

# MANETのための複数経路を用いたルーティングプロトコルMR-LBSRとその性能評価

東京電機大学 理工学部 情報システム工学科  
浅野 知倫 桧垣 博章  
E-mail: {tom, 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)、LBSR (Loop-Based Source Routing Protocol) などが提案されている。これらのプロトコルは、アプリケーションメッセージの配送を開始する直前に送信元移動コンピュータから送信先移動コンピュータ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、検出される経路が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到達範囲の変動やコンピュータの電源断などによって通信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も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、LBSRを拡張し、片方向リンクを含む場合でも、ループ経路探索によって送信元から送信先を通り送信元へと戻る複数のループ経路を検出し、アプリケーションメッセージの配送に用いられる経路が切断された場合には他の経路へと切替えるとともに、他に切断された経路があれば、それを以降の切替え対象から取り除くMR-LBSR (Multiple-Route LBSR) プロトコルを提案する。

## MR-LBSR: Multiple-Route Loop-Based Source Routing Protocol

Tomonori Asano and Hiroaki Higaki  
Department of Computers and Systems Engineering  
Tokyo Denki University  
E-mail: {tom, hig}@higlab.k.dendai.ac.jp

Several routing protocols, e.g. DSR (Dynamic Source Routing Protocol), AODV (Ad-hoc On-Demand Distance Vector Routing Protocol) and LBSR (Loop-Based Source Routing Protocol), have been proposed for routing data packets in ad-hoc networks. These are on-demand routing protocols. Only when a mobile computer requires to transmit data packets, it searches a route to a destination one. In addition, these are single-path protocols which detect only one route. However, in an ad-hoc network, due to mobility of computers and instability of communication links, multiple-route protocols are required. In this paper, we propose a novel multiple-route protocol MR-LBSR (Multiple-Route Loop-Based Source Routing Protocol) by which multiple loops including a source mobile computer and a destination one are detected. If a disconnected link is detected in the currently using route, packet transmission is switched to another route and invalid routes are removed from a set of available routes.

### 1 背景と目的

近年、PDAやノート型PCなどの移動コンピュータの普及が進んでいる。また、IEEE802.11 [1]やHIPER-LAN [3]、Bluetooth [2]といった無線LANプロトコル技術の研究開発、使用が進み、移動コンピュータを含むネットワークの活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、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 (Infrastructure Networks) 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の無線信号到達範囲内に位置するときのみ、これを經由して他のコンピュータと通信することができる。しかし、インフラストラクチャ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使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に困難である。そこで、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

べての移動コンピュータがメッセージの配送を行なう、すなわち、すべての移動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) [6]やリンクステートに基づくOSPF (Open Shortest Path First) [12]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を管理するプロアクティブ (Proactive) 型の手法が採られている。DSDV [15]は、この手法をアドホックネットワークに適用したルーティングプロトコルである。しかし、無線ネットワークでは通信帯域幅が十分ではないため、通信要求の有無に関わらず経路情報を交換するためのメッセージが必要と

なるこれら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、メッセージ配送を開始するときに送信元から送信先までの経路を探索する。各移動コンピュータは、検出した経路のみを管理し、移動コンピュータ間の定期的な経路情報の交換は必要とされない。オンデマンド型のプロトコルとして、AODV [16]、DSR [5]、TORA [14]、LBSR [18]、C-LBSR [19] などが提案されている。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やバッテリー切れによる電源断、電磁波ノイズなどによって移動コンピュータ間の通信リンクが頻繁に切断される。このため、単一経路検出のルーティングプロトコルでは、経路上の通信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。そこで、経路探索時に迂回経路となり得る複数の経路を検出するプロトコルが必要とされている。

本論文では、LBSR を拡張し、片方向リンクを含む場合でも、ループ経路探索によって送信元から送信先を通り送信元へと戻る複数のループ経路を検出し、経路が切断された場合には他の経路へと切替えるとともに、他に切断された経路があれば、それを以降の切替え対象から取り除く MR-LBSR (Multiple-Route LBSR) プロトコルを提案する。

## 2 従来手法

### 2.1 LBSR

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 [11] を無線 LAN 環境に適用したフラッディング [4] という手法を用いている。無線 LAN に使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、マルチホップで到達可能なすべての移動コンピュータに  $m$  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。

無線信号には、光や電波といった電磁波が用いられる。無線信号を用いたアドホックネットワークでは、多くの有線ネットワークのように各隣接移動コンピュータ間が必ずしも双方向リンクで接続されるとは限らない。ところが、現在のアドホックネットワークにおけるルーティングプロトコルの多くは隣接移動コンピュータ間のリンクが双方向であることを仮定している。本論文の議論の対象である LBSR [18] は、アドホックネットワーク内に片方向リンクが存在するものと仮定したオンデマンド型ルーティングプロトコルである。双方向接続のみでなく、片方向接続もメッセージの配送経路に用いることによって、経路検出の成功率を高めることができる。片方向接続もメッセージ配送経路として使用するアドホックルーティングプロトコルには DSR [5]、ULSR [20]、

RODA [8] がある。なお、LBSR は、検出した配送経路を送信元が獲得し、その経路をアプリケーションメッセージのヘッダに格納し、各移動コンピュータがこれを用いてメッセージを配送するソースルーティングプロトコルである。

隣接移動コンピュータ間のすべてのリンクが双方向であるアドホックネットワークにおいては、送信元移動コンピュータ  $S$  から送信先移動コンピュータ  $D$  までの経路  $R_{S \rightarrow D}$  を検出すれば、同時に経路  $R_{D \rightarrow S}$  を検出したことになる。 $R_{S \rightarrow D}$  を反転させて得られる  $R_{D \rightarrow S}$  を用いることで、 $D$  で検出された  $R_{S \rightarrow D}$  を  $S$  が得ることができる。しかし、片方向リンクを含むアドホックネットワークにおいては、 $R_{D \rightarrow S}$  は  $R_{S \rightarrow D}$  の反転とはなっていない。 $R_{S \rightarrow D}$  を  $S$  が得るためには、 $R_{S \rightarrow D}$  と  $R_{D \rightarrow S}$  が必要である。DSR では、これらを 2 つの独立なフラッディングによって検出している。これに対して、LBSR では、 $R_{S \rightarrow D}$  と  $R_{D \rightarrow S}$  を連結して得られるループ経路を探索している。特に、求める経路である  $S$  から  $D$  を通り  $S$  に戻るループ経路の探索中に検出される  $S$  から  $D$  を通らずに  $S$  に戻るループ経路をユニキャストで使用するにより、ブロードキャストメッセージ数を削減し、経路探索に要するプロトコルオーバーヘッドを低減している。

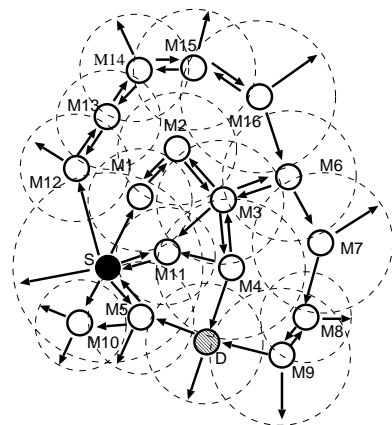


図 1: LBSR におけるループ経路探索

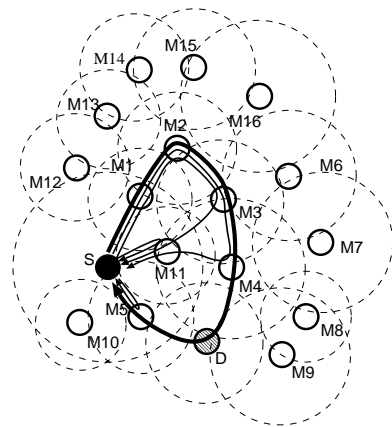


図 2: LBSR におけるループ経路構築

LBSR では、経路探索時に 2 種類のメッセージ  $Lreq$  と  $Lconf$  を用いる。 $Lreq$  は、 $S$  から  $S$  へ戻るループ経

路を探索するためのメッセージであり、経路上にある移動コンピュータのアドレスのシーケンスが格納されている。 $Lreq$  は、 $S$  を起点とするフラッディングによって配送される。 $S$  が  $Lreq$  を受信することによって、 $S$  を含むひとつのループ経路が検出される。 $Lconf$  には、検出された  $S$  から  $S$  に戻るループ経路上のアドレスのシーケンスが含まれている。 $Lconf$  は、このループ経路上をユニキャストで配送され、受信した移動コンピュータに  $S$  へ至る経路が通知される。これによって、以降受信した  $Lreq$  をブロードキャストせず、検出済みのループ経路を用いて  $S$  まで配送し、新しいループ経路を検出する。

#### [LBSR プロトコル]

0. 各移動コンピュータ  $M_i$  は以下の 4 つの変数を初期化する。 $req\_flag_i \leftarrow false$ 、 $stop\_flag_i \leftarrow false$ 、 $next_i \leftarrow null$ 、 $hops_i \leftarrow \infty$ 。
1. 送信元  $S(=M_0)$  は自身のみからなるアドレスシーケンスを含む  $Lreq$  メッセージを、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  $M_i$  にブロードキャストする。
2. 送信元  $S$  ではない移動コンピュータ  $M_i$  が  $Lreq$  メッセージを受信した場合、以下の手順でメッセージを処理する。
  - $stop\_flag_i = true$  である場合、 $Lreq$  メッセージを破棄する。
  - 自身が送信先  $D$  かつ  $req\_flag_i = true$  であるならば、 $Lreq$  メッセージを破棄する。
  - $req\_flag_i = true$  かつ  $stop\_flag_i = false$  である場合
    - $next_i = null$  の場合、 $M_i$  は  $Lconf$  メッセージを受信し、 $next_i$  が設定されるまで待機する。再開はステップ 2 の先頭からとする。
    - $next_i \neq null$  の場合、 $Lreq$  メッセージのアドレスシーケンスの末尾に自身のアドレスを追加し、 $next_i$  へユニキャストで  $Lreq$  を送信する。
  - $req\_flag_i = false$  かつ  $stop\_flag_i = false$  である場合、 $Lreq$  メッセージのアドレスシーケンスの末尾に自身のアドレスを追加し、無線信号到達範囲内にあるすべての移動コンピュータへブロードキャストする。このとき、 $req\_flag_i \leftarrow true$  とする。
3.  $Lreq$  メッセージを受信した送信元  $S$  は、 $Lreq$  メッセージのアドレスシーケンスの末尾に自身のアドレスを追加し、以下の手順でメッセージを処理する。
  - $stop\_flag_0 = false$  の場合
    - 受信した  $Lreq$  メッセージのアドレスシーケンスに送信先  $D$  が含まれる場合、 $S$  は  $stop\_flag_0 \leftarrow true$  とし、 $Lreq$  メッセージのアドレスシーケンスを  $Lconf$  メッセージに格納する。このシーケンスに含まれるアドレスの数から 1 を減じた値を  $addr\_num$  に格納し、アドレスシーケンスにおける  $S$  の次のアドレスにこの  $Lconf$  メッセージをユニキャスト送信する。
    - 受信した  $Lreq$  メッセージのアドレスシーケンスに送信先  $D$  が含まれない場合、 $Lreq$  メッセージのアドレスシーケンスを  $Lconf$  メッセージに格納する。このシー

ケンスに含まれるアドレスの数から 1 を減じた値を  $addr\_num$  に格納し、アドレスシーケンスにおける  $S$  の次のアドレスにこの  $Lconf$  メッセージをユニキャスト送信する。

- $stop\_flag_0 = true$  の場合、 $Lreq$  メッセージのアドレスシーケンスを  $Lstop$  メッセージに格納する。このシーケンスに含まれるアドレスの数から 1 を減じた値を  $addr\_num$  に格納し、アドレスシーケンスにおける  $S$  の次のアドレスにこの  $Lstop$  メッセージをユニキャスト送信する。
4. 送信元  $S$  ではない移動コンピュータ  $M_i$  が  $Lconf$  メッセージを受信した場合、以下の手順でメッセージを処理する。
    - $next_i = null$  の場合、 $M_i$  は、 $Lconf$  メッセージのアドレスシーケンスにおける  $M_i$  の次のアドレスを  $next_i$  に、 $S$  までのホップカウントを示す  $addr\_num$  の値を  $hops_i$  にそれぞれ格納する。 $Lconf$  メッセージの  $addr\_num$  から 1 を減じ、この  $Lconf$  メッセージを  $next_i$  にユニキャストで送信する。
    - $next_i \neq null$  の場合
      - $hops_i$  の値が、 $Lconf$  メッセージの  $addr\_num$  よりも大きい場合、受信した  $Lconf$  メッセージのアドレスシーケンスにおける  $M_i$  の次のアドレスを  $next_i$  に、 $addr\_num$  の値を  $hops_i$  にそれぞれ格納する。 $Lconf$  メッセージの  $addr\_num$  から 1 を減じ、この  $Lconf$  メッセージのアドレスシーケンスにおける  $M_i$  の次のアドレスに  $Lconf$  メッセージをユニキャストで送信する。
      - $hops_i$  の値が、 $Lconf$  メッセージの  $addr\_num$  よりも小さい場合、受信した  $Lconf$  メッセージの  $addr\_num$  から 1 を減じ、この  $Lconf$  メッセージのアドレスシーケンスにおける  $M_i$  の次のアドレスに  $Lconf$  メッセージをユニキャストで送信する。
  5.  $Lconf$  メッセージを受信した送信元  $S$  はこのメッセージを破棄する。
  6.  $Lstop$  メッセージを受信した移動コンピュータ  $M_i$  は、 $stop\_flag_i \leftarrow true$  とし、 $Lstop$  メッセージのアドレスシーケンスにおける  $M_i$  の次のアドレスにこの  $Lstop$  メッセージを送信する。
  7.  $Lstop$  メッセージを受信した送信元  $S$  は、このメッセージを破棄する。□

## 2.2 複数経路構築プロトコル

AODV、DSR、TORA、LBSR、C-LBSR 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、競合が発生する [17]。経路上にはない移動コ

ンピュータもルート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、経路探索時に  $S$  から  $D$  への複数の経路を検出し、検出された経路のうちの 1 つを用いてアプリケーションメッセージを配送し、この経路上にあるリンクが切断されたならば、他の経路に切替えてアプリケーションメッセージを配送する複数経路検出ルーティングプロトコルが求められている。これまでに、MultipathDSR [13]、SMR [10]、AODV-BR [9]、MNH [7]、MR-AODV [21] などが提案されている。

AODV-BR、MNH、MR-AODV では、ネクストホップベースルーティングを行なう AODV を拡張して複数の経路を検出する。AODV を複数経路検出プロトコルに拡張するために AODV-BR では、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とする。プライマリルートに含まれない移動コンピュータも  $Rrep$  を受信する。リンクの切断を検出した中継移動コンピュータはルート変更要求メッセージを無線信号到達範囲内にブロードキャストする。 $Rrep$  を受信した移動コンピュータがこのルート変更要求メッセージを受信することによって、この移動コンピュータを中継移動コンピュータとして含む迂回経路への切替えを行なう。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替えしか行なうことができない。一方、MNH は、 $Rreq$  のフラッディングを AODV と同様の方法で行なう。中継移動コンピュータは、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、中継移動コンピュータが複数回  $Rrep$  を受信する場合は、受信した  $Rrep$  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  $Rrep$  を破棄する。これにより複数の経路が検出される。しかし、複数経路を検出するためには、 $Rreq$  の受信タイミングが厳しい制約条件を満たすことが求められる。

これらの問題を解決したプロトコルが MR-AODV である。MR-AODV では、 $Rreq$  のフラッディングによりプライマリルートを検出するとともに、これに接続する枝経路を検出する。これらの枝経路を互いに接続することによって  $S$  から  $D$  への複数経路を検出する。このプロトコルでは、最初に検出された経路から複数ホップ離れた経路を検出することが可能であるとともに、 $Rreq$  の受信タイミングに依存することなく、複数経路が検出できる。この結果、MR-AODV では、AODV-BR や MNH と比較してより多くの経路を検出することが可能であり、すべての迂回経路が切断され、経路再探索を開始するまでの時間が延長される。また、複数経路検出とアプリケーションメッセージの配送を並列に実行することが可能であり、経路探索の開始からアプリケーションメッセージの配送の開始までに要する時間が AODV と同じであるという利点を持つ。ただし、リンクの切断によって、アプリケーションメッセージの配送に使用される経路を切替える際には、どのリンクが切断されたかを特定する必要があり、アプリケーションメッセージの配送が一時停止するという問題がある。論文 [22] で提案されている AODV の拡張プロトコルでは、送信元と送信先以外に共通のリンクを持たない複数の経路を検出し

ている。この方法によって、アプリケーションメッセージの配送に用いている経路に含まれるいずれかのリンクが切断した場合、その切断リンクを特定することなく、別の経路へと切替えることができる。ただし、共通のリンクを持たない経路の数が少ないために、経路再探索までの時間を延長することができない。

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。 $S$  から  $Rreq$  がフラッディングされ、 $D$  に到達すると、 $D$  は到達した複数の  $Rreq$  に格納された経路を格納した  $Rrep$  を  $S$  に送信する。これによって、 $D$  への複数の経路を  $S$  が得ることができる。

これらの複数経路検出ルーティングプロトコルでは、移動コンピュータ間のリンクがすべて双方向であることを仮定している。そこで、次章では、片方向リンクを含むアドホックネットワークのための単一経路検出ルーティングプロトコルである LBSR [18] を拡張し、複数経路検出を可能とし、リンク切断の検出とともに他の経路へと切替え、さらに無効経路を決定するプロトコルである MR-LBSR を提案する。

### 3 MR-LBSR ルーティングプロトコル

#### 3.1 複数経路検出

MR-LBSR では、 $D$  を通る複数のループ経路を検出する点が LBSR と異なる。そこで、 $D$  が複数の  $Lreq$  を受信することができるようにする点、 $D$  を通るループ経路に対しても  $Lconf$  によるループ経路確定を行なう点、 $D$  を通るループ経路を検出してもループ経路探索を終了させない点、という 3 点の変更を行なう。

0. 各移動コンピュータ  $M_i$  は以下の 3 つの変数を初期化する。 $req\_flag_i \leftarrow false$ 、 $next_i \leftarrow null$ 、 $hops_i \leftarrow \infty$ 。
1. 送信元  $S$  は自身のみからなるアドレスシーケンスを含む  $Lreq$  メッセージを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  $M_i$  にブロードキャストする。
2. 送信元  $S$  ではない移動コンピュータ  $M_i$  が  $Lreq$  メッセージを受信した場合、以下の手順でメッセージを処理する。
  - $req\_flag_i = true$  である場合
    - $next_i = null$  の場合、 $M_i$  は  $Lconf$  メッセージを受信し、 $next_i$  が設定されるまで待機する。再開はステップ 2 の先頭からとする。
    - $next_i \neq null$  の場合、 $Lreq$  メッセージのアドレスシーケンスの末尾に自身のアドレスを追加し、 $next_i$  へユニキャストで  $Lreq$  を送信する。
  - $req\_flag_i = false$  の場合、 $Lreq$  メッセージのアドレスシーケンスの末尾に自身のアドレスを追加し、無線信号到達範囲内にあるすべての移動コンピュータへブロードキャストする。このとき、 $req\_flag_i \leftarrow true$  とする。
3.  $Lreq$  メッセージを受信した送信元  $S$  は、 $Lreq$  メッセージのアドレスシーケンスの末尾に自身のアドレスを追加し、以下の手順でメッセージを処理する。
  - 受信した  $Lreq$  メッセージのアドレスシーケンスに送信先  $D$  が含まれる場合、 $Lreq$  メッセージのアドレスシーケンスを  $Lconf$  メッセージ

に格納する。このシーケンスに含まれるアドレスの数から 1 を減じた値を  $addr\_num$  に格納し、アドレスシーケンスにおける  $S$  の次のアドレスにこの  $Lconf$  メッセージをユニキャスト送信する。

- 受信した  $Lreq$  メッセージのアドレスシーケンスに送信先  $D$  が含まれない場合、 $Lreq$  メッセージのアドレスシーケンスを  $Lconf$  メッセージに格納する。このシーケンスに含まれるアドレスの数から 1 を減じた値を  $addr\_num$  に格納し、アドレスシーケンスにおける  $S$  の次のアドレスにこの  $Lconf$  メッセージをユニキャスト送信する。
- 4. 送信元  $S$  ではない移動コンピュータ  $M_i$  が  $Lconf$  メッセージを受信した場合、以下の手順でメッセージを処理する。

- $next_i = null$  の場合、 $M_i$  は、 $Lconf$  メッセージのアドレスシーケンスにおける  $M_i$  の次のアドレスを  $next_i$  に、 $S$  までのホップカウントを示す  $addr\_num$  の値を  $hops_i$  にそれぞれ格納する。 $Lconf$  メッセージの  $addr\_num$  から 1 を減じ、この  $Lconf$  メッセージのアドレスシーケンスにおける  $M_i$  の次のアドレスに  $Lconf$  メッセージをユニキャストで送信する。
- $next_i \neq null$  の場合
  - $hops_i$  の値が、 $Lconf$  メッセージの  $addr\_num$  よりも大きい場合、受信した  $Lconf$  メッセージのアドレスシーケンスにおける  $M_i$  の次のアドレスを  $next_i$  に、 $addr\_num$  の値を  $hops_i$  にそれぞれ格納する。 $Lconf$  メッセージの  $addr\_num$  をから 1 を減じ、この  $Lconf$  メッセージのアドレスシーケンスにおける  $M_i$  の次のアドレスに  $Lconf$  メッセージをユニキャストで送信する。
  - $hops_i$  の値が、 $Lconf$  メッセージの  $addr\_num$  よりも小さい場合、受信した  $Lconf$  メッセージの  $addr\_num$  から 1 を減じ、この  $Lconf$  メッセージのアドレスシーケンスにおける  $M_i$  の次のアドレスに  $Lconf$  メッセージをユニキャストで送信する。

- 5.  $Lconf$  メッセージを受信した送信元  $S$  はこのメッセージを破棄する。□

### 3.2 経路切断検出と経路切替

$R_{S \rightarrow D} + R_{D \rightarrow S} = \langle S = M_0, M_1, M_2, \dots, S \rangle$  (ただし  $\exists M_j = D$ ) のいずれかのリンク  $M_i M_{i+1}$  が切断されたことは、タイムアウトによって検出される。タイムアウトを用いるために、もし  $S$  が送信すべきアプリケーションメッセージが一時的に存在しない場合には、空メッセージを送信する。 $\{M_1, \dots, M_{j-1}\}$  を送信元とする切断検出メッセージを  $S$  が受信しないならば、切断リンクは  $\langle M_i, M_{i+1} \rangle (i = 0, \dots, j-2)$  のいずれかである。そこで、経路探索によって検出された複数の経路のうち、これらをすべて含まない経路のひとつを新しくアプリケーションメッセージの配送に用いる。切断リンク

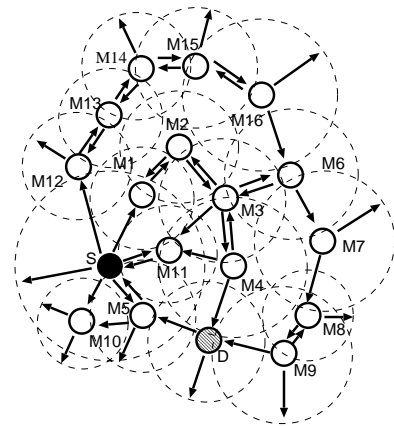


図 3: MR-LBSR におけるループ経路探索

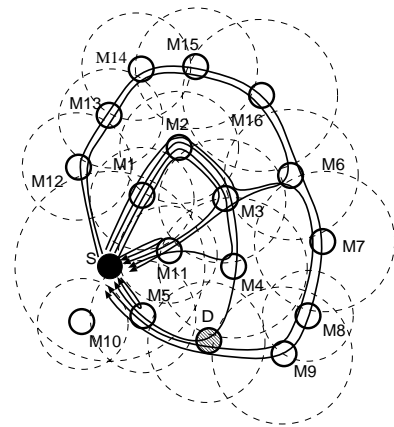


図 4: MR-LBSR におけるループ経路構築

を特定する手続きを行わないまま新しい経路を決定することから論文 [22] と同様に、アプリケーションメッセージ配送の中断時間を短縮することができる。

1. 切断検出メッセージ  $Lbreak$  を送信していない  $M_i$  が  $M_{i-1}$  からメッセージを受信することなくタイムアウトしたならば、 $M_i$  のアドレスを含む  $Lbreak$  メッセージを  $M_{i+1}$  にユニキャスト送信する。
2.  $M_i$  が  $M_{i-1}$  から  $Lbreak$  メッセージを受信したならば、この  $Lbreak$  メッセージを  $M_{i+1}$  にユニキャスト送信する。
3.  $S$  が初めての  $Lbreak$  メッセージを受信したならば、以下の処理を行う。

- 3-1. 現在メッセージ配送に用いているループ経路を無効化する。
- 3-2. 有効である  $D$  を通るループ経路のうち、経路  $\langle S, M_1, \dots, M_j \rangle$  に含まれるリンク  $M_i M_{i+1} (i = 0, \dots, j-1)$  のいずれか ( $M_j$  は  $Lbreak$  にアドレスが含まれる移動コンピュータ) を含むものをすべて無効化する。
- 3-3. 有効である  $D$  を通るループ経路のうちの 1 つを選択し、この経路に沿って  $Linit$  メッセージをユニキャストする。もし、このようなループ経路が存在しない場合には、2 回目以降の  $Lbreak$  メッセージを受信もしくは 3.3 節のブ

ロトコルにより有効なループ経路が検出されるまで待つ。それでも検出されない場合には、3.1 節の経路探索をやり直す。

- 3-4.  $M_i$  は *Linit* メッセージを受信すると、選択したループ経路における  $M_i$  の次のアドレスにこの *Linit* メッセージをユニキャスト送信する。
4.  $S$  が 2 回目以降の *Lbreak* メッセージを受信したならば、無効化されている  $D$  を通るループ経路のうち、経路  $\langle S, M_1, \dots, M_j \rangle$  ( $M_j$  は *Lbreak* メッセージにアドレスが含まれる移動コンピュータ) に含まれるリンク  $M_i M_{i+1}$  ( $i = 0, \dots, j-1$ ) のいずれも含まないものを有効化する。□

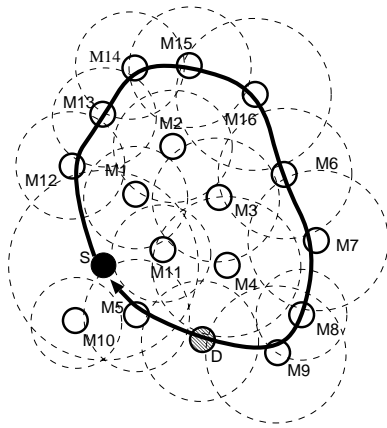


図 5: *Linit* メッセージの送信

### 3.3 無効経路の再有効化

3.2 節のプロトコルにより、十分な時間が経過することにより、 $S$  が受信した *Lbreak* にアドレスが含まれる移動コンピュータのうち、最も上流にあるものを  $M_j$  とすると、リンク  $M_{j-1} M_j$  が切断されていることが分かる。そして、3.1 節で検出された  $D$  を通るループ経路のうち、経路  $\langle S, M_1, \dots, M_j \rangle$  に含まれるリンク  $M_i M_{i+1}$  のいずれかを含む経路が無効化される。この無効化されたループ経路には、いずれのリンクも切断されていないものが含まれている。そこで、これらを検出し、再有効化するプロトコルを設計する。ここでは  $\langle S, M_1, \dots, M_{j-1} \rangle$  に含まれるリンクのうち、 $S$  から連続する切断されていない経路  $\langle S, M_1, \dots, M_k \rangle$  ( $k < j$ ) を、 $D$  を通らないループ経路に沿ってループ検査メッセージ *Lprob* をユニキャスト配送することによって検出し、経路  $\langle M_k, \dots, M_j \rangle$  に含まれるリンクを含まないループ経路を再有効化する。 $\langle S, M_1, \dots, M_k \rangle$  に切断リンクが含まれていないことは  $\langle S, M_1, \dots, M_k \rangle$  を含むループ経路にそって *Lprob* メッセージをユニキャスト配送し、これを  $S$  が受信することによって確認できる。このようなループ経路は *Lprob* メッセージの送信時に検出されるのではなく、3.1 節に述べた経路探索時に検出できる。経路探索プロトコルでは  $S$  を含み、 $D$  を含まない複数のループ経路が検出されることから、これらを経路キャッシュに保持し、*Lprob* メッセージの配送経路として用いればよい。

1.  $S$  は 3.1 節で検出したループ経路 ( $D$  を含まなくてもよい) のうち、経路  $\langle S, M_1, \dots, M_{j-1} \rangle$  に含まれるリンクを 1 つ以上含むすべてのループ経路に

対して、この経路を含む *Lprob* をユニキャスト送信する。

2. *Lprob* を受信した  $M_i$  は、これに含まれる経路に沿って受信した *Lprob* をユニキャスト送信する。
3. *Lprob* を受信した  $S$  はこれに含まれる経路が経路  $\langle S, M_1, \dots, M_k \rangle$  を含むならば、現在無効化されているループ経路のうち経路  $\langle M_{k+1}, \dots, M_j \rangle$  に含まれるリンクを含まないものを再有効化する。□

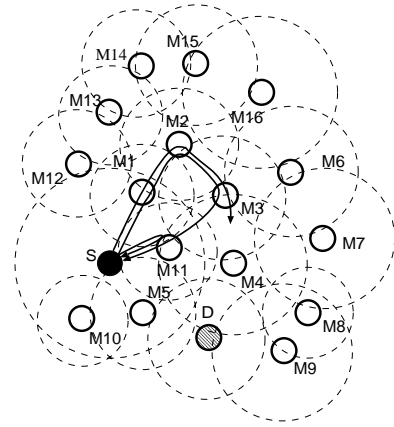


図 6: *Lprob* による経路切断箇所検出

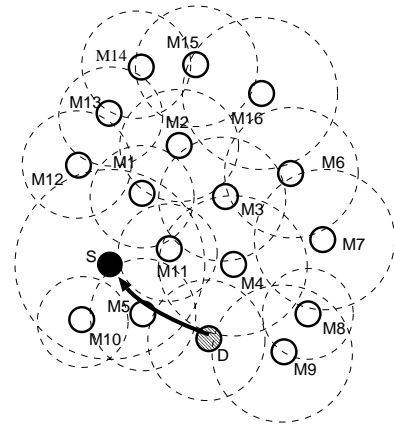


図 7: *Lbreak* メッセージの送信

## 4 評価

本章では、MR-LBSR の性能を評価する。評価はシミュレーションにより行った。シミュレーション領域は  $500\text{m} \times 500\text{m}$  の正方形とし、各移動コンピュータの無線信号到達範囲の直径は  $20\text{m}$  から  $200\text{m}$  までの均等分布であるとする。各移動コンピュータの移動モデルは Random Way Point とし、移動速度は  $0\text{km/h}$  から  $20\text{km/h}$  までの均等分布である。送信元移動コンピュータ  $S$ 、送信先移動コンピュータ  $D$  の組はランダムに選択したシミュレーションを各 1000 回行った結果を示している。

まず、経路探索プロトコルによる検出経路数、検出に要するメッセージ数、検出に要する時間の測定結果を

それぞれ図 8、図 9、図 10 に示す。MR-LBSR による経路探索は LBSR に比べて長い時間を要するが、最初に検出した経路は他の経路の検出を待たずに、アプリケーションメッセージの配送に用いることができる。これは、他の経路の探索とアプリケーションメッセージの配送が並列に実行できるからであるためである。図 10 に示すように、経路探索を開始してからアプリケーションメッセージの配送が開始されるまでの時間は、MR-LBSR と LBSR では同じである。

次にアプリケーションメッセージの配送に用いる経路が切断したときに、経路を切替えるのに要する時間を測定した結果を図 11 に示す。LBSR の測定値は経路検出時間であり、図 10 と等しい。MR-LBSR の測定値は切断検出時に共通リンクを含まない迂回経路が存在するならば、*Linit* の配送時間に等しく、迂回経路が経路キャッシュに存在しないならば、迂回経路を検出するまでの時間と *Linit* の配送時間の和に等しい(迂回経路が検出できない場合は、迂回経路の探索時間と再経路探索の時間の和に等しい)。このグラフからアプリケーションメッセージの配送の一時停止時間が MR-LBSR に比べて約 13% に削減されていることが分かる。また、この切断リンクの特定に要する制御メッセージ数の測定結果を図 12 に示す。経路検査に要する制御メッセージ数は LBSR の経路探索の約 10% しか必要がないことがわかる。

最後に再経路探索までの時間を測定した結果を図 13 に示す。MR-LBSR によって LBSR と比較して再経路探索までの時間を 50% 延長することができた。特に経路の再有効化プロトコルによって 30% の延長が実現されている。

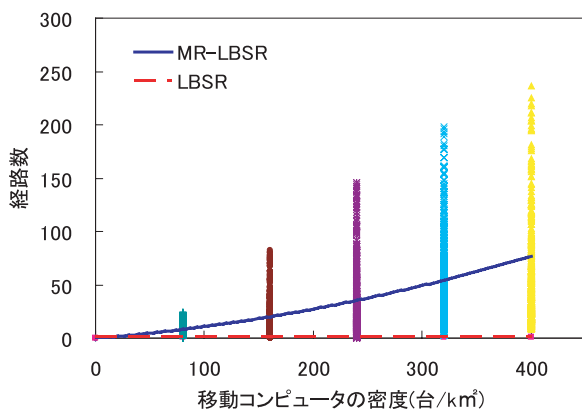


図 8: 経路探索プロトコルによる検出経路数

## 5 まとめ

本論文では、片方向リンクを含むアドホックネットワークを対象として、ループ経路探索に基づいて経路探索を行なう LBSR プロトコルを、リンク切断に対して安定な通信を提供するために複数経路を検出するように拡張した MR-LBSR を提案した。MR-LBSR では、リンク切断の検出により、無効化した現経路と共通リンクを持たない経路に直ちに切替えることにより、アプリケーションメッセージ配送の中断時間を短縮している。また、経路探索時に検出した送信元を含み送信先を含ま

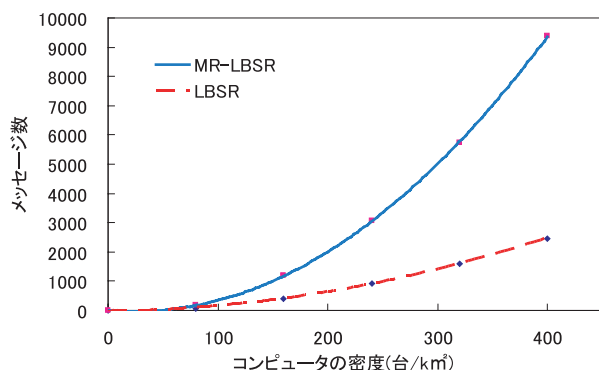


図 9: 検出に要するメッセージ数

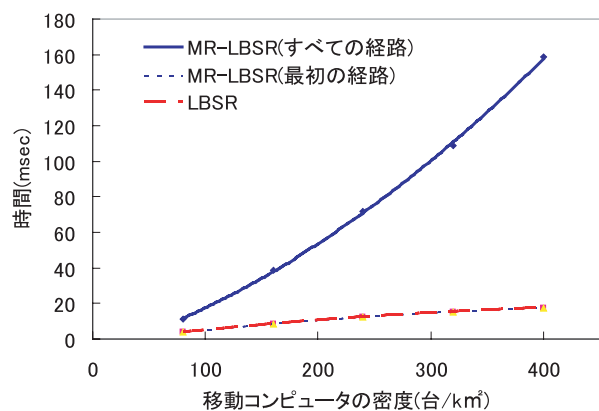


図 10: 検出に要する時間

ないループ経路を使用して切断リンクを特定し、それを含まない経路を再有効化することで、再経路探索までの時間を延長している。

## 参考文献

- [1] "Wireless LAN Medium Access control (MAC) and Physical Layer (PHY) Specifications," Standard IEEE 802.11 (1997).
- [2] "The Official Bluetooth Wireless Info Site," <http://www.bluetooth.com>.
- [3] "Radio Equipment and Systems (RES); HIPERLAN," ETSI Functional Specifications (1995).
- [4] Corson, M.S. and Ephremides, A., "A Distributed Routing Algorithm for Mobile Wireless Networks," *Wireless Networks*, vol. 1, No. 1, pp. 61-81 (1995). Internet Draft, draft-ietf-manet-dsr-04.txt (2000).
- [5] 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).
- [6] Hedrick, C., "Routing Information Protocol," RFC 1058 (1988).
- [7] 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).
- [8] Kim, D.K., Toh, C.K. and Choi, Y.H., "RODA: A new dynamic routing protocol using dual paths to support

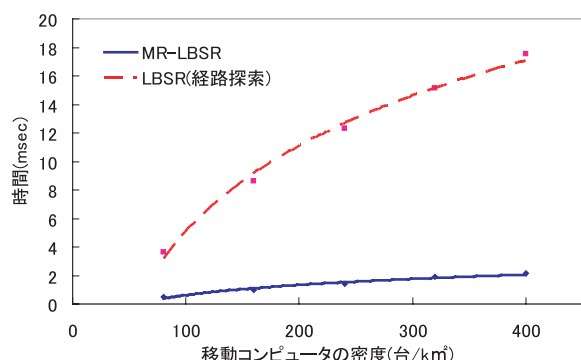


図 11: 経路切替えに要する時間

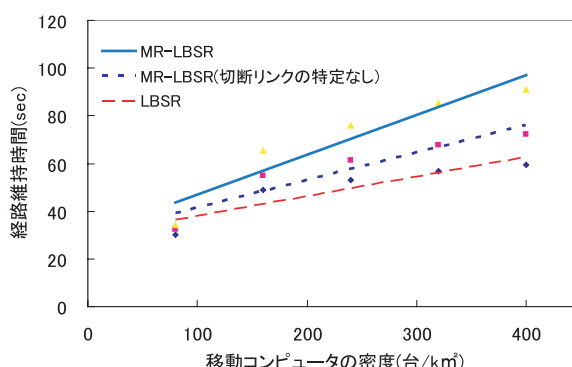


図 13: 再経路探索までの時間

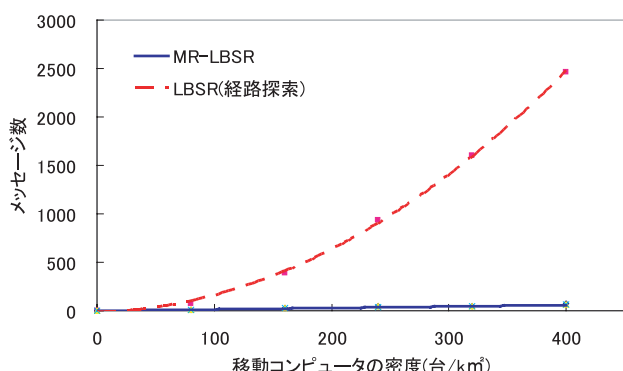


図 12: 切断リンクの特定に要するメッセージ数

asymmetric links in mobile ad hoc networks,” Proc. of the 9th International Conference on Computer Communications and Networks, pp. 4–8 (2000).

- [9] 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).
- [10] 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).
- [11] Moses, Y. and Roth, G., “On reliable message diffusion,” Proc. of the 8th ACM Symposium on Principles of Distributed Computing, pp. 119–128 (1989).
- [12] Moy, J., “Open Shortest Path First specification,” RFC 1131 (1989).
- [13] 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).
- [14] Park, V. and Corson, S., “Temporally-Ordered Routing Algorithm (TORA) Version 1 Functional Specification,” Internet Draft, draft-ietf-manet-tora-spec-04.txt (2001).
- [15] Perkins, C.E. and Bhagwat, P., “Highly Dynamic Destination-Sequenced Distance-Vector Routing (DSDV) for Mobile Computers,” ACM SIGCOMM’ 94, pp. 234–244 (1994).
- [16] Perkins, C.E. and Royer, E.M., “Ad-hoc On-Demand Distance Vector Routing,” Proc. of IEEE 2nd Work-

shop on Mobile Computing Systems and Applications, pp. 90–100 (1999).

- [17] Tseng, Y., Ni, S. and Shih, E., “Adaptive Approaches to Relieving Broadcast Storms in a Wireless Multihop Mobile Ad Hoc Network,” Proc. of the 21st International Conference on Distributed Computing Systems, pp. 481–488 (2001).
- [18] 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).
- [19] 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).
- [20] 西澤, 荻野, 原, 塚本, 西尾, “アドホックネットワークにおける片方向リンクを考慮したルーティング方式,” 情報処理学会論文誌, Vol. 41, No. 3, pp. 783–791 (2000).
- [21] 長谷部, 梅島, 桧垣, “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,” 情報処理学会論文誌, Vol. 49, pp. 25–32 (2002).
- [22] 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,” 信学技報, IN2002-125, pp. 51–56 (2002).