

複数経路を用いた安定なメッセージ配送のための アドホックルーティングプロトコル

東京電機大学 理工学部 情報システム工学科
長谷部 顕司 梅島 慎吾 桧垣 博章
{namu, shin5, hig}@higlab.k.dendai.ac.jp

アドホックネットワークのためのルーティングプロトコルとして DSR (Dynamic Source Routing Protocol)、AODV (Ad-hoc On-Demand Distance Vector Routing Protocol) などが提案されている。これらのプロトコルでは、データ通信を開始する直前に送信元から送信先までの経路探索を行ない、検出された経路情報のみを管理し、これを用いてデータ通信を行なうオンデマンド型のプロトコルである。また、1度の探索によって検出される経路は1つのみである単一経路検出プロトコルである。しかし、アドホックネットワークでは、コンピュータの移動などによる無線信号の到達範囲の変動やコンピュータの電源断などによってリンクが頻繁に切断される。このため、探索時に単一の経路のみではなく、迂回路となり得る複数の経路を検出する複数経路検出プロトコルが必要である。本論文では、AODV を拡張し、未接続状態のリバースパスを互いに接続することによって複数の経路を検出する MRAODV (Multiple-Route Ad-hoc On-Demand Distance Vector) ルーティングプロトコルを提案する。

Multiple-Route Adhoc Ondemand Routing (MRAODV) Protocol

Kenji Hasebe, Shingo Umeshima and Hiroaki Higaki
{namu, shin5, hig}@higlab.k.dendai.ac.jp
Department of Computers and Systems Engineering
Tokyo Denki University

Several routing protocols, e.g. DSR (Dynamic Source Routing Protocol) and AODV (Ad-hoc On-Demand Distance Vector Routing Protocol), have been proposed for routing data packets in ad-hoc networks. These are on-demand routing protocols. Only when a node requires to transmit data packets, it searches a route to a destination node. In addition, these are single-path protocols which detect only one route. However, in an ad-hoc network, due to mobility of nodes and instability of communication links, multiple-route protocols are required. In this paper, we propose a novel multiple-route protocol MRAODV (Multiple-Route On-Demand Distance Vector Routing Protocol) where separated reverse path fragments are connected to achieve additional routes. According to simulation results, more additional routes are detected in our protocol than MNH protocol.

1 背景と目的

近年、PDA やノート型 PC などのモバイルコンピュータの普及が進んでいる。また、IEEE802.11 [1] や HIPERLAN [2] といった無線 LAN プロトコル技術の研究開発、利用が進み、モバイルコンピュータでのネットワーク利用が期待されている。従来のコンピュータネットワークは、固定ネットワークに接続されたルータのみがメッセージの配送を行なうインフラストラクチャネットワーク (Infrastructure Networks) であった。しかし、固定ネットワークを対象とした従来の構築、管理、運用手法を、災害救助活動やイベント会場などに利用されるコンピュータネットワークのように、移動性、緊急性、一時性を要求される用途に適用することは、その構築コストが大きいため困難である。そこで、固定されたルータのみによってメッセージが配送される従来のネットワークに代わって、すべてのモバイルコンピュータがメッセージの配送を行なう、すなわち、すべてのモバイルコンピュータがルーティング機能を持つネットワークであるアドホックネットワーク (Ad-hoc Networks) への要求が高まっている。アドホックネットワークでは、すべてのコンピュータがメッセージ配送を行ない、かつ、これらのコンピュータが移動することから、既存の有線ネットワークを対象としたルーティングプロトコルとは異なるルーティングプロトコルが必要とされる。有線ネットワークでは、距離ベクトルに基づく RIP (Routing

Information Protocol) [4] やリンクステートに基づく OSPF (Open Shortest Path First) [9] といった、各ルータが定期的に経路情報を交換し、ネットワーク全体のトポロジーを管理するプロアクティブ (proactive) 型の手法が採られている。しかし、無線ネットワークでは、通信帯域幅が十分ではなく、コンピュータの移動によってリンクが頻繁に切断されることから、従来のルーティングプロトコルをアドホックネットワークに適用するのは困難である。

そこで注目されているのがリアクティブ (reactive) 型あるいはオンデマンド (On-Demand) 型のルーティングプロトコルである。各ルータのルーティングテーブルに格納された経路情報を定期的に交換し、自身のルーティングテーブルの経路情報を更新するのではなく、データ通信を開始するときに送信元から送信先までの経路探索を行なう。そして、コンピュータは、検出した経路のみを管理し、コンピュータ間の定期的な経路情報の交換を必要としない手法である。オンデマンド型のプロトコルとして、DSR [3]、LSR [13]、C-LSR [14]、AODV [11] などが提案されている。

これらは、Message Diffusion Protocol [8] を無線 LAN 環境に適用したフラッディングという手法を用いている。無線 LAN に利用される無線通信メディアの多くはブロードキャストベースであるため、あるコンピュータが送信した無線信号は、その到達範囲内にあるすべ

てのコンピュータが受信することができる。あるコンピュータがメッセージ m をブロードキャストし、それを受信したすべてのコンピュータが同様に m をブロードキャストする。これを繰り返すことによって、マルチホップで到達可能なすべてのコンピュータに m を配送することが可能である。これがフラッディングである。

AODV [11] では、送信元 S からルート要求メッセージ RREQ(Route Request) がフラッディングされると、中間のコンピュータ M_i は最初に受信した RREQ を送信したコンピュータを上流としたリバースパスを設定し、RREQ を再ブロードキャストする。RREQ が送信先 D に到達すると、 D はルート応答メッセージ RREP(Route Reply) をリバースパスに沿ってユニキャストする。RREP を受信した中間のコンピュータ M_i は、この RREP を送信したコンピュータを下流としたフォワードパスをルーティングテーブルに登録し、設定したリバースパスに沿って RREP をユニキャストする。RREQ を受信したが RREP を受信しなかった中間コンピュータは、タイムアウトによって設定したリバースパスを解除する。RREP が S に到達すると、 S から D に至る経路が構築される (図 1)。すなわち、 S および経路上にあるすべての中間のコンピュータ M_i のルーティングテーブルにフォワードパスが設定され、これに沿ってメッセージを転送することによって、 S から D までの配送が可能となる。

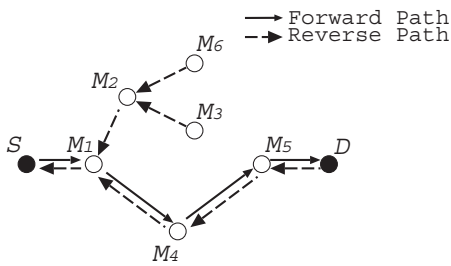


図 1: AODV

AODV [11]、DSR [3] はいずれも単一経路検出のルーティングプロトコルである。検出された送信元 S から送信先 D への経路を用いてデータ通信を行なっているときに、その経路上にあるコンピュータの移動やバッテリー容量切れによる電源断によって経路上にあるいずれかのリンクが切断された場合、再度経路探索を行なわなければならない。経路探索では、ルート要求メッセージをフラッディングするため、衝突、競合の発生頻度が高く [12]、経路上にはないコンピュータもメッセージをブロードキャストするため、コストが大きい。そこで、1度の経路探索で複数の経路を検出する複数経路検出ルーティングプロトコルへの要求が高まっている。これまで、複数経路検出プロトコルとして、MultipathDSR [10]、SMR [7]、AODV-BR [6]、MNH [5] などが提案されている。

MultipathDSR と SMR は、ソースルーティングを行なう DSR を拡張したプロトコルである。送信元 S からルート要求メッセージ RREQ がフラッディングされ、送信先 D に到達すると、 D は到達した複数の RREQ に格納された経路の中から重複部分のない経路を選別する。これらを格納したルート応答メッセージ RREP を S に送信することで、 D への複数の経路を S が得ることが

できる。

一方、AODV-BR、MNH では、AODV を拡張して複数の経路を構築する。AODV を複数経路探索プロトコルに拡張するために AODV-BR では、無線メディアがブロードキャストベースであることを利用する。ここで、AODV によって検出される経路をプライマリルートと呼ぶことにする。プライマリルートに含まれない中間コンピュータも RREP を受信する。リンクの切断を検出した中間コンピュータがメッセージを無線信号到達範囲内にブロードキャストし、このメッセージを RREP を受信したコンピュータが受信することによって、迂回路への切り替えを行うことができる。ただし、この方法では、プライマリルートから 1 ホップ外れた経路への切り替えしか行なうことができない。

MNH は、RREQ のフラッディングを AODV と同様の方法で行なう。中間コンピュータは、2 回目以降に受信した RREQ に対して、1 回目と同様にリバースパスを設定する。ただし、RREQ をブロードキャストせずに破棄する。RREP は複数設定されたリバースパスに沿って配送される。また、中間のコンピュータが複数回 RREP を受信する場合は、受信した RREP の送信コンピュータへのフォワードパスを設定し、2 回目以降に受信した RREP を破棄する。これにより複数の経路が構築される。しかし、MNH の手法では複数の経路を構築できるケースは稀である。図 1 において、 $S \rightarrow M_1 \rightarrow M_4 \rightarrow M_5 \rightarrow D$ というプライマリルートに対して、 $S \rightarrow M_1 \rightarrow M_2 \rightarrow M_3 \rightarrow M_5 \rightarrow D$ という迂回路を構築できるのは、 M_5 が M_4 からの RREQ を受信してから RREQ をブロードキャストする (これは M_3 によって受信される) までの間に M_3 が RREQ をブロードキャストした場合のみである。もし、 M_3 のブロードキャストよりも M_5 のブロードキャストの方が先に行なわれたならば、リバースパスは図 2 のように設定され、複数の経路を構築することができない。

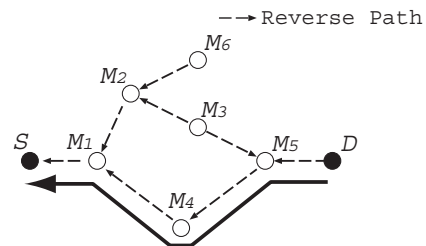


図 2: MNH におけるリバースパス問題

以上のことから、本論文では図 2 のような状況でも複数の経路を構築することが可能な MRAODV (Multiple Route Ad-hoc On-Demand Distance Vector) ルーティングプロトコルを提案する。

なお、無線ネットワークでは、電波の回折、干渉などによる無線信号の到達範囲の変動が起こるため、有線ネットワークのように通信が常に双方向で行なわれるとは限らない。一方からのみ送信可能な、片方向通信が行なわれる場合がある。DSR、LBSR は片方向通信路を経路として利用可能であるルーティングプロトコルである。本論文では、すべての通信路は双方向であると仮定する。

2 MRAODV ルーティングプロトコル

2.1 AODV の複数経路検出への拡張

AODV は、送信元コンピュータ S から RREQ をフラッディングすることによって、 S を最上流として中間コンピュータの上流、下流の関係を暫定的に決定する。このとき、各中間コンピュータは唯一の上流コンピュータを持ち、この上流コンピュータに対してリバースパスを設定する。したがって、リバースパスのうち、連続した1組のリバースパス群だけが S から送信先コンピュータ D に到達する。つまり、 D から S までの1つのリバースパス (メインリバースパス) と、メインリバースパスに接続し、途中まで延びた複数のリバースパス (サブリバースパス) が構築される (図 1)。

D から S までのリバースパスに沿って RREP を配送し、中間コンピュータがルーティングテーブルに D までのメッセージの配送先が RREP の送信コンピュータであることを設定することによってフォワードパス (メインリバースパス) を作成する。このとき RREP を受信しない中間のコンピュータは、タイムアウトによって自身が設定したサブリバースパスを削除する。AODV が単一経路検出プロトコルであるのは、作成したリバースパスのうち、1つのみが送信先に接続するためである。そこで、サブリバースパスを互いに接続することで複数経路検出プロトコルへ拡張することが可能である。

しかし、サブリバースパスを接続するためには上流、下流の区別を壊すことなく接続しなければならない。図 3 のように異なるサブリバースパスのコンピュータ間 (M_2, M_3) を接続する場合、送信元から遠いリバースパス $M_3 \rightarrow M_5$ は $S \rightarrow M_1 \rightarrow M_2 \rightarrow M_3 \rightarrow M_5 \rightarrow D$ という迂回パスを構築するために、上流、下流の関係を反転させなければならない。

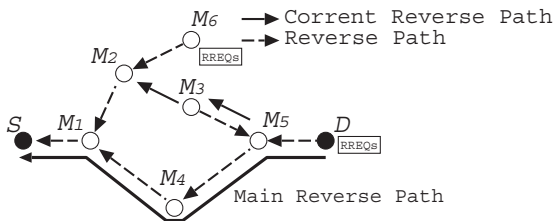


図 3: リバースパスの接続と反転

そこで、リバースパス $M_6 \rightarrow M_2 \rightarrow M_1$ 、 $M_3 \rightarrow M_5$ は、それぞれメインリバースパス上の1つのコンピュータに接続していることに着目する。メインリバースパスは上流、下流の関係が確定したパスであり、それに接続する $M_6 \rightarrow M_2 \rightarrow M_1$ 、 $M_3 \rightarrow M_5$ のようなメインリバースパスの枝となるサブリバースパスは、接続点となるコンピュータ (M_1 と M_5) の S からのホップ数 (枝番号) を基準として上流、下流の区別がなされると考えられる。そこで、MRAODV では確定されたパスとの接続点を求め、それぞれの枝番号を比較することでパスの接続を行なう。

2.2 枝番号による上流、下流の判別

MRAODV では、RREP をメインリバースパスのみでなく、サブリバースパスにも配送する。検出済みのメインリバースパスとの接続点となるコンピュータの S

からのホップ数を枝番号とするため、MRAODV では RREP をメインリバースパスに沿って配送する際に枝番号を割り当てる。

RREQ のフラッディングの際にホップ数をカウントすることにより、 S から D までのメインリバースパスに沿った経路のホップ数を知ることができる。これを RREP に付加し、メインリバースパスに沿って配送しながら (図 3)1 ずつデクリメントすることで、メインリバースパスに含まれるコンピュータに枝番号が与えられる。メインリバースパスに含まれるコンピュータは RREP を必ず下流コンピュータのうちの1つから受信し、それを唯一の上流コンピュータと他の下流コンピュータに送信する。この下流コンピュータはサブリバースパスに含まれる。このように、サブリバースパスに含まれるコンピュータは上流コンピュータから RREP を受け取ることを利用して、自身がメインリバースパスではなくサブリバースパスに含まれることを認識する。RREP を上流コンピュータから受信したコンピュータは、RREP に格納されたホップ数を自身の枝番号として保存する。

例えば図 4 の場合、メインリバースパス $D \rightarrow M_4 \rightarrow M_3 \rightarrow M_2 \rightarrow M_1 \rightarrow D$ に対し、コンピュータ M_4 で接続しているリバースパス $M_9 \rightarrow M_4$ 上のコンピュータは枝番号 4、コンピュータ M_1 で接続しているリバースパス $M_8 \rightarrow M_6 \rightarrow M_5 \rightarrow M_1$ 、 $M_7 \rightarrow M_5 \rightarrow M_1$ 上のコンピュータは枝番号 1 を持つ。

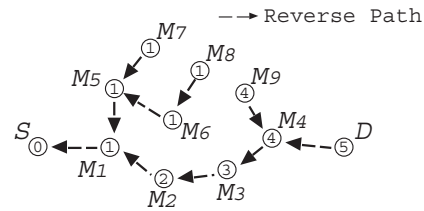


図 4: 同一枝番号であるリバースパス

コンピュータ M_7 、 M_8 、 M_9 はサブリバースパスの終端コンピュータである。 M_8 と M_9 は直接通信可能であるが、異なるサブリバースパスに含まれている。 M_8 、 M_9 の枝番号を比較すると M_8 の方が小さいことから M_8 を含むサブリバースパス $M_8 \rightarrow M_6 \rightarrow M_5 \rightarrow M_1$ が M_9 を含むサブリバースパス $M_9 \rightarrow M_4$ に対して上流であると判別される。

一方、 M_7 、 M_8 に注目すると、これらも互いに直接通信可能であるが、枝番号が共に 1 であり、上流、下流の区別ができない。しかし、 $S \rightarrow M_1 \rightarrow M_5 \rightarrow M_7 \rightarrow M_8 \rightarrow M_9 \rightarrow M_4 \rightarrow D$ という迂回パスを構築することが可能であることから M_7 と M_8 を接続することが望ましい。 M_8 と M_9 の接続により上流、下流の区別がされた $M_8 \rightarrow M_6 \rightarrow M_5 \rightarrow M_1$ と $M_9 \rightarrow M_4$ を接続することで新しく検出されたパスに含まれるコンピュータでは、新たに上流、下流関係が定められる。新しく検出された経路 $S \rightarrow M_1 \rightarrow M_5 \rightarrow M_6 \rightarrow M_8 \rightarrow M_9 \rightarrow M_4 \rightarrow D$ はコンピュータ M_1 、 M_4 を接続点としてメインリバースパスに接することから、 M_1 、 M_4 から再番号割り当てメッセージ RNUM を送信することで M_5 、 M_6 、 M_8 、 M_9 のホップ数とこれらに接続する枝 (サブリバースパ

ス)の枝番号を割り当てる(図5)。

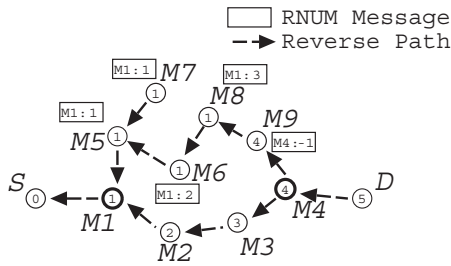


図5: 枝番号再割り当て

再番号割り当てを行なうことで、接続できなかったサブリバースパスは新しく検出された経路に含まれる1つのコンピュータに接続するサブリバースパスとなる。これらのサブリバースパスのいくつかは、新しい枝番号を用いてメインリバースパスに直接接続するサブリバースパスと同じように上流、下流の区別をすることが可能となり、互いに接続される。この手続きを再帰的に適用することにより多数の迂回経路を検出することができる。

2.3 近隣テーブル

無線メディアがブロードキャストベースであることから、RREQ、RREPなど経路探索に用いるメッセージは、無線信号の到達範囲内にあるすべてのコンピュータに受信される。各コンピュータはこれらの制御メッセージを受信することによって、周囲のコンピュータとの上流、下流関係を把握することができる。

MRAODVでは各コンピュータが持つ枝番号を比較してサブリバースパスを接続することから、各コンピュータが接続可能な近隣コンピュータの情報を持つことが必要となる。各コンピュータが近隣コンピュータとの上流、下流関係を把握するためには、制御メッセージを送信するコンピュータの上流アドレスをメッセージに格納すればよい。これにより近隣コンピュータの情報を交換するための制御メッセージを用いることなく近隣のコンピュータの情報を保存することができる。

近隣コンピュータの情報は、近隣テーブルに保存され、RREQ、RREPなどの制御メッセージを受信することにより更新される。枝番号比較のため、近隣テーブルには以下の情報を保存する。

- 近隣コンピュータのアドレス
- 自身との関係(上流、下流、その他、未定義)
- 枝番号

自身との関係は制御メッセージに含まれる上流コンピュータアドレスから、自身と同じパスを構成する、「上流」、「下流」、他のサブリバースパスに所属している「その他」のいずれかを判別し、保存する。

3 複数経路探索プロトコル

[RREQの送受信]

ルート要求メッセージRREQには、以下の情報が格納されている。

RREQ(送信元アドレス、送信先アドレス、上流アドレス、ホップカウント)

(送信元コンピュータ S)

- S は、 D への経路 $R_{S \rightarrow D}$ が自身のルーティングテーブルに存在しているかを確認する。
- $R_{S \rightarrow D}$ が存在する場合は、この経路を使用する。
- $R_{S \rightarrow D}$ が存在しない場合、 S は、上流アドレスフィールドに自身のIPアドレスを格納したルート要求メッセージRREQを無線信号到達範囲内のすべてのモバイルコンピュータ M_i へブロードキャストする。

(中間コンピュータ M_i)

- M_i が最初のRREQを受信した場合(RREQの送信先アドレスが自身のアドレスと異なる場合)、RREQを送信したコンピュータにリバースパスを設定し、そのコンピュータを上流コンピュータとして近隣テーブルに登録する。RREQの上流アドレスフィールドに、受信したRREQを送信したコンピュータのアドレスを格納し、ホップカウントフィールドをインクリメントして、自身の無線信号到達範囲内に存在するすべてのモバイルコンピュータにこのRREQをブロードキャストする。
- M_i が2回目以降のRREQを受信した場合、そのRREQの上流アドレスフィールドが、自身のアドレスであるかを確認する。自身のアドレスであるならば、RREQを送信したコンピュータが自身の下流のコンピュータであることを近隣テーブルに登録する。
- 自身のアドレスでないならば、送信したコンピュータは自身の周囲に存在するコンピュータ(異なるサブリバースパスに含まれる)として登録し、そのRREQメッセージを破棄する。

(送信先コンピュータ D)

- D が最初のRREQを受信した場合、自身あてであるならば、近隣テーブルにRREQを送信したコンピュータを上流として登録する。RREQ内のホップカウントフィールドをインクリメントした値を送信元 S からのホップ数とし、ルート応答メッセージRREPを生成し、リバースパスを設定したコンピュータに対してブロードキャストする。

[RREPの送受信]

ルート応答メッセージRREPには、以下の情報が格納されている。

RREP(送信元アドレス、送信先アドレス、上流アドレス、ホップカウント)

(送信先コンピュータ D)

- RREQを受信した D は、上流アドレスフィールドを近隣テーブルに登録されている上流コンピュータのアドレス、ホップカウントフィールドを S からのホップ数としたルート応答メッセージRREPをブロードキャストする。

(中間コンピュータ M_i)

- RREPを受信した M_i は、RREP内の上流アドレスフィールドに自身のアドレスが格納されているかを確認する。
- 自身のアドレスが格納されている場合、自身がメインリバースパス上に存在することを認識し、RREPを送信したコンピュータに対してフォワードパスを設定する。RREPの上流アドレスフィールドに自

身の上流コンピュータのアドレスを格納し、ホップカウントフィールドをデクリメントして、再ブロードキャストする。

- 自身のアドレスが格納されていない場合、RREPを送信したコンピュータが、自身の上流コンピュータであるかを確認する。
- 自身の上流コンピュータからのメッセージである場合には、RREP内のホップカウント、送信者アドレスを自身の枝番号として近隣テーブルに保存する。その後、再ブロードキャストする。
- 上流コンピュータ以外からのメッセージである場合には、RREP内のホップカウント、送信者アドレスを、そのコンピュータの枝番号として近隣テーブルに保存し、RREPを破棄する。

[枝番号の比較]

(枝刈り)

- 近隣テーブルに上流コンピュータのみが登録されている場合、そのコンピュータに対するリバースパスを削除し、ルート削除メッセージ RDEL をブロードキャストする。

(サブリバースパスの接続)

- 近隣テーブルに下流コンピュータが登録されておらず、その他のコンピュータのみが登録されている場合、その他のコンピュータの枝番号と、自身の枝番号を比較する。
- 自身の枝番号が大きい場合、相手 (その他のコンピュータ) に対し、リバースパスを設定する。自身の上流として登録されていたコンピュータに対し、自身のアドレスとシーケンスナンバを格納した下流接続メッセージ RREV を送信する。その後、自身の上流コンピュータとして相手を登録し、既存の上流コンピュータとして登録されていたコンピュータを下流コンピュータとして登録変更する。
- 自身の枝番号が小さい場合、相手 (その他のコンピュータ) に対し、フォワードパスを設定する。自身の上流として登録されているコンピュータに対し、自身のアドレスとシーケンスナンバを格納した上流接続メッセージ RFOR を送信する。その後、相手を自身の下流コンピュータとして登録する。
- 近隣テーブルに上流、下流コンピュータが登録されており、その他のコンピュータが登録されている場合、その他のコンピュータの枝番号と自身の枝番号を比較する。
- 自身の枝番号が大きい場合、相手 (その他のコンピュータ) に対し、リバースパスを設定し、相手を上流コンピュータとして登録する。自身のホップ数を 0 とし、自身のアドレスを含めたシーケンスナンバを格納した再番号割り当てメッセージ RNUM をブロードキャストする。
- 自身の枝番号が小さい場合、相手 (その他のコンピュータ) に対し、フォワードパスを設定し、相手を自身の下流コンピュータとして登録する。自身のホップ数を 0 とし、自身のアドレスとシーケンスナンバを格納した再番号割り当てメッセージ RNUM をブロードキャストする。

[RFOR の送受信]

上流接続メッセージ RFOR には、以下の情報が格納されている。

RFOR (送信元アドレス、送信先アドレス、上流アドレス、シーケンスナンバ、ホップカウント)

- 自身が検出済みの経路上に存在するならば、RFOR を送信したコンピュータにフォワードパスを設定し、RFOR を破棄する。その後、自身を経路としたサブリバースパスに再番号割り当てメッセージ RNUM を送信する。
- そうでなければ、RFOR を送信したコンピュータにフォワードパスを設定し、自身の上流コンピュータに RFOR を送信する。

[RREV の送受信]

下流接続メッセージ RREV には、以下の情報が格納されている。

RREV (送信元アドレス、送信先アドレス、上流アドレス、シーケンスナンバ、ホップカウント)

- 自身が検出済みの経路上に存在するならば、RREV を送信したコンピュータにリバースパスを設定し、RREV を破棄する。その後、RREV を送信したコンピュータを上流コンピュータとして登録変更し、自身を経路としたサブリバースパスに再番号割り当てメッセージ RNUM を送信する。
- そうでなければ、RREV を送信したコンピュータにリバースパスを設定する。自身の上流として登録されていたコンピュータに対するリバースパスを破棄し、RREV をそのコンピュータに送信する。その後、自身の上流として登録されていたコンピュータを下流として、新規にリバースパスを設定したコンピュータを上流として登録変更する。

[RNUM の送受信]

- 自身の枝番号を 0 とし、受信した RFOR または RREV に格納されたシーケンスナンバを設定した再番号割り当てメッセージ RNUM を無線信号到達範囲内にブロードキャストする。
- 同じシーケンスナンバを持つ RNUM を以前に受信したかを確認する。すでに同じシーケンスナンバを持つ RNUM を受信したならば、この RNUM を破棄する。
- RNUM 内のシーケンスナンバが、自身が以前送信した RFOR または RREV に格納されたシーケンスナンバと一致するか確認する。
- シーケンスナンバが一致した場合、ホップカウントをインクリメントまたはデクリメントし、自身の枝番号として近隣テーブルに追加登録する。その後、再ブロードキャストする。
- シーケンスナンバが一致しなかった場合、RNUM を送信したコンピュータが上流コンピュータであるかを確認する。
- RNUM を送信したコンピュータが上流コンピュータである場合、ホップカウントを自身の枝番号として近隣テーブルに追加登録し、RNUM を再ブロードキャストする。
- それ以外の場合、RNUM を送信したコンピュータの枝番号として近隣テーブルに追加登録し、RNUM を破棄する。

[RDEL の送受信]

- RDEL を送信したコンピュータが下流コンピュータであるかを確認する。
- 下流コンピュータである場合、フォワードパスが登録されているならば削除し、そのコンピュータを未定義のコンピュータとして登録変更する。その後自身の下流コンピュータが他に存在するかを確認し、枝番号比較を行なう。

4 性能評価

MRAODV と MNH が検出する経路数を以下の条件でシミュレーションにより計算した。

- エリアサイズ: 200(m) × 200(m)
- 信号伝達距離: 20(m)
- コンピュータの分布: 一様 (ランダム)
- シミュレーション回数: 500 回

コンピュータの数を 100, 120, 140, 160, 180, 200 とした場合の平均検出経路数を図 6 に、コンピュータの数を 160 とした場合の探索開始からの経過時間と検出経路数との関係を図 7 に示す。

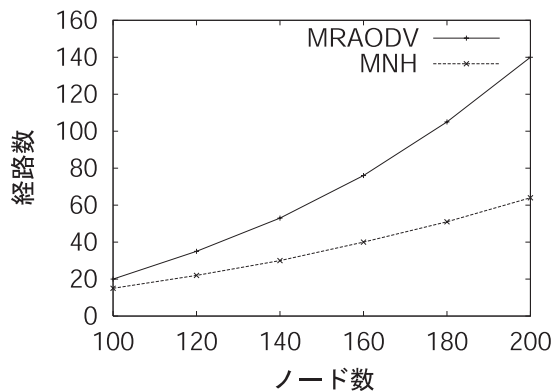


図 6: 検出経路数 (対ノード数)

図 6 に示されるように、MNH、MRAODV とともにコンピュータの分布が密になるにつれ、検出された経路数が増加しているが、その数を比較すると MRAODV の方がより多くの経路を検出している。これは、MNH ではリバースパスを正しく構築できなかった場合、それを使用することができなくなるが、MRAODV では、上流、下流の区別を壊すことなく迂回経路の構築に使用することが可能であるためである。しかし、MRAODV では比較可能な枝番号を受け取るまでサブリバースパスの接続処理を行わないため、図 7 に示すように複数の経路検出に要する時間が長い。しかし、最初の経路を検出するまでの時間、すなわち、データ送信の開始までにかかる時間は同じである。

5 まとめ

本論文では、AODV を拡張し、サブリバースパスを相互接続することにより、複数の経路を構築する MRAODV を提案した。MRAODV では RREQ、RREP などの制御メッセージから上流、下流の関係を保存した近隣テーブルを作成し、接続可能なコンピュータが順次、再接続処理を行なうことで複数の経路を構築する。今後は、シミュレーションによりコンピュータの移動と有効

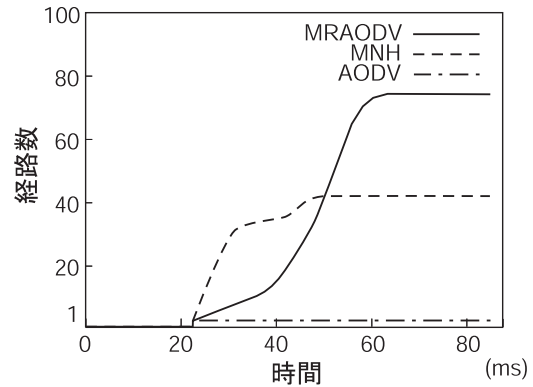


図 7: 検出経路数の時間変化

な経路数との関係性を求め、従来プロトコルとの比較を行ない、その有効性を検証する。

参考文献

- [1] "Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications," Standard IEEE 802.11 (1997).
- [2] "Radio Equipment and Systems (RES); HIPERLAN," ETSI Functional Specifications (1995).
- [3] David, B., David, A., Hu, Y.C., Jorjeta, G. and Jetcheva, "The Dynamic Source Routing Protocol for Mobile Ad Hoc Networks," Internet Draft, draft-ietf-manet-dsr-04.txt (2000).
- [4] Hedrick, C., "Routing Information Protocol," RFC 1058 (1988).
- [5] Jiang, M.H. and Jan, R.H., "An Efficient Multiple Paths Routing Protocol for Ad-hoc Networks," Proc. of the 15th International Conference on Information Networking, pp. 544-549 (2001).
- [6] Lee, S.J. and Gerla, M., "AODV-BR: Backup Routing in Ad hoc Networks," Proc. of IEEE Wireless Communications and Networking Conference, pp. 1311-1316 (2000).
- [7] Lee, S.J. and Gerla, M., "Split Multipath Routing with Maximally Disjoint Paths in Ad hoc Networks," Proc. of IEEE International Conference on Communications, pp. 3201-3205 (2001).
- [8] Moses, Y. and Roth, G., "On reliable message diffusion," Proc. of the 8th ACM Symposium on Principles of Distributed Computing, pp. 119-128 (1989).
- [9] Moy, J., "Open Shortest Path First specification," RFC 1131 (1989).
- [10] Nasipuri, A. and Das, D.S., "On-Demand Multipath Routing for Mobile Ad Hoc Networks," Proc. of IEEE 8th International Conference on Computer Communications and Networks, pp. 64-70 (1999).
- [11] Perkins, C.E. and Royer, E.M., "Ad-hoc On-Demand Distance Vector Routing," Proc. of IEEE 2nd Workshop on Mobile Computing Systems and Applications, pp. 90-100 (1999).
- [12] Tseng, Y., Ni, S. and Shih, E., "Adaptive Approaches to Relieving Broadcast Storms in a Wireless Multihop Mobile Ad Hoc Network," Proceedings on the 21st International Conference on Distributed Computing Systems, pp. 481-488 (2001).
- [13] 佐川, 松垣, "アドホックネットワークにおけるループ型ルーティングプロトコル," 第9回マルチメディアと分散処理ワークショップ論文集, IPSJ Symposium Series, No. 13, pp. 157-162 (2001).
- [14] 佐川, 松垣, "ループ経路接合によるアドホックルーティングプロトコル (C-LBSR)," 情報処理学会第64回全国大会 (2001).