

MRAODVにおける経路切替手法

東京電機大学 理工学部 情報システム工学科

初田 紗矢香 森田 一樹 桧垣 博章

E-mail: {saya, kazuki, 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、メッセージ配送を開始する直前に送信元から送信先までの経路構築を行ない、構築された経路情報のみを管理し、これを用いてメッセージ配送を行なうオンデマンド型のプロトコルである。また、構築される経路が1つのみである単一経路構築プロトコルである。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、単一の経路のみではなく、迂回経路となり得る複数の経路を構築する複数経路構築プロトコルが必要である。本論文では、AODVを拡張し、未接続状態のバックワードリンクを互いに接続することによって複数の経路を構築する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。提案プロトコルは、従来の複数経路探索プロトコルよりも多くの経路を検出し、経路再探索までの時間を延長している。また、経路切替プロトコルは、切替前の経路と共通の移動コンピュータを多く含む経路へと切替えることによって、切替えに要する時間を短縮し、切替えによって紛失されるアプリケーションメッセージを削減する。

Route Switching Protocol in MRAODV (Multiple-Route Adhoc Ondemand Distance Vector)

Sayaka Hatsuta, Kazuki Morita and Hiroaki Higaki

Department of Computers and Systems Engineering

Tokyo Denki University

E-mail: {saya, kazuki, hig}@higlab.k.dendai.ac.jp

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、ノート型PCやPDAなどの移動コンピュータの普及にともない、IEEE802.11 [1]やHIPERLAN [2]などの無線LAN技術の研究開発が行なわれている。従来のコンピュータネットワークは、有線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であった。移動コンピュータは、有線ネットワークに接続されたアクセスポイントを経由して、他の移動コンピュータとの通信を可能としている。しかし、インフラストラクチャネットワークの構築と運用に要する金銭的、時間的コストは大きい。そのため、イベント会場などで利用者の要求を充足するために一時的に構築されるネットワークや、工事現場、災害現場、戦場のように要求の変化に応じて構成が変化する必要のあるネットワーク、航空機や乗用車などに搭載されたコンピュータを相互接続するネットワークなどに適用することは困難である。そこで、有線ネットワークとそれに接続するアクセスポイントを用いず、無線通信する移動コンピュータがメッセージをマルチホップで配送するモバイルアドホックネットワークが注目されて

いる。

モバイルアドホックネットワークでは、メッセージを配送する移動コンピュータが移動することから、送信元移動コンピュータが通信要求を発生した時点で送信先移動コンピュータまでの有効な配送経路を探索、検出するオンデマンド型のルーティングプロトコルが用いられている。DSR [4]とAODV [14]は、経路探索要求メッセージ *Rreq* の送信元移動コンピュータからのフラッディングを用いるプロトコルである。ここでは、送信先移動コンピュータが最初に受信した *Rreq* の配送経路が唯一の検出経路となる。しかし、移動コンピュータが時々刻々その位置を変化させることや、有限容量のバッテリーで駆動される移動コンピュータは、残電力量が0となることで通信できなくなるなどにより、通信途中であっても検出した経路が無効になることがある。経路探索時に単一の経路のみを検出するプロトコルでは、経路の無効化とともに、フラッディングをともなう経路探索を再度行なわなければならない。この通信オー

バヘッドを抑制する手法として、経路全体を無効化するのではなく、切断した通信リンクを含む経路の一部のみを無効化し、迂回経路に切替える経路修復がある。AODV-BR [7] では、経路切断時に、切断を検出した移動コンピュータがルート変更要求メッセージをブロードキャストすることで、迂回経路を探索する。ただし、検出できる迂回経路は、プライマリルートから1ホップ外れた経路のみであるため、検出できる迂回経路は少ない。このように、経路再検出、経路修復の手法では、多くの制御メッセージを要する、経路の切替えに要する時間が長く、アプリケーションメッセージの配送が一時停止するなどの問題がある。そこで、あらかじめ複数の経路を検出しておき、通信リンクの切断によって配送に使用している経路が無効化したならば、別の有効な経路へと切替える複数経路検出プロトコルを用いることが有効である。本論文では、AODVを拡張し、未接続状態の枝経路(ブランチルート)を互いに接続することで複数の経路を構築するMRAODV(Multiple-Route Ad-hoc On-Demand Distance Vector)プロトコルを提案する。また、配送に使用している経路の切断を検出した場合に、別の有効な経路へと切替えるプロトコルについて述べる。最後に、シミュレーションによって既存のルーティングプロトコルと性能を比較し、提案プロトコルの有効性を明らかにする。

2 従来手法

これまでに提案された多くのアドホックルーティングプロトコルは、Message Diffusion Protocol [10] を無線LAN環境に適用したフラッディング [3] 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、ある移動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移動コンピュータが受信することができる。ある移動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移動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、マルチホップ配送で到達可能なすべての移動コンピュータが m を受信することが可能である。これがフラッディングである。

AODV [14] では、移動コンピュータ M_s から経路探索要求メッセージ $Rreq$ (Route Request) がフラッディングされると、移動コンピュータ M_i は最初に受信した $Rreq$ を送信した隣接移動コンピュータを上流移動コンピュータとした逆リンク(バックワードリンク)を設定し、 $Rreq$ を再ブロードキャストする。 $Rreq$ が移動コンピュータ M_d に到達すると、 M_d は経路探索応答メッセージ $Rrep$ (Route Reply) をバックワードリンクに沿ってユニキャストする。 $Rrep$ を受信した移動コンピュータ M_i は、この $Rrep$ を送信した移動コンピュータを下流(次ホップ)移動コンピュータとした順リンク(フォワードリンク)を設定する。すなわち、この隣接移動コンピュータを M_d への配送に対する次ホップ移動コンピュータとしてルーティングテーブルに登録する。そして、設定したバックワードリンクに沿って $Rrep$ をユニキャストする。 $Rreq$ を受信したが $Rrep$ を受信しなかった移動コンピュータは、タイムアウトによって設定したバックワードリンクを解除する。 $Rrep$ が M_s に到達すると、 M_s から M_d に至る経路 $R_{M_s \rightarrow M_d}$ が検出される(図1)。すなわち、経路 $R_{M_s \rightarrow M_d}$ 上にあるすべての移動コンピュータ M_i からのフォワードリンクの接続先移動コンピュータが M_d を送信先移動コンピュータとするメッセージの次ホップとしてルーティングテーブルに登録され、これに従ってアプリケーションメッセージを転送することに

よって、 M_s から M_d までの配送が可能となる。

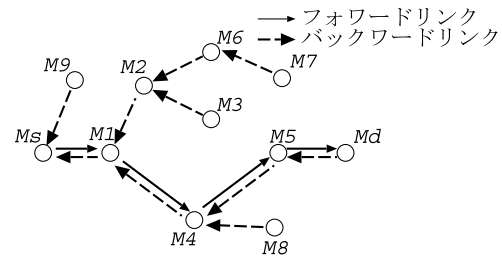


図1: AODV

AODV [14]、DSR [4]、LBSR [16]、C-LBSR [17] 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 M_s から M_d への経路を用いてメッセージ配送を行なっているときに、その経路上にある中継移動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、経路探索要求メッセージ $Rreq$ をフラッディングするため、衝突、競合が発生する [15]。経路上にはない移動コンピュータも経路探索要求メッセージをブロードキャストするため、要する通信コストが大きい。そこで、複数経路構築ルーティングプロトコルへの要求が高まっている。

これまでに提案された複数経路検出プロトコルには、アドホックネットワークのためのマルチパスルーティング [19]、AOMDV [9]、MNH [6] などがある。いずれもAODVを拡張したネクストホップ型のプロトコルである。マルチパスルーティングは、送信元移動コンピュータを送信先移動コンピュータ以外に共通の移動コンピュータを含まない複数の経路を検出するプロトコルである。経路上の通信リンクが切断したならば、送信元移動コンピュータが次ホップの移動コンピュータを切替えることで、アプリケーションメッセージの配送に用いる経路を切替えることができる。しかし、この方法では、検出される経路数が少ないために、再経路探索までの時間を十分に延長することができない。また、経路上の移動コンピュータが次ホップの移動コンピュータとの間の通信リンクの切断を検出した場合には、これを送信元移動コンピュータに伝えなければならない(図2)。この間に送信されたアプリケーションメッセージは、送信先移動コンピュータに配送されることなく破棄される。

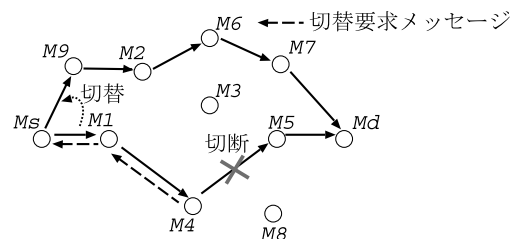


図2: マルチパスルーティングによる経路切替

一方、AOMDV、MNHは、共通の移動コンピュータを含むことを許す複数経路検出プロトコルである。マルチパスルーティングと比較すると、より多くの経路を検出できることから、再経路探索までの時間を延長することができる。経路探索は、AODVと同様に $Rreq$ のフラッディングによってなされる。ただし、AODVで

は各移動コンピュータが最初に受信した $Rreq$ を送信した隣接移動コンピュータのみに対してバックワードリンクを設定する(上流移動コンピュータとする)のに対して、2回目以降に受信した $Rreq$ を送信した隣接移動コンピュータに対してもバックワードリンクを設定する。ただし、 $Rreq$ をブロードキャストせずに破棄する。 $Rrep$ は複数設定されたバックワードリンクに沿って配送される。また、中間移動コンピュータが複数回 $Rreq$ を受信する場合は、受信した $Rreq$ を送信した移動コンピュータへのフォワードリンクを設定し、2回目以降に受信した $Rreq$ を破棄する。これによって複数経路の検出が可能であるとしている。しかし、[18]で指摘されているように、移動コンピュータが最初の $Rreq$ を受信してからそれをブロードキャスト送信するまでの間に他の隣接移動コンピュータから $Rreq$ を受信することは稀であり、複数経路を検出することは難しい。

図1において、 $M_s \rightarrow M_1 \rightarrow M_4 \rightarrow M_5 \rightarrow M_d$ というプライマリルートに対して、 $M_s \rightarrow M_1 \rightarrow M_2 \rightarrow M_3 \rightarrow M_5 \rightarrow M_d$ という迂回経路を検出できるのは、 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する場合のみである¹。もし、 M_3 よりも先に M_5 が $Rreq$ をブロードキャストするならば、バックワードリンクは図3のように設定され、複数の経路を検出することができない。

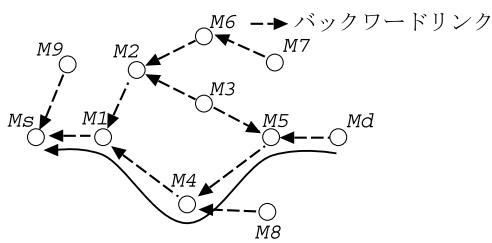


図 3: MNH における複数経路検出失敗例

以上のことから、本論文では図3のような状況でも複数の経路を検出することが可能な 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、C-LBSR は片方向通信リンク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、すべての通信リンクは双方向であると仮定する。

3 MRAODV ルーティングプロトコ

MRAODVでは、AODV同様、経路探索要求メッセージ ($Rreq$) を送信元移動コンピュータからフラッディングする。この $Rreq$ が、マルチホップ配送により送信先移動コンピュータに到達することによって、アプリケーションメッセージの配送経路の1つ(プライマリルート)を検出する。このとき、各移動コンピュータが最初に受信した $Rreq$ の送信者を親と定めることによって、送信元移動コンピュータを根とするスパニングツリーを構成することができる。このスパニングツリーのそれぞ

¹ RTS/CTS を用いて排他的にブロードキャストが行なわれているものと仮定する。

れの枝は、プライマリルート上の移動コンピュータのいずれかを含む。異なる枝に属し、かつ隣接する移動コンピュータを接続することで、複数の経路を検出する。このとき、それぞれの枝のプライマリルートに含まれる移動コンピュータの位置に基づいて、枝の上流、下流関係を決定し、各通信リンクの配送方向を決定することによって、MNH と AOMDV を用いたときに発生する問題点を解決している。

3.1 AODV の複数経路構築への拡張

AODVは、送信元移動コンピュータ M_s から $Rreq$ をフラッディングすることによって、 M_s を最上流として中間移動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、各移動コンピュータは唯一の上流移動コンピュータを持ち、この上流移動コンピュータに対してバックワードリンクを設定する。つまり、 M_s を根とし、マルチホップで到達可能なすべての移動コンピュータを含むスパニングツリーが構成される。このとき、バックワードリンクのうち、連続した1組のバックワードリンク列だけが M_s から送信先コンピュータ M_d に到達する。すなわち、 M_d から M_s までの1つの幹経路(プライマリバックワードルート)と、プライマリバックワードルートに接続し、途中まで延びた複数の枝経路(ブランチバックワードルート)が構築される(図1)。

M_d から M_s までのプライマリバックワードルートに沿って $Rrep$ を配送し、中継移動コンピュータがルーティングテーブルに M_d を送信先移動コンピュータとするメッセージの転送先が $Rrep$ の送信移動コンピュータであることを設定することによって、フォワードリンクを作成する。このとき、 $Rrep$ を受信しない移動コンピュータは、タイムアウトによって自身が設定したバックワードリンクを削除する。AODVが単一経路検出プロトコルであるのは、作成したバックワードルートのうち、1つのみが送信先 M_d に接続するためである。そこで、ブランチバックワードルートを互いに接続することで複数経路構築プロトコルへ拡張することが可能である。

しかし、ブランチバックワードルートを相互接続するときには、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4のように異なるブランチバックワードルートに含まれる移動コンピュータ間 (M_2, M_3) を接続する場合、送信元移動コンピュータから遠いバックワードリンク $M_3 \rightarrow M_5$ は $M_s \rightarrow M_1 \rightarrow M_2 \rightarrow M_3 \rightarrow M_5 \rightarrow M_d$ という迂回経路を検出するために、上流、下流の関係を反転させなければならない。

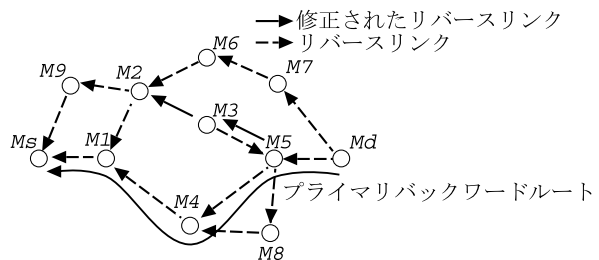


図 4: バックワードリンクの接続と反転

そこで、ブランチバックワードルート $M_6 \rightarrow M_2 \rightarrow M_1$ 、 $M_3 \rightarrow M_5$ は、それぞれプライマリバックワードルート上の1つの移動コンピュータに接続していることに着目する。プライマリバックワードルートは、上流、下流の関係が確定したリンク列であり、それに接続する

$M_6 \rightarrow M_2 \rightarrow M_1$, $M_3 \rightarrow M_5$ のようなプライマリバックワードルートの枝経路となるブランチバックワードルートは、接続点となる移動コンピュータ (M_1 と M_5) の M_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 では各ブランチバックワードルートと上流、下流の関係が確定している移動コンピュータからなるバックワードルートとの接続点を求め、それぞれの枝番号を比較することでブランチバックワードルートの相互接続を行なう。

3.2 枝番号による上流、下流の判別

MRAODV では、 $Rrep$ をプライマリバックワードルートのみでなく、ブランチバックワードルートにも配送する。検出済みのプライマリバックワードルートとの接続点となる移動コンピュータの M_s からのホップ数を枝番号とするため、MRAODV では $Rrep$ をプライマリバックワードルート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、 M_s から M_d までのプライマリバックワードルート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、プライマリバックワードルートに沿って配送しながら 1 ずつデクリメントすることで、プライマリバックワードルートに含まれる移動コンピュータを接点とするブランチバックワードルートの枝番号が与えられる。プライマリバックワードルートに含まれる移動コンピュータは、 $Rrep$ を必ず下流移動コンピュータのうちの一つから受信し、それを唯一の上流移動コンピュータと他の下流移動コンピュータに送信する。この下流移動コンピュータは、ブランチバックワードルートに含まれる。このように、ブランチバックワードルートに含まれる移動コンピュータは上流移動コンピュータから $Rrep$ を受け取ることを利用して、自身がプライマリバックワードルートではなく、ブランチバックワードルートに含まれることを認識する。 $Rrep$ を上流移動コンピュータから受信した移動コンピュータは、 $Rrep$ に格納されたホップ数を自身から分岐するブランチバックワードルートの枝番号として保存する。

例えば図 5 の場合、プライマリバックワードルート $M_d \rightarrow M_4 \rightarrow M_3 \rightarrow M_2 \rightarrow M_1 \rightarrow M_s$ に対し、移動コンピュータ M_4 で接続しているブランチバックワードルート $M_9 \rightarrow M_4$ 上の移動コンピュータには枝番号 4、移動コンピュータ M_1 で接続しているバックワードリンク $M_8 \rightarrow M_6 \rightarrow M_5 \rightarrow M_1$ 、 $M_7 \rightarrow M_5 \rightarrow M_1$ 上の移動コンピュータは枝番号 1 を持つ。

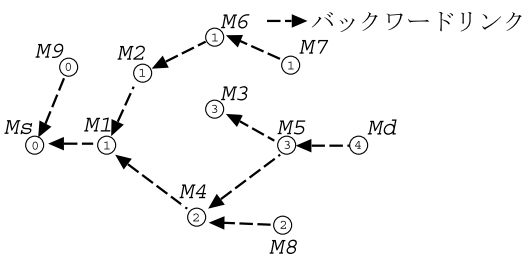


図 5: ブランチバックワードルートへの枝番号の付与

移動コンピュータ M_7 , M_8 , M_9 はブランチバックワードルートの末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、異なるブランチバックワードルートに含まれている。 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 を含むブランチバックワードルート $M_8 \rightarrow M_6 \rightarrow M_5 \rightarrow M_1$ が M_9 を含むブランチバックワードルート $M_9 \rightarrow M_4$ 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、 $M_s \rightarrow M_1 \rightarrow M_5 \rightarrow M_7 \rightarrow M_8 \rightarrow M_9 \rightarrow M_4 \rightarrow M_d$ という迂回経路を検出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ルートに含まれる移動コンピュータでは、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 $M_s \rightarrow M_1 \rightarrow M_5 \rightarrow M_6 \rightarrow M_8 \rightarrow M_9 \rightarrow M_4 \rightarrow M_d$ は移動コンピュータ M_1 , M_4 を接続点としてプライマリバックワードルートに接することから、 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 (ブランチバックワードルート) の枝番号を割り当てる (図 6)。

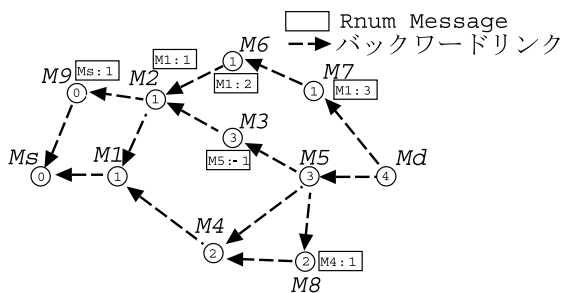


図 6: 枝番号再割り当て

再番号割り当てを行なうことで、接続できなかったブランチバックワードルートは新しく検出された経路に含まれる 1 つの移動コンピュータに接続するブランチバックワードルートとなる。これらのブランチバックワードルートのいくつかは、新しい枝番号を用いてプライマリバックワードルートに直接接続するブランチバックワードルートと同じように上流、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。

3.3 隣接テーブル

無線通信メディアがブロードキャストベースであることから、 $Rreq$, $Rrep$ などの経路検出に用いる制御メッセージは、無線信号到達範囲内にあるすべての移動コンピュータに受信される。各移動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、周囲の移動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 では各移動コンピュータが持つ枝番号を比較してブランチバックワードルートを接続することから、各移動コンピュータが接続可能な隣接移動コンピュータ (互いの無線信号到達範囲内にあり、直接メッセージ交換可能な移動コンピュータ) の情報を持つことが必要となる。各移動コンピュータが隣接移動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移動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。これによって、近隣移動コンピュータの情報を交換するための追加制御メッセージを用いることなく、隣接移動コンピュータの情報を獲得、保存す

ることができる。

隣接移動コンピュータの情報は、隣接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ごとに更新される。枝番号比較のため、隣接テーブルには以下の情報を保存する。

- 隣接移動コンピュータのアドレス
- 自身との関係(上流、下流、並行、未定)
- 枝番号

自身との関係は、制御メッセージに含まれる上流移動コンピュータアドレスを用いて決定することができる。自身と同じ経路に含まれている場合は、送信元移動コンピュータに近い方が「上流」、遠い方が「下流」である。「並行」とは、自身が含まれているものとは異なる他のブランチルートに含まれていることを示す。これらのいずれでもない場合は「未定」とする。

4 複数経路構築プロトコル

4.1 スパニングツリーの構成とプライマリルートの検出

[*Rreq*の送受信]

経路探索要求メッセージ *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)

(送信元移動コンピュータ M_s)

- M_s は、 M_d への経路 $R_{M_s \rightarrow M_d}$ 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。
- $R_{M_s \rightarrow M_d}$ が存在する場合は、この経路を使用する。
- $R_{M_s \rightarrow M_d}$ が存在しない場合、 M_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経路探索要求メッセージ *Rreq* を無線信号到達範囲内のすべての移動コンピュータ M_i へブロードキャストする。

(中継移動コンピュータ 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* を送信した移動コンピュータにバックワードリンクを設定し、そのコンピュータを上流コンピュータとして隣接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* を送信した移動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移動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* を送信した移動コンピュータが自身の下流移動コンピュータであることを隣接テーブルに登録する。
- 自身のアドレスでないならば、送信した移動コンピュータは自身の周囲に存在する移動コンピュータ(異なるブランチバックワードルートに含まれる)として登録し、その *Rreq* メッセージを破棄する。

(送信先移動コンピュータ M_d)

- M_d が最初の *Rreq* を受信した場合、自身あてであるならば、隣接テーブルに *Rreq* を送信した移動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移動コンピュータ M_s からのホップ数とし、経路探索応答メッセージ *Rrep* を生成し、バックワードリンクを設定した移動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]

経路探索応答メッセージ *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)

(送信先移動コンピュータ M_d)

- *Rreq* を受信した M_d は、上流アドレスフィールドを隣接テーブルに登録されている上流移動コンピュータのアドレス、ホップカウントフィールドを M_s からのホップ数とした経路探索応答メッセージ *Rrep* をブロードキャストする。

(中継移動コンピュータ 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、自身がプライマリバックワードルート上に存在することを認識し、*Rrep* を送信した移動コンピュータに対してフォワードリンクを設定する。*Rrep* の上流アドレスフィールドに自身の上流移動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* を送信した移動コンピュータが、自身の上流移動コンピュータであるかを確認する。
- 自身の上流移動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隣接テーブルに保存する。その後、再ブロードキャストする。
- 上流移動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、その移動コンピュータの枝番号として隣接テーブルに保存し、*Rrep* を破棄する。

4.2 ブランチルートの接続による迂回経路の構築

[枝番号の比較]

(枝刈り)

- 隣接テーブルに上流移動コンピュータのみが登録されている場合、その移動コンピュータに対するバックワードリンクを削除し、ルート削除メッセージ *Rdel* をブロードキャストする。

(ブランチバックワードルートの接続)

- 隣接テーブルに下流移動コンピュータが登録されておらず、その他の移動コンピュータのみが登録されている場合、その他の移動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(その他の移動コンピュータ)に対し、バックワードリンクを設定する。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、自身の上流移動コンピュータとして相手を登録し、既存の上流移動コンピュータとして登録されていた移動コンピュータを下流移動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(その他の移動コンピュータ)に対し、フォワードリンクを設定する。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、相手を自身の下流移動コンピュータとして登録する。
- 隣接テーブルに上流、下流移動コンピュータが登録されており、その他の移動コンピュータが登録されている場合、その他の移動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(その他の移動コンピュータ)に対し、バックワードリンクを設定し、相手を上流移動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(その他の移動コンピュータ)に対し、フォワードリンクを設定し、相手を自身の下流移動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* を送信した移動コンピュータにフォワードリンクを設定し、*Rfor* を破棄する。その後、自身を経路としたブランチバックワードルート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* を送信した移動コンピュータにフォワードリンクを設定し、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* を送信した移動コンピュータにバックワードリンクを設定し、*Rrev* を破棄する。その後、*Rrev* を送信した移動コンピュータを上流移動コンピュータとして登録変更し、自身を経路としたブランチバックワードルート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* を送信した移動コンピュータにバックワードリンクを設定する。自身の上流として登録されていた移動コンピュータに対するバックワードリンクを破棄し、*Rrev* をその移動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移動コンピュータを下流として、新規にバックワードリンクを設定した移動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隣接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* を送信した移動コンピュータが上流移動コンピュータであるかを確認する。
- Rnum* を送信した移動コンピュータが上流移動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隣接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* を送信した移動コンピュータの枝番号として隣接テーブルに追加登録し、*Rnum* を破棄する。

[*Rdel* の送受信]

- Rdel* を送信した移動コンピュータが下流移動コンピュータであるかを確認する。
- 下流移動コンピュータである場合、フォワードリンクが登録されているならば削除し、その移動コンピュータを未定の移動コンピュータとして登録変更する。その後自身の下流移動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。

5 経路切替手法

MRAODV は、ネクストホップ型のルーティングプロトコルであることから、検出された複数の経路のいずれかに含まれる各移動コンピュータは、送信先移動コンピュータへの配送のための次ホップ移動コンピュータ(下流移動コンピュータ)を1つ以上持つ。特に、現在用いられているアプリケーションメッセージの配送経路上の移動コンピュータは、次ホップ移動コンピュータの1つをアプリケーションメッセージの転送先移動コンピュータとしている。これを、アクティブ次ホップ移動コンピュータとよぶ。一方、各移動コンピュータは、自身を次ホップとする1つ以上の移動コンピュータ(上流移動コンピュータ)を持つ。各移動コンピュータは、次ホップ移動コンピュータとの間の通信リンクが切断した場合には、これを次ホップ移動コンピュータから取り除く。このとき、これがアクティブ次ホップ移動コンピュータであるならば、他の次ホップ移動コンピュータのうちの1つをアクティブ次ホップ移動コンピュータとすることで経路を切替える。このとき、切替えられたアクティブ次ホップ移動コンピュータには、経路切替要求メッセージ(*Rswitch*)を送信する。*Rswitch*を受信した移動コンピュータにアクティブ次ホップ移動コンピュータが存在しない場合には、次ホップ移動コンピュータのうちの1つをアクティブ次ホップ移動コンピュータとする。一方、次ホップ移動コンピュータとの間の通信リンクの切断検出によって次ホップ移動コンピュータを持たなくなった場合には、経路無効化要求メッセージ(*Rinvalid*)をすべての上流移動コンピュータに送信する。*Rinvalid*を受信した移動コンピュータは、*Rinvalid*を送信した移動コンピュータを次ホップ移動コンピュータから取り除く。特に、アクティブ次ホップ移動コンピュータから *Rinvalid*

を受信した場合には、上記切断検出時と同様の切替え処理を要する。

逆に、上流移動コンピュータとの間の通信リンクの切断を検出したならば、これを上流移動コンピュータから取り除く。この結果、上流移動コンピュータが1つも存在しなくなったならば、*Rinvalid* をすべての次ホップ移動コンピュータに送信する。この *Rinvalid* を受信した移動コンピュータは、この *Rinvalid* を送信した移動コンピュータを上流移動コンピュータから取り除く。

これらの手続きによって、送信元移動コンピュータの次ホップ移動コンピュータが存在しなくなったならば、フラディングを用いた経路探索を再度行なう。

図7において、プライマリルート上の移動コンピュータ M_4 において、アクティブ次ホップ移動コンピュータである M_5 への通信リンクが切断したとする。このとき、 M_4 は他の次ホップ移動コンピュータ M_8 を持っているため、 M_4 は M_8 に *Rswitch* を送信し、 M_8 をアクティブ次ホップ移動コンピュータとすることで経路を切替える。また、その経路切替が行なわれた後に、経路

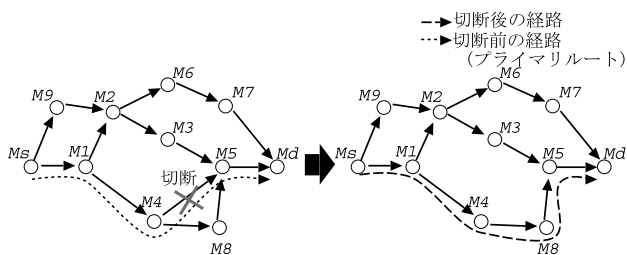


図 7: MRAODV における経路切替 (1)

として使用されている移動コンピュータ M_5 からアクティブ次ホップ移動コンピュータ M_d への通信リンクが切断されたとする (図8)。このとき、 M_5 は次ホップ移動コンピュータをすべて失ってしまったため、自身の上流移動コンピュータである M_3 と M_8 に *Rinvalid* を送信する。*Rinvalid* を受信した M_3 と M_8 では、 M_5 を次ホップ移動コンピュータから取り除く。ここで、 M_5 は M_8 のアクティブ次ホップ移動コンピュータであるため、 M_5 同様の経路切替処理を行なう必要がある。さらに、 M_5 は M_8 の唯一の下流移動コンピュータであることから、 M_8 でも M_d へ至る経路上にある次ホップ移動コンピュータへの通信リンクがすべて失われているため、上流移動コンピュータ M_4 に *Rinvalid* を送信する。このようにして、他の次ホップ移動コンピュータを持つ M_1 まで *Rinvalid* 送信される。 M_1 においてアクティブ次ホップ移動コンピュータを M_2 とし、 M_2 に *Rswitch* を送信することで、切替え処理は完了する。

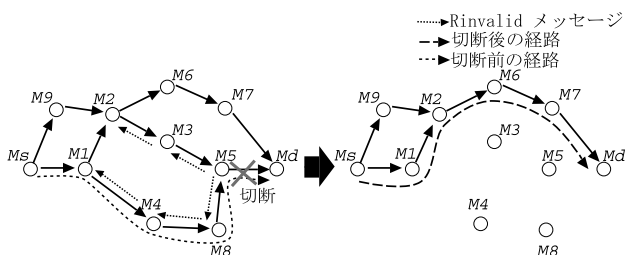


図 8: MRAODV における経路切替 (2)

6 性能評価

MRAODV と MNH、AODV-BR が構築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。ただし、AODV-BR では、リンクの切断が検出されてから近接経路の1ホップ探索が行なわれるため、このような切替が可能ならすべての経路を数え上げることとする。

- エリアサイズ: 500(m) × 500(m)
- 信号伝達距離: 100(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: 100回

移動コンピュータの数を 20, 40, 60, 80, 100 とした場合の平均構築経路数を図9に、移動コンピュータの数を 50、移動速度を 0~5km/h の一様分布とし、移動方向と速度を一定時間で変化させた場合の構築開始からの経過時間と構築経路数との関係を図10に示す。

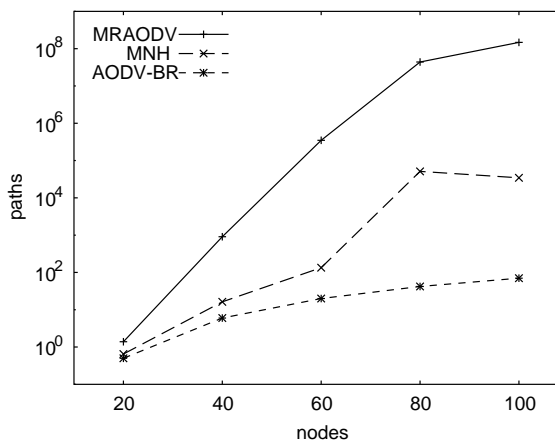


図 9: 構築経路数 (対ノード数)

図9に示されるように、いずれのルーティングプロトコルにおいてもコンピュータの分布が密になるにつれ、構築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構築している。これは、MNH ではバックワードリンク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ブランチバックワードルートの接続処理を行なわないため、図10に示すように複数の経路構築に要する時間が長い。しかし、最初の経路を構築するまでの時間、すなわち、メッセージ配送の開始までにかかる時間は図11に示すように他のプロトコルと同じである。

また、図10に示すように、モバイルコンピュータの移動速度を 0~5km/h とした場合、MRAODV は、AODV、MNH と比較すると、それぞれ 120[s]、80[s] 長く送信元から送信先まで通信が可能であった。これは、MNH では正しく構築できなかった経路を MRAODV では使用することができるため通信可能時間を延長できることを示している。また、経路構築に用いる制御メッセージは約 100ms で収束することから、モバイルコンピュータの多少の移動にかかわらず、多くの経路を構築することができ、多くの経路を構築することで、送信元から送信先まで安定したメッセージ配送を行うことが可能であるといえる。

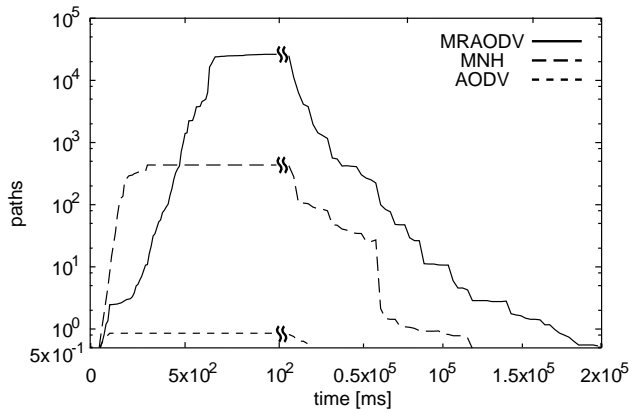


図 10: 構築経路数の時間変化

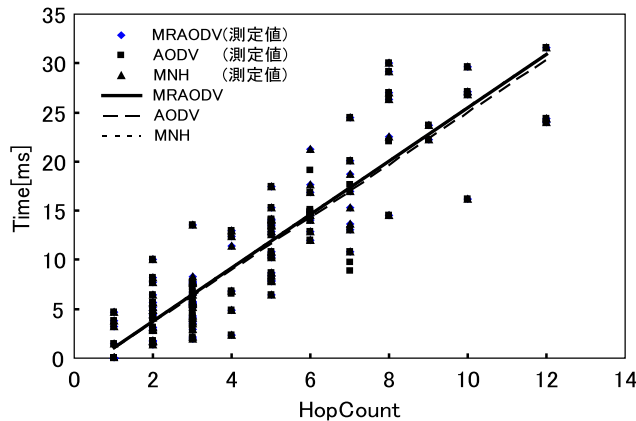


図 11: プライマリルート構築時間

7 まとめ

本論文では、AODV を拡張し、ブランチバックワードルート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、本プロトコルを非同期プロトコルへ拡張し、経路探索経過時間と構築経路数の関係の評価を行ない、有効性を示す。

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] Corson, M.S. and Ephremides, A., “A Distributed Routing Algorithm for Mobile Wireless Networks,” ACM Journal of Wireless Networks, vol. 1, No. 1, pp. 61–81 (1995).
- [4] David, B., David, A. and Hu, Y.C., “The Dynamic Source Routing Protocol,” Internet Draft, draft-ietf-manet-dsr-09.txt (2003).
- [5] Hedrick, C., “Routing Information Protocol,” RFC 1058 (1988).

- [6] 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).
- [7] 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).
- [8] 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).
- [9] Marina, K.M. and Das, R.S., “On-demand Multipath Distance Vector Routing in Ad Hoc Networks,” Proc. of IEEE International Conference on Network Protocols (ICNP), pp. 14–23 (2001).
- [10] Moses, Y. and Roth, G., “On reliable message diffusion,” Proc. of the 8th ACM Symposium on Principles of Distributed Computing, pp. 119–128 (1989).
- [11] Moy, J., “Open Shortest Path First specification,” RFC 1131 (1989).
- [12] 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).
- [13] Perkins, C.E. and Bhagwat, P., “Highly Dynamic Destination-Sequenced Distance-Vector Routing (DSDV) for Mobile Computers,” ACM SIGCOMM’ 94, pp. 234–244 (1994).
- [14] 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).
- [15] 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).
- [16] 佐川, 桧垣, “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, pp. 157–162 (2001).
- [17] 佐川, 桧垣, “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 回全国大会論文集, No. 3, pp. 317–318 (2002).
- [18] 長谷部, 梅島, 桧垣, “複数経路を用いた安定な通信のための AODV アドホックルーティングプロトコルの拡張,” 電子情報通信学会情報処理ネットワーク研究会, 信学技報, Vol. 102, No. 441, pp. 45–50 (2002).
- [19] 茂木, 吉原, 堀内, “アドホックネットワークのためのマルチパス・ルーティングの提案,” 電子情報通信学会情報処理ネットワーク研究会, 信学技報, Vol. 102, No. 441, pp. 51–56 (2002).