

LBSR(Loop-Based Source Routing Protocol) の Linux への実装とその評価

東京電機大学大学院 理工学研究科 情報システム工学専攻

黒沢 祐介 浅野 知倫 卯木 宏幸 桧垣 博章

E-mail: {kuro,tom,unoki,hig}@higlab.k.dendai.ac.jp

ネットワークを移動コンピュータのみで構成したアドホックネットワークは、アクセスポイントを介した無線ネットワークに比べ高い柔軟性と移動性を持つ。本論文では、モバイルコンピュータが送信する無線信号の到達範囲を同一であると仮定せず、アドホックネットワーク内に複数の片方向リンクが存在することを仮定した新たなオンデマンド型ルーティングプロトコル LBSR(Loop-Based Source Routing) プロトコルを提案し、その Linux OS のアプリケーションとして実装について述べる。また、性能評価実験結果を報告する。

Impementation and Evaluation of LBSR in Linux Environment

Yusuke Kurosawa Tomonori Asano Hiroyuki Unoki Hiroyuki Higaki

Department of Computers and Systems Engineering

Tokyo Denki University

E-mail:{kuro,tom,unoki,hig}@higlab.k.dendai.ac.jp

In an ad-hoc network where only mobile computers are included, higher mobility and flexibility are supported than an infrastructured network with access point. In this paper, we propose LBSR(Loop-Based Source Routing) which is a novel on-demand routing protocol supporting asymmetric wireless communication links. For achieving a route from a source to destination, LBSR searches multiple loop routes. LBSR is implemented as an application of Linux operating system. We report brief evaluation results.

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 [2] や HIPER-LAN [1] といった無線 LAN プロトコル技術の研究開発、利用が進み、移動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、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 (Infrastructured 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、これを経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、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。

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は、様々なものが提案されている [6,9,11,12,5,7,16]。これらの多くは、各移動コンピュータから送信される無線信号の到達範囲の大きさがすべて同一であり、移動コンピュータ間が双方向接続されていることを仮定している。しかし、無線通信では、無線信号の減衰、反射、回折により

必ずしも双方向接続できるとは限らない。また、移動コンピュータの電源容量により、その送受信能力は一定ではない。さらに、アドホックネットワークは今後様々な無線通信メディアで構成されることが考えられるため、従来の双方向接続を仮定したルーティングプロトコルでは、片方向接続リンクが存在する場合に、経路の検出確率が低下する可能性がある。すなわち、アドホックネットワークの可用性 (アベイラビリティ) を高く保つために、片方向接続リンクをも用いたルーティングプロトコルの設計、仕様が求められている。これまでに提案された片方向接続リンクを用いるアドホックルーティングプロトコルとして DSR [5] や CBRP [7]、ULSR [16] などがある。しかし、経路探索時に 2 組のフラッディングを用いる、経路探索をするために大きなトラフィックが発生する、片方向接続リンクの割合が小さい場合にのみ効率的な経路探索が可能となる、といった問題がある。そのため、本論文では、片方向接続リンクを持つアドホックネットワークにおいて、経路探索時に送信元から送信先を経由し送信元へと戻るループ経路を検出することにより、1 組のフラッディングと複数のユニキャストの組み合わせによって経路検出可能なルーティングプロトコルである LBSR (Loop-Based Source Routing) プロトコルを提案し、その Linux コンピュータへの実装について報告する。

2 アドホックルーティングプロトコル

アドホックネットワークにおいて、移動コンピュータが他の移動コンピュータと通信を行うためには、各移動コンピュータが経路情報を何らかの方法で取得する必要がある。その基本的な方法は、テーブルドリブ型とオンデマンド型の 2 つに分類することが可能である [10]。

[テーブルドリブン型]

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) [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 [11] 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。□

[オンデマンド型]

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、検出した経路に関する情報のみを管理し、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 [12]、DSR [5] などが提案されている。移動コンピュータの移動によってネットワークポロジが経時的に変化するアドホックネットワークに対しては、通信を行おうとした時点から実際にデータが送信されるまでに経路探索のための遅延が生じるという問題がある。しかし、通信開始時のネットワーク構成に基づいた経路が検索される点が優れている。□

無線通信に用いられる無線信号には、光や電波などがある。このような無線信号を用いたネットワークでは、固定ネットワークのように隣接コンピュータ間が必ずしも双方向通信可能であるとは限らない。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移動コンピュータ間が双方向接続されていることを仮定している。本論文では、アドホックネットワーク内に片方向接続リンクが多数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を提案する。オンデマンド型を用いた場合の経路探索において、双方向接続を使用する場合の経路探索と片方向接続リンクも用いる場合の経路探索との相違を以下に示す。

[双方向接続リンクのみを用いるルーティング]

ネットワーク内のすべての移動コンピュータが送信する無線信号の到達距離は同一であると仮定する方法である。無線信号到達距離が等しいので、経路要求メッセージを送信元から送信先へ配送するだけで、送信元から送信先への経路と送信先から送信元への経路を検出することが可能である。□

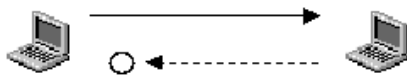


Figure 1: 双方向接続リンク

[片方向接続リンクも用いるルーティング]

移動コンピュータが送信する無線信号の到達距離は同一ではないと仮定する方法である。無線信号到達距離が同一ではないとすることで、無線信号の減衰、反射、回折や、移動コンピュータのバッテリー残量が異なるために無線信号出力電力が移動コンピュータごとに異なる環境、複数の無線通信メディアが混在するネットワーク環境に対応することが可能である。各モバイルコンピュータの無線信号到達距離は同一ではないとしているため、送信元からの経路要求メッセージによって得られる経路は送信元から送信先への経路のみである。よって、送信先から送信元への経路を検出するための手法が必要となる。□



Figure 2: 片方向接続リンク

3 DSR

すべての移動コンピュータ間の接続が双方向であることを仮定しないオンデマンド型ルーティングプロトコルとして DSR (Dynamic Source Routing) [5] プロトコルがある。DSR では、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 への経路を探索し、検出した経路 $R_{S \rightarrow D}$ を用いて S がメッセージをソースルーティングする。経路探索にはフラッディングが用いられる [4]。フラッディングとは、message diffusion protocol [8] を無線 LAN 環境に適用したものである。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。DSR では、経路要求メッセージ RREQ をフラッディングによって S から D まで配送するとともに、 D で検出された $R_{S \rightarrow D}$ を S に伝えるために、 $R_{S \rightarrow D}$ を含む経路応答メッセージ RREP をフラッディングによって S まで配送する。以下に、DSR の経路探索プロトコルを示す。

[DSR プロトコル (図 3, 図 4)]

1. S は、 S のアドレスを格納した経路要求メッセージ RREQ (Route Request) を S の無線到達範囲内にある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。RREQ には D があて先として指定される。
2. M_i が RREQ を受信する。このとき既に RREQ を受信していた場合、この RREQ を破棄する。初めて受信する RREQ である場合、受信した RREQ に自身のアドレスを加え、 M_i の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 RREQ をブロードキャストする。
3. 2. を繰り返すことにより RREQ のうちの 1 つを D が受信する。このとき、RREQ には $R_{S \rightarrow D}$ 上にある移動コンピュータのアドレスのシーケンスが含まれる。 D は、 $R_{S \rightarrow D}$ を含む RREP を D の無線信号到達範囲内にあるすべての移動コンピュータ M_i に対してブ

ロードキャストする。

4. M_i が RREP を受信する。このとき既に RREP を受信していた場合は、この RREP を破棄する。初めて受信する RREP である場合、 M_i の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 RREP をブロードキャストする。
5. 4. を繰り返すことにより RREP のうちの1つを S が受信する。これによって、 S は $R_{S \rightarrow D}$ を得ることができる。以降、データを含むメッセージを $R_{S \rightarrow D}$ を用いたソースルーティングにより配送する。□

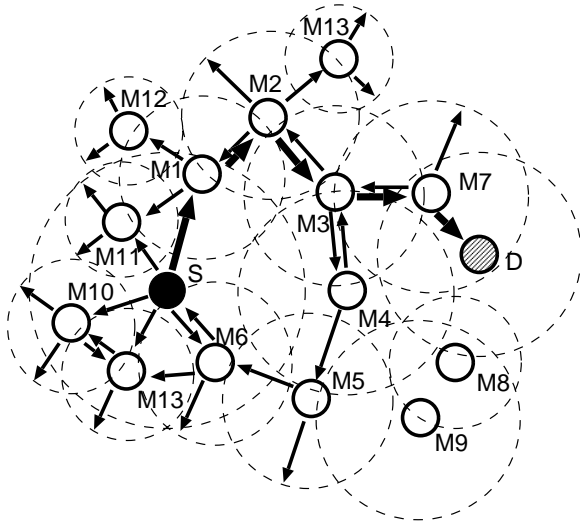


Figure 3: RREQ のフラッディング

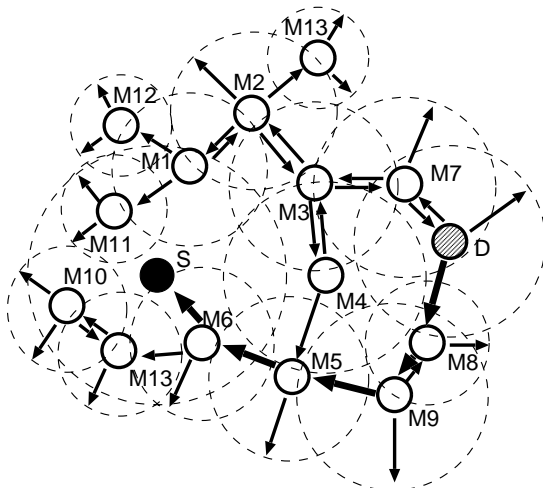


Figure 4: RREP のフラッディング

4 LBSR

片方向接続を含むアドホックネットワークにおいて、送信元 S から送信先 D までの経路情報を S が取得するためには、 S から D への経路 $R_{S \rightarrow D}$ と D から S への経路 $R_{D \rightarrow S}$ が必要である。DSR では、これらの2つの経路は独立なフラッディングによって求められるのに対して、LBSR ではこれらを連結して得られるループ経路

を探索する。特に、新しく検出したループ経路が既に検出されているループ経路の一部を含む場合、その共通部分においては制御メッセージをブロードキャストせず、ユニキャストで配送することによって通信オーバーヘッドを削減している。LBSR では、経路探索時に3種類のメッセージ LREQ と LCONF、LSTOP 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シーケンスが含まれている。LCONF には、 S から S へ戻るループ上のアドレスシーケンスが含まれている。LCONF は、このループ上をユニキャストで配送される。LCONF を受信した各移動コンピュータ M_i は、LCONF のアドレスシーケンスの情報から自分の1ホップ先に存在する移動コンピュータのアドレスを獲得し、以後 LREQ メッセージを受信した場合、獲得した移動コンピュータにユニキャストで送信する。LSTOP は、既に送信元 S が $R_{S \rightarrow D}$ を保持している場合、LREQ を受信するとそのループ経路に含まれる移動コンピュータに対して LSTOP を送信する。LSTOP を受信した M_i は、以後受信した LREQ メッセージを破棄する。これにより、 D を含むループ経路が検出された後に、経路探索のために交換される制御メッセージを削減することができる。

[LBSR プロトコル (図 5, 図 6)]

1. 送信元 S は LREQ メッセージのアドレスシーケンスに自身を加え、無線信号到達範囲内に存在する移動コンピュータ 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。
 - $stop_flag_i = true$ の場合、LREQ メッセージを破棄する。
 - 自身が送信先 D であり、かつ、 $req_flag_i = true$ であるならば、LREQ メッセージを破棄する。
 - $req_flag_i = true$ であり、 $stop_flag_i = false$ 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、経路情報を保持し待機する。
 - $next_i \neq null$ の場合、設定されている送信先に対し、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、ユニキャストで送信する。
 - $req_flag_i = false$ 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、 $req_flag_i = true$ とする。
3. 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 hop_count_i にそれぞれ格納する。このとき、LCONF メッセージのアドレスシーケンスから自身のアドレスを削除し、 $addr_num$ をデクリメントした後に、この LCONF メッセージを $next_i$ にユニキャスト

ストで送信する。

- $next_i \neq null$ の場合

- hop_count_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 hop_count_i にそれぞれ格納する。LCONF メッセージのアドレスシーケンスから自身のアドレスを削除し、 $addr_num$ をデクリメントした後にこの LCONF メッセージを $next_i$ にユニキャストで送信する。
- hop_count_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 メッセージのアドレスシーケンスから自身のアドレスを削除し、 $addr_num$ をデクリメントした後に、この LCONF メッセージをアドレスシーケンスの先頭にある移動コンピュータにユニキャストで送信する。

4. LSTOP メッセージを受信した移動コンピュータ M_i は、 $stop_flag_i$ を *true* にし、LSTOP メッセージをアドレスシーケンスに格納されている次のノードに送信する。

5. LREQ メッセージを受信した送信元 S は、以下の手順でメッセージを処理する。

- $detect_flag=false$ 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 が存在する場合、 S は $detect_flag$ を *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。アドレスシーケンスから自身のアドレスを削除し、このシーケンスに含まれるアドレスの数を $addr_num$ に格納した後に、アドレスシーケンスの先頭にある移動コンピュータにこの 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。アドレスシーケンスから自身のアドレスを削除し、このシーケンスに含まれるアドレスの数を $addr_num$ に格納した後に、アドレスシーケンスの先頭にある移動コンピュータにこの LCONF メッセージをユニキャスト送信する。

- $detect_flag=true$ 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。アドレスシーケンスから自身のアドレスを削除し、このシーケンスに含まれるアドレスの数を $addr_num$ に格納した後に、アドレスシーケンスの先頭にある移動コンピュータにこの LSTOP メッセージをユニキャスト送信する。

6. 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。

5 LBSR の実装

本章では、LinuxOS をインストールした PC への LBSR の実装について述べる。ここでは、LBSR プロトコルをアプリケーションプログラムとして実装した。LBSR の制御メッセージ、アプリケーションメッセージ

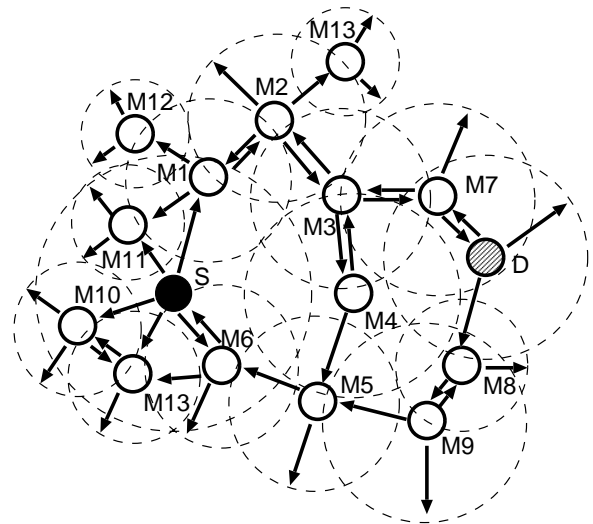


Figure 5: LREQ のフラッディング

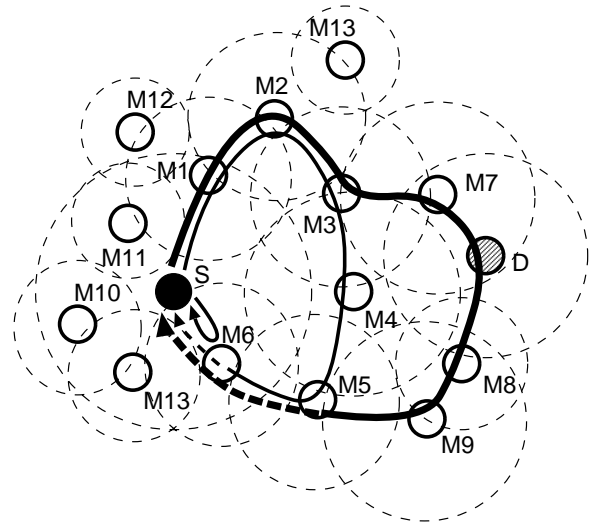


Figure 6: LCONF のユニキャスト

は図7に示す MAC フレームによって配送される。MAC ヘッダの送信先フィールドには常にブロードキャストアドレスを格納する。このようにすることで、MAC 層での受信確認フレーム (ACK) の返信を抑制することが可能となり、片方向接続リンクを用いた通信を実現することができる。ユニキャストとブロードキャストの区別は、IP ヘッダの送信先フィールドに格納されたアドレスを参照することで可能としている。LBSR ヘッダの送信元フィールドと送信先フィールドには、メッセージの送信元と送信先のアドレスを格納する。LBSR のプロトコルの機能を実現する関数とその関係を図8に示す。

MAC Header	IP Header	LBSR Header	DATA
------------	-----------	-------------	------

Figure 7: LBSR メッセージを配送するフレーム構成

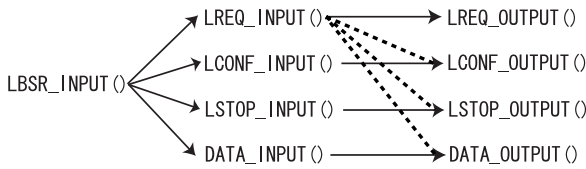


Figure 8: LBSR の構成

送信元 S は、アプリケーションプログラムの通信要求を受けるとまずキャッシュに求める経路が格納されているかを確認する。もし、送信先を含む経路情報がキャッシュに存在するならば、その経路情報が関数 $data_output$ に渡され、送信処理が行われる。もし、キャッシュに送信先への経路情報が保持されていないならば、関数 $lreq_output$ に制御が渡され、経路探索が開始される。

他の移動コンピュータからの受信は、PacketSocket [3] を通して行うことにより、アプリケーションプログラムで受信フレーム全体についての処理を行うことを可能としている。受信したパケットはその種類にかかわらず、関数 $main_lbsr$ によって処理される。ここでは、LBSR ヘッダ内の $type$ フィールドの値によって、LBSR メッセージの分類が行われる。送信元ではない移動コンピュータ M_i が LREQ を受信したならば、関数 $lreq_input$ に制御が渡される。 $lreq_flag=false$ 、もしくは、 $next_i \neq null$ の場合、LREQ メッセージを送信するために関数 $lreq_output$ が呼び出される。それ以外の場合は、キャッシュにアドレスシーケンスを格納し、 $next_i$ が設定されるまで、サスペンドするか破棄される。送信元 S が LREQ メッセージを受信した場合も同様に、関数 $lreq_input$ が呼び出される。このとき、 $detect_flag=true$ であるならば、関数 $lstop_output$ が呼び出される。 $detect_flag=false$ であり、受信した LREQ メッセージに含まれるアドレスシーケンス内に送信先 D が存在するならば、 $detect_flag$ を $true$ とし、関数 $lstop_output$ を呼び出す。それ以外の場合は、関数 $lconf_output$ が呼び出される。LCONF を受信したならば、関数 $lconf_input$ に制御が渡される。このとき、受信した LCONF メッセージのアドレスシーケンスを調べる。アドレスシーケンス内に存在する自身のアドレスの次のアドレスを $next_i$ に格納し、送信元までのホップ数を hop_count_i に格納する。その後、関数 $lconf_output$ に制御が渡され、ユニキャストで次のコンピュータに送信される。LSTOP を受信したならば、関数 $lstop_input$ に制御が渡される。LSTOP を受信した移動コンピュータ M_i は、 $stop_flag$ を $true$ にし、関数 $lstop_output$ に制御が渡され、ユニキャストで LSTOP のアドレスシーケンスに存在する自身の次のノードに送信する。DATA メッセージを受信したならば、LBSR ヘッダの送信先フィールドに格納されたアドレスを調べる。自身のアドレスであれば、受信したデータをアプリケーションプログラムに渡す。それ以外の場合、DATA メッセージのアドレスシーケンスにしたがってユニキャストでメッセージを送信する。なお、LBSR で用いる 4 種類のメッセージフォーマットと各移動コンピュータが保持するキャッシュのフォーマットを図 9、図 10 にそれぞれ示す。

現在までに主要な機能の実装は終了し、6 台の移動コンピュータにおける経路探索、データ配送の正常動作を

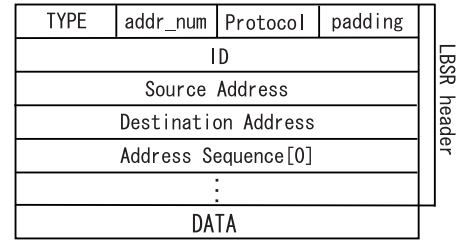


Figure 9: LBSR メッセージフォーマット

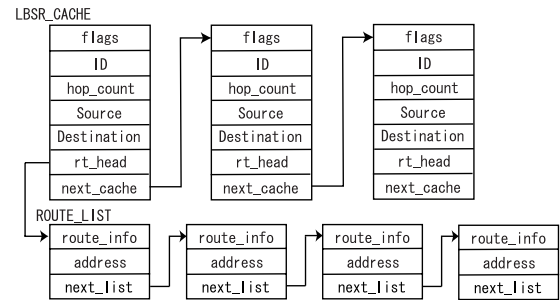


Figure 10: LBSR キャッシュフォーマット

確認した。片方向接続リンクを含む場合の動作についても確認済みである。

6 性能評価

本章では、LBSR の性能を DSR と比較する。送信元 S から送信先 D までの経路を DSR を用いて探索する場合、2 組のフラッディングを必要とする。フラッディングでは、制御メッセージが全ての通信リンクを通して配達されることから、リンク数 $|L|$ のメッセージが交換される。したがって、DSR の経路探索に要するメッセージ数は $2|L|$ である。一方、LBSR では 1 組のフラッディングと複数のユニキャストによって経路が探索される。LCONF メッセージは、検出したループ経路に沿って配達されることから、 i 番目に検出したループ経路の長さを l_i とすると、LBSR で必要とされるメッセージ数は、 $|L| + \sum_i l_i$ であると言える。まず、シミュレーション

実験について述べる。シミュレーション領域の大きさを $500m \times 500m$ とし、20 台 ~100 台の移動コンピュータが均等分布 (Unique Distribution) するものとする。また、各異動コンピュータの無線信号到達範囲は 20m から 200m までの均等分布であるとする。図 11 はブロードキャストメッセージ数、図 12 は全体のメッセージ数を示している。LBSR は、DSR の半分のブロードキャストメッセージしか要しないものの、全体では DSR の約 2 倍のメッセージを必要としている。これは、LCONF メッセージが同一のリンク上を複数回配送されること、特に、送信元 S と近辺のリンクではその傾向が顕著であることが原因である。ただし、論文 [13] で述べられているように、LBSR は DSR と比較して 1 回の経路探索でより多くの経路をキャッシュに格納することができる。図 13 と図 14 に格納されるキャッシュエントリ数の評価結果を示す。DSR は一般にキャッシュが有効に機能するプロトコルであると言われているが、それは双方向接続リンクのみを用いる環境においてである。片方向接

続リンクを含む場合には、図 13 と図 14 に示すように、移動コンピュータ数や検出した経路のホップ数 (送信元 S と送信先 D との間の距離) によらず、わずかな経路しかキャッシュに格納できない。一方、LBSR では、DSR より多くの経路をキャッシュに格納できる。また、移動コンピュータの数の増加や送信元 S と送信先 D が離れている場合、図 11 に示すようにより多くのメッセージを必要とするが、それによってキャッシュに格納される経路情報量も増加している。

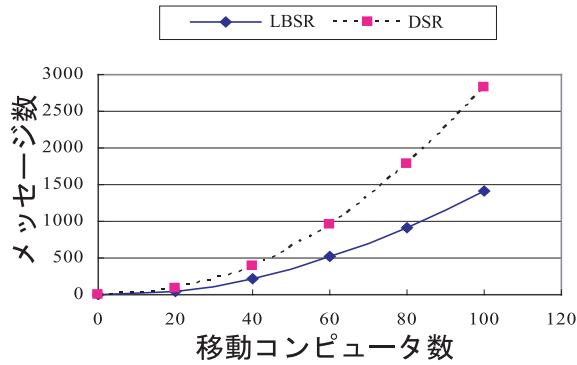


Figure 11: Number of Broadcast Messages.

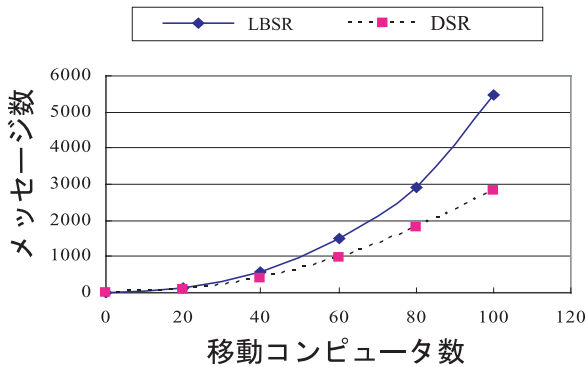


Figure 12: Total Number of Messages.

2 つのネットワークポロジと 2 つの送信タイプによる合計 4 種類の環境において DSR と LBSR の性能評価を行った。ネットワークポロジを図 16 と図 17 にそれぞれ示す。Network1 の想定環境は、電車内や道路といった場所での通信を想定している。図 16 は、電車内や道路といった場所での通信を想定したものである。また、図 17 は、2 つの片方向接続リンクで接続されているグループが双方向接続リンクで互いに接続された状況を想定している。

表 1: 実験環境

実装 OS	Vine Linux, RedHat Linux
無線 LAN プロトコル	IEEE802.11
移動コンピュータ数	6 台

また、表 2 に示す 2 種類の送信環境で評価実験を行った。

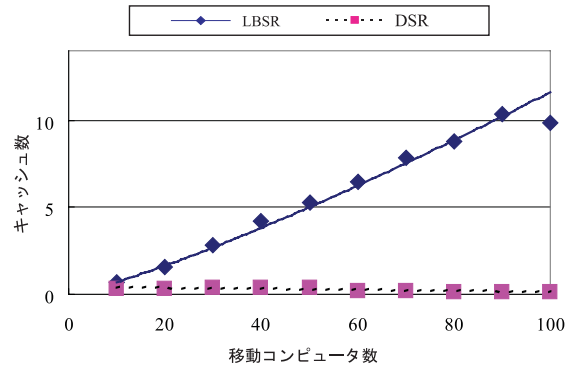


Figure 13: Number of Cache Entries in Each Mobile Computer(1).

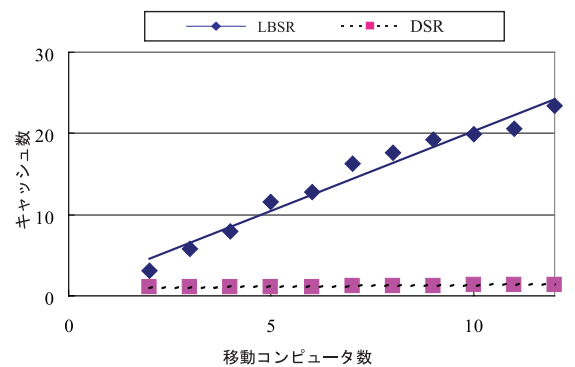


Figure 14: Number of Cache Entries in Each Mobile Computer(2).

表 2: 送信方法

	1 回目	2 回目	3 回目
送信方法 1	$M6 \rightarrow M1$	$M1 \rightarrow M6$	$M3 \rightarrow M6$
送信方法 2	$M6 \rightarrow M1$	$M6 \rightarrow M3$	$M6 \rightarrow M5$

このような環境下において、LBSR と DSR でネットワークに流れるメッセージ数を計測した結果を、表 3、表 4、表 5、表 6 に示す。プロトコル名の後に (S), (R) はそれぞれ送信メッセージ数と受信メッセージ数を示す。

表 3: ネットワーク 1-送信方法 1

	DSR(S)	DSR(R)	LBSR(S)	LBSR(R)
1 回目	20.0	28.4	56.0	60.0
2 回目	20.0	28.0	10.0	10.0
3 回目	14.0	18.8	6.0	6.0

表 4: ネットワーク 1-送信方法 2

	DSR(S)	DSR(R)	LBSR(S)	LBSR(R)
1 回目	20.0	28.0	56.0	60.0
2 回目	20.0	28.0	10.0	10.0
3 回目	14.0	18.8	6.0	6.0

表 5: ネットワーク 2-送信方法 1

	DSR(S)	DSR(R)	LBSR(S)	LBSR(R)
1 回目	18.0	22.0	36.0	39.0
2 回目	18.0	22.0	8.0	8.0
3 回目	13.0	13.0	8.0	8.0

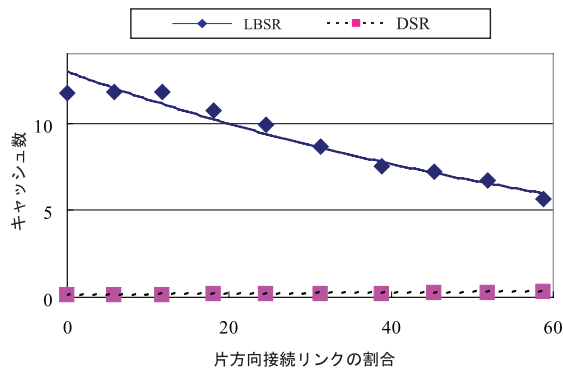


Figure 15: Number of Cache Entries in Each Mobile Computer(3).

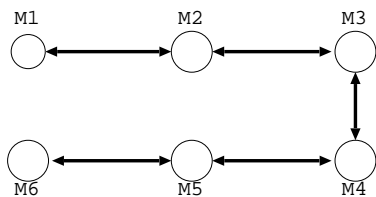


Figure 16: 実験ネットワーク (1)

表 6 : ネットワーク 2-送信方法 2

	DSR(S)	DSR(R)	LBSR(S)	LBSR(R)
1 回目	18.0	22.0	36.0	39.0
2 回目	8.0	8.0	8.0	8.0
3 回目	8.0	8.0	8.0	8.0

以上の結果より、最初の経路構築において DSR より多くのメッセージ数を必要とするが、2 回目以降は LBSR のキャッシュの効果により、DSR よりも必要なメッセージ数が少なくなる。この結果は、シミュレーションによる結果とほぼ同一の結果である。今後、キャッシュに経路情報が含まれていない場合のメッセージ数を削減することが必要である。この拡張プロトコルは、論文 [14] で述べられる予定である。

7 まとめと今後の課題

本論文では、PacketSocket を用いることで LinuxOS における実装を行った。プロトコルスタック、動作システムについて述べ、DSR と LBSR の比較評価を行った。今後は、LBSR のメッセージを減らすための拡張プロトコルを実装すること、数 10 台から数 100 台規模の実機

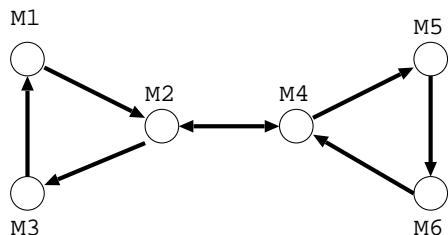


Figure 17: 実験ネットワーク (2)

による性能評価実験を行うことが課題である。

References

- [1] "Radio Equipment and Systems (RES); HIPER-LAN," ETSI Functional Specifications (1995).
- [2] "Wireless LAN Medium Access control (MAC) and Physical Layer (PHY) Specifications," Standard IEEE 802.11 (1997).
- [3] Online Manual "socket," Red Hat Linux.
- [4] 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).
- [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-09.txt (2003).
- [6] Hedrick, C., "Routing Information Protocol," RFC 1058 (1988).
- [7] Jiang, M., Li, J. and Tay, Y.C., "Cluster Based Routing Protocol(CBRP) Functional Specification," Internet Draft, draft-ietf-manet-cbrp-00.txt (1999).
- [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] Perkins, C.E., "Ad Hoc Networking," Addison-Wesley (2001).
- [11] Perkins, C.E. and Bhagwat, P., "Highly Dynamic Destination-Sequenced Distance-Vector Routing (DSDV) for Mobile Computers," ACM SIGCOMM' 94, pp. 234-244 (1994).
- [12] 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).
- [13] Sagawa, Y., Asano, T. and Higaki, H., "Loop-Based Source Routing Protocol for Mobile Ad-hoc Networks," Proc. of the 17th International Conference on Advanced Information Networking and Applications (AINA2003), pp. 834-837 (2002).
- [14] 卯木, 桧垣, "LBSR プロトコルのメッセージ数削減手法," 情報処理学会研究報告, (to appear).
- [15] 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).
- [16] 西澤, 萩野, 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).