるものについては適切でない. その本研究では, このように地理的な局所性を持った人々の日常

生活でのコミュニケーションを支援するため、ストリーミングに適したアドホックネットワークの構成方法を議論するものとする。

経路制御

アドホックネットワークにおける経路制御は、主に IETF(Internet Engineering Task Force)の MANET (Mobile Ad-hoc Networks)作業部会[1]によって、開発や策定が行われており、現在、DSDV (Destination Sequenced Distance Vector Routing)[2]、DSR (Dynamic Source Routing)[3]、AODV (Ad-hoc On-demand Distance Vector routing)[4]、TORA (Temporally-Ordered Routing Algorithm)[5]など、様々な経路制御が提案されている。アドホックネットワークの経路制御は大きく分けてオンデマンド型とテーブル駆動型に分類される。また、両者の特徴を合わせたハイブリッド型経路制御も存在する。

一般的にアドホックネットワークは通信中にノードの移動や消滅によって、経路が切断され、通信が不可能な状態になることが多い。DSRではルート探索とルートメンテナンスを用いて再接続を行うが、ここで問題となるのは、過去に記録されたルートキャッシュである。何故なら、アドホックネットワークのように、頻繁にノードが移動しネットワーク構成が変化するネットワークにおいては、少しでも昔の経路は利用できないことが多いと予想されるからである。また、送信ノードは、現在のルートキャッシュ内のルートが利用可能かどうかかわからないため、現在使用中のリンクが切断された場合、キャッシュされているルートが利用可能不可能にかかわらず、そのルートを使用してパケットを送信せざるをえない。もし、そのルートが過去の情報で、既に利用不可能となっている場合は、ルートメンテナンスによりリンク切断を発見し、その後、ルートキャッシュを再検索、最悪の場合、再びルート探索を行う必要がある。再接続中は一時的に通信が途絶えるため、以上のように再経路探索に時間を浪費した場合、音声や動画のようなリアルタイム性を要するアプリケーションに対して、大きなボトルネックとなることが予想できる。

そのために、現在、DSRを拡張した経路制御の提案が行われている。特に、再接続時間の短縮を目的としたものとして、Multipath DSR[6]、FR-DSR[7]等がある。Multipath DSRはDSRでルート探索の際に複数の経路を取得し、リンクが切断した場合、保持しておいた予備経路を使用し、再経路探索をなるべく行わないようにすることで、再経路探索時間の短縮を図っている。予備経路の取り方としては、主経路にまったく関与しない予備経路を取る方法と、主経路の中間ノードを含む予備経路を取る方法の二種類が提案されている。FR-DSRはMultipath DSRのようにルート探索の際に

複数の経路を取得するが、予備経路の信頼性を向上させるため、データ通信中に保持しておいた予備経路を定期的にチェックし、その経路が利用可能かどうか確認する。取得している予備経路が無くなればデータ通信中にでもルート探索を行い、予備経路を取得する。以上のプロトコルは、それぞれ再経路探索時間の短縮を目的としたプロトコルであるが、複数経路を取得する際のループの問題や、予備経路の信頼性において、問題が残っている。

テーブル駆動型の経路制御では、ネットワーク内のすべての端末が、それぞれ全端末へのルート情報をテーブルとして保持し、定期的にビーコンを交換することでその情報を最新に保つ。しかし、トポロジーが変化する度に全端末がルート情報を更新するため、ノードが頻繁に動くアドホックネットワークでは更新メッセージが大量に発生してしまい、適していない。一方、オンデマンド型の経路制御は、通信要求が発生したときのみ、宛先端末へのルートを探索する。そのため通信要求発生から実際にデータを送信するまでに時間がかかってしまうが、トポロジー変化によるルート情報の更新が無い場合、テーブル駆動型よりもトポロジー変化の激しいアドホックネットワークの経路制御に適している。本研究では、ストリーミング通信など、リアルタイム性を必要とする利用法を前提としている。また、各ノードが頻繁に移動し、トポロジーの変化が激しいネットワークにも耐えることができないため、経路探索をオンデマンドで行い、常に信頼性の高い予備経路を取得するような経路制御の提案を行う。

3. 提案方式

本研究では、前節で指摘したような問題点を解決するように、既存の経路制御であるDSRの拡張を行う。DSRは実機にも実装されているプロトコルであるため、将来的に実装する場合、最も現実的であると思われる。解決すべき点は、以下の2点である。

1. 現時点で最適な予備経路を取得
2. 複数経路取得時のループ問題

1. のように、常に最適な予備経路を取得するためには、常にルートキャッシュが新しい情報でなければならない。もし、ルートキャッシュの情報が新しければ、それ以降の端末移動量もわずかであると思われるので、リンクが切断された時でもすぐに再接続することができる。ここでは特にストリーミング通信のような、リアルタイム性を要する利用法を考えているため、再接続時間の短縮が必要である。VoIP(Voice over Internet Protocol)を行うと仮定すると、人間が会話可能である限界遅延は0.2sと言われているため、再接続時間はこれ以下でなければならない、従って目標値を0.2sと定

めるものとする。

2.の問題点は、最初の経路探索の際に複数の経路を取得しようとするために起こる問題である。よって本提案では、最初の経路探索では一つの経路だけ取得し、データ通信後、予備の経路を取得するようにする。ここで、予備経路が主経路となるべく関与しないようにするため、データ通信後のRREQ(Route Request)に関しては、主経路を構成するノードがそれを受信した場合、RREQパケットを破棄するようにする。

提案する経路制御はルート探索フェーズ、予備ルート探索フェーズ、データパケット送信フェーズ、ルート切替えフェーズの4フェーズからなる。

ルート探索フェーズ

データ通信を行う要求が発生したとき、送信ノードはデータ通信を行うためのルートを発見するため、ルート探索を行う。一連の動作を以下に示す。

1. 送信ノードはルート探索を開始し、RREQパケットをフラッディングする。
2. RREQパケットを受信した宛先ノードはRREQパケットに付加されているルートに沿ってRREP(Route Reply)をユニキャストする。
3. 送信ノードはRREPパケットに付加されているルートを用いてデータ通信を開始する。その際、宛先ノードDと現在データ通信中であるルート情報を、ルートキャッシュと通信中キャッシュに格納する。

既存のDSRと異なる点は、送信ノードが通常のルートキャッシュとは別に「通信中キャッシュ」を持つことである。通信中キャッシュは、現在データ通信を行っている経路のソースルートを保持している。この通信中キャッシュはリンクの切断や、データ通信が終了した場合には、通信中キャッシュ内のソースルートは削除される。ゆえに、本研究で提案する経路制御では、通信中キャッシュをチェックすることにより、現在データ通信中かどうかの判断ができる。

Multipath DSRなど複数経路を取得するプロトコルと異なり本プロトコルでは、ルート探索のときには単一経路のみ取得するところが特徴である。これはルート探索時に複数経路を取得した場合、RREQパケットのループが発生し、理想的な予備経路を取得できない場合があるからである。本プロトコルでは通常のデータ通信中にルート探索を行うことによって予備経路を取得する。

予備ルート探索フェーズ

本提案経路制御ではデータ通信中、送信ノードは宛先ノードへ向けて定期的にRREQパケットをフラッディ

ングし、予備経路を取得する。このとき、フラッディングされるRREQパケットはルート探索時のRREQパケットに以下のルールを付加する。

通信中キャッシュに格納されているルートを構成するノードがRREQパケットを受信した場合、それを破棄する。

送信ノードがデータ通信中にRREPパケットを受信した場合、それに付加されている情報に基づき、ルートキャッシュを更新する。もし、RREQパケットをフラッディングして一定時間内にRREPパケットを受信できなかった場合は、通信中キャッシュを参照し、現在データ通信中のルート以外の、その宛先へのルート情報をルートキャッシュ内から削除する。このように使用不可能となった予備経路を削除することにより、使用不可能なルートキャッシュを利用して通信の再開を試み、再びルートメンテナンスを行う二度手間を省くことができる。結果として再接続までの時間を短縮することができる。

そして最後に、送信ノードの通信中キャッシュを削除する。ここで削除された直後も、データ通信中であればデータパケット送信後、常に通信中キャッシュが格納されることになり、ここで通信中キャッシュを削除することは、データ通信プロセスが終了したということを検知するためである。

データパケット送信フェーズ

データパケットを送信後、送信ノードはデータパケットを送信したルートを通信用中キャッシュに格納する。予備ルート探索を行った際、通信中キャッシュは逐一削除されるが、データパケットを送信中は通信中キャッシュが存在することになる。送信が終了した場合、予備ルート探索により通信中キャッシュが削除された後、通信中キャッシュに格納されないため、プロトコルは通信が終了したと検知することができる。

ルート切替えフェーズ

DSRでは、リンク切断を認識した中継ノードは、その送信元ノードへ向けてRERRパケットを送信する。本プロトコルでは、RERRパケットを受信した場合、通信中キャッシュを削除した後、ルートキャッシュに格納されている予備ルート情報を用いてデータ通信を開始する。そして、そのルート情報を通信中キャッシュにコピーする。

4. 評価実験と結果

実験環境

ネットワークシミュレータであるGloMoSim(Global Mobile Information Systems Simulation Library)[8]上に提案プロトコルを実装し、シミュレーションをおこなう

ことで特性を評価する。

GloMoSimは、カリフォルニア大学ロサンゼルス校で開発され、無線ネットワークの環境を想定したシミュレータである。GloMoSimは、レイヤ構造をもち、レイヤごとにライブラリを選択できる。評価環境に関するパラメータや、ネットワークプロトコルのパラメータは、設定ファイルに記述する。GloMoSimで選択可能なプロトコルの大半は、UCLAで開発されたC言語に基づくシミュレーション言語であるParsec (Parallel simulation environment for complex systems)[9]で実装されている。また、Javaで実装された視覚的ツールを用いると、シミュレーション環境の確認、シミュレーションの一時停止と再開、および、パケットの伝播状態のアニメーション表示が可能になる。評価結果は、数値データとしてテキスト表示される。現在、GloMoSimをもとにして実装されたシミュレータQualNet[10]が商用化されている。

シミュレーションは大きく分けて三種類のアプローチで行う。シミュレーション1では少数のノードでシミュレーションをおこない、リンク切断から再接続までの時間を計測する。シミュレーション2では100個以上の移動ノードでシミュレーションをおこない、ネットワークポロジの変化に対応できるかどうか、評価を行う。シミュレーション3では複数の通信プロセスが性能に及ぼす影響を評価をする。

また、比較対象の経路制御として、DSRおよび、DSRに複数経路を取得する機能を付加したプロトコル(ほぼMultipath DSRと同じ機能。以下「拡張DSR」)を使用する。

シミュレーション1: 経路切り替え時間

シミュレーション環境として、提案プロトコルがもっとも機能しやすい環境を考える。まず、予備ルートを一つだけ取得できる状況を考える。この予備ルートはデータ通信中も変化はしない。一定の時間が経過した後、主経路のリンクを強制的に切断し、予備ルートと切り替えることによって、リンク切断から再接続し、宛先ノードまでデータパケットが到着するまでの時間を計測する。シミュレーションはそれぞれ10回ずつ行い、その平均を測定数値として採用する。シミュレーション環境を表1に示す。

表1:シミュレーション1の環境

上位プロトコル	UDP
パケットサイズ	512バイト
パケット送信速度	20パケット/秒
予備ルート探索間隔	1秒
シミュレーション時間	60秒

結果を図1に示す。ここでパケットの再転送時間とは、リンクが切断してから経路変更した後、データパケットが宛先ノードまで到達するまでの時間を表すものとする。なお、拡張DSRと本研究での提案プロトコルとは、経路切り替えに要する時間は原理上同じであるため、実験データによる比較は意味がなく、従ってここでは行わない。

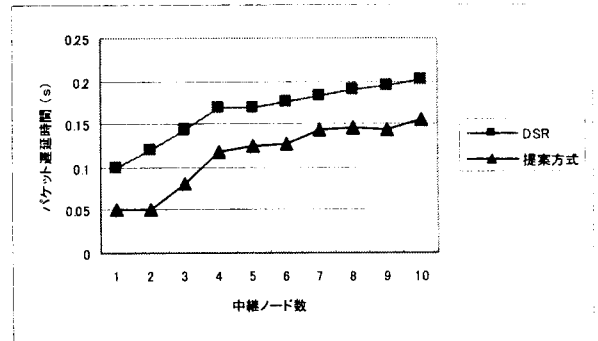


図1:シミュレーション1における測定結果
(予備ルート変化なし)

図1のグラフが示すように、提案方式は既存のDSRに比べて、経路切り替え時のパケット遅延が約0.05s程少ない。提案方式はデータ通信中に予備ルートを確保しているため、このことはリンク切断後のルート探索の時間を短縮したことを示している。

次に予備経路を二つ以上取得できるような環境においてのシミュレーションをおこなう。このシミュレーションでは、ルートキャッシュ内に格納されているルートキャッシュが、ノードの移動などによって使用できなくなった場合、主経路のリンク切断後、再接続からデータパケット到着までの時間の比較した。シミュレーション環境は同様である。

図2に測定結果を示す。このシミュレーションにおいても、既存のDSRよりもルート探索の時間分だけパケット遅延時間を短縮することができている。また、今回はルート探索の際に予備ルートを取得する、拡張DSRとの比較も行った。この拡張DSRはルート探索時のみ予備ルートを取得するため、今回のシミュレーションのように、データ通信中に予備ルートが使用不可能となった場合、ルートメンテナンスを行った後、再ルート探索を行わなければならないため、既存のDSRよりわずかながらパケット遅延時間が大きくなっていることがわかる。

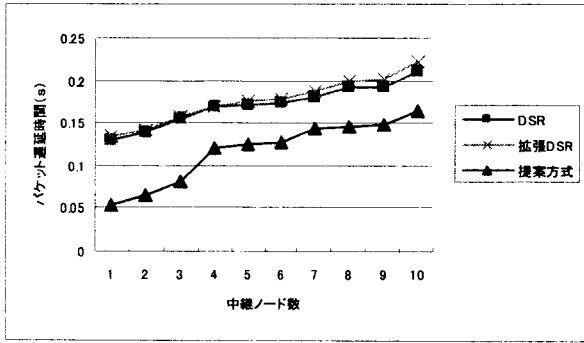


図2:シミュレーション1における測定結果
(予備ルート変化あり)

シミュレーション2:ランダムなノード移動

人が携帯端末をもって移動を行う状況を模して、ノードがランダムに移動する条件でのシミュレーションを行う。表2にシミュレーション環境を示す。

表2:シミュレーション2の環境

上位プロトコル	UDP
フィールド	1000m×1000m
ノード数	100個～
ポーズタイム	0秒
ノードの移動速度	0～5m/s
シミュレーション回数	各10回
シミュレーション時間	200秒

1000m×1000mを(0.0)～(1000.1000)の座標として、(200.200)に送信ノード、(800.800)に受信ノードを配置する。そして、この二つのノード間で通信をおこない、パケットの遅延、到達率を調べた。提案プロトコルについては、データ通信中のRREQパケットの送信間隔を、0.5、1、2、3秒としてそれぞれ計測し、既存のDSRと拡張DSRとの比較を行った。シミュレーションはそれぞれについて10回ずつ行い、その平均値を測定数値として採用した。

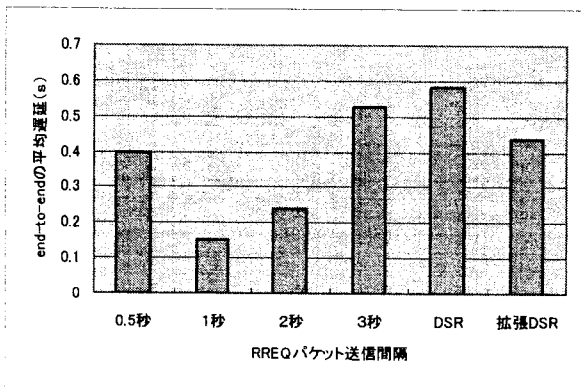


図3:シミュレーション2におけるend-to-endの平均遅延

測定結果を図3に示す。データ通信中のRREQパケットの送信間隔が1秒の提案プロトコルが最も効率が良いことがわかる。また、人間が会話可能である限界遅延値は0.2sであり、本研究ではこの目標値を下回ること成功している。

今回はシミュレーションをそれぞれ10回ずつ行ったが、それぞれについて、平均値遠地が0.2s未満であったシミュレーションの回数をグラフで表したのが図4である。このグラフでも、RREQパケットの送信間隔が1秒のものが最も多くなっていることがわかる。

図5にパケット到達率の測定結果を示す。パケット到達率は、受信ノードが受信することに成功したパケット数を、送信ノードが送信したパケット数で割ることで算出した。

このシミュレーションにおいても、RREQパケット送信間隔が1秒のものが最も良い結果を出している。また、提案内容は総じて既存のDSRよりもパケット到達率が高い。これは、データ通信中にルート探索をおこなっているため、常に最適なルートで通信を行うことが可能であるためであると考えられる。

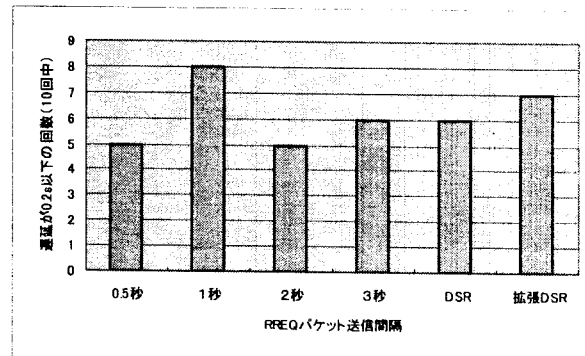


図4:平均遅延が0.2s以下のシミュレーション回数

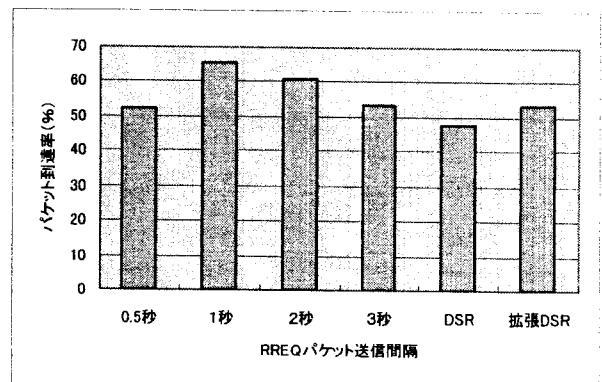


図5:シミュレーション2におけるパケット到達率

シミュレーション3:複数の通信ノード

シミュレーション2ではフィールド上に通信プロセスが一つしか存在していなかった。しかし、複数のノードが

それぞれ通信を行った場合、通信プロセスに互いに影響が出る可能性がある。そこでここではend-to-endの平均遅延とパケット到達率の二点についてそれぞれ測定をおこなった。シミュレーションはそれぞれ10回ずつ行い、その平均値を測定値として採用する。なお、提案内容のプロトコルのデータ通信中RREQパケット送信間隔は1秒とする。

図6に測定結果のグラフを示す。他の通信プロセス数が3を超えた時点で提案方式が他の方式の平均遅延を上回る。これは、提案方式が、データ通信中にRREQパケットのフラッディングをおこなっているため、パケットの衝突が頻繁に起こっているためであると考えられる。

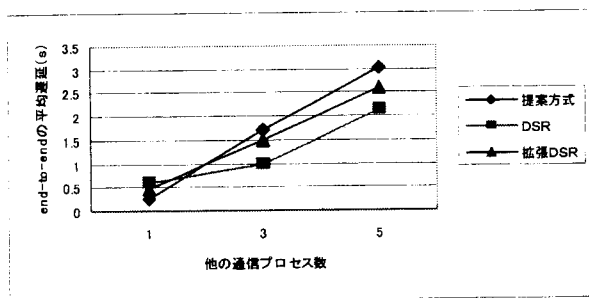


図6:シミュレーション3におけるend-to-endの平均遅延

図7に測定結果のグラフを示す。他の通信プロセスが増えるにつれて、パケット到達率が低下する。このグラフでも、通信プロセス数が3の時点で提案方式が既存のDSR、拡張DSRよりもパケット到達率が低くなるのがわかる。

5. まとめ

本論文では、アドホックネットワークにおいて、冗長な予備経路を維持する経路制御を提案し、それをネットワークシミュレータ上に実装することで特性評価をおこなった。このプロトコルはデータ通信中に定期的なRREQパケットをフラッディングし、常に信頼性の高い予備経路を取得することが可能である。

データ通信に必要なノードのみ存在する条件で、ルート切り替え時の再接続時間についてシミュレーションにて評価を行った。この評価では、再ルート探索を行う時間を短縮でき、既存のDSRや、ルート探索時に予備経路を取得する拡張DSRよりも良い結果となった。また、再接続時間の目標値である0.2sを下回り、VoIP等の応用に適用可能であることを示した。

次に実際の利用状況を模してノードをランダムに移動させてデータ通信を行い、シミュレーションにて評価した。結果、既存のDSR、拡張DSR以上に良い値を測定できた。本提案プロトコルでパケット到達率が向上したのは、定期的にRREQパケットをフラッディングし、

常に最適な経路を維持しているからである。

さらにノードの移動はそのままに、複数の通信プロセスの同時通信が性能に与える影響を評価した。結果、他の通信プロセスが増えれば増えるほど、既存のDSR、拡張DSRよりも機能が低下することがわかった。これはデータ通信中にRREQパケットをフラッディングしているため、パケット同士の衝突が頻繁に起こっているためであると考えられる。

データ通信中のRREQパケットのフラッディング時にパケットの衝突を減少させるように改良を加え、複数の通信プロセスによるスループット低下を防ぐことが必要となると思われる。また、実際の環境における応用の実現や性能の評価も今後の課題である。

参考文献

- [1] MANET, <http://www.ietf.org/html.charters/manet-charter.html>.
- [2] C. E. Perkins and P. Bhagwat, "Highly dynamic destination-sequenced distance-vector routing," Proc. ACM SIGCOMM, London, UK, pp. 234-244, Oct. 1994.
- [3] D. B. Johnson and D. A. Maltz, "Dynamic Source Routing in Ad Hoc Wireless Networks," in book on "Mobile Computing," edited by T. Imielinski and H. Korth, Kluwer Academic Publishers, 1996.
- [4] V. D. Park and M. S. Corson, "A highly adaptive distributed routing algorithm for mobile wireless networks," Proc. IEEE INFOCOM '97, 1997.
- [5] P. Johansson, T. Larsson, N. Hedman, B. Mielczarek and M. Degermark, "Scenario-based performance analysis of routing protocols for mobile ad-hoc networks," Proc. ACM/IEEE MOBICOM '99, pp. 195-206, Aug. 1999.
- [6] A. Nasipuri and S. R. Das, "On-demand multipath routing for mobile ad hoc networks," in Proc. of IEEE International Conference On Computer Communications and Networks (ICCCN 2000), Oct. 2000.
- [7] 溝口和寛, 古庄伸一, 北須賀輝明, 中西恒夫, 福田晃, "再接続時間を短縮したアドホックネットワークルーティングプロトコルの提案と評価," 情報処理学会 DICO2003シンポジウム論文集, Jun. 2003.
- [8] GloMoSim, <http://pcl.cs.ucla.edu/projects/glomosim/>.
- [9] Parsec, <http://pcl.cs.ucla.edu/projects/parsec/>.
- [10] QualNet, <http://www.scalable-networks.com/>.