

ループ探索に基づいたアドホックルーティングプロトコルの実装

神林 洋平 佐川 陽介 桧垣 博章
東京電機大学 理工学部 情報システム工学科
〒350-0394 埼玉県比企郡鳩山町石坂

E-mail: {kan, sgw, hig}@higlab.k.dendai.ac.jp

ノート型PCやPDAといった移動コンピュータをIEEE802.11などの無線LAN技術によって接続することで構成されるモバイルネットワークを基礎としたネットワークコンピューティング環境への要求が高まっている。モバイルネットワークのひとつにアドホックネットワークがある。アドホックネットワークでは、専用のルータではなく、移動コンピュータがメッセージをルーティングすることによって、移動性、柔軟性、経済性の高いネットワークを構築できる。コンピュータの移動性を考慮したオンデマンド型のアドホックルーティングプロトコルとしてDSR, AODV, DSDV等が提案、実装されている。本論文では、アドホックネットワークの多様性と不安定性を考慮し、ネットワーク内に片方向接続がする場合においても経路検出可能なルーティングプロトコルLBSR (Loop Based Source Routing) を提案し、FreeBSD3.3+PAOへの実装について述べる。

Implementation of Loop Based Source Routing Protocol

Youhei Kambayashi, Yousuke Sagawa and Hiroaki Higaki

{kan, sgw, hig}@higlab.k.dendai.ac.jp

Department of Computers and Systems Engineering
Tokyo Denki University

In an ad-hoc network where only mobile computers are included, higher mobility and flexibility are supported than an infrastructured network with access points and routers. In this paper, we propose LBSR (Loop Based Source Routing) protocol which is a novel on-demand routing protocol supporting asymmetric wireless communication links. For achieving a route from a source to a destination, LBSR searches multiple loop routes. Here, smaller number of multicasts in a flooding protocol are required than conventional protocols like DSR. We implement LBSR into the kernel of the FreeBSD3.3+PAO environment.

1 背景と目的

近年、コンピュータの小型化と高性能化により、ノート型パーソナルコンピュータやPDAといった携帯型の移動コンピュータが広く利用されるようになってきた。また、ネットワーク技術の発達とWWWをはじめとするネットワークアプリケーションの普及により、ネットワークに接続されるコンピュータの比率が急速に高まっている。このような背景から、移動コンピュータをネットワークに接続して利用するモバイルコンピューティングへの要求が高まっており、IEEE802.11 [1] や HIPERLAN [2] などの無線LAN技術が研究開発されている。会議場やイベント会場におけるネットワークサービスのような移動コンピュータの移動性を活用したネットワークアプリケーションの実行環境として、アクセスポイントを用いないアドホックネットワークが注目されている。アドホックネットワークは、複数の移動ロボット群からなる協調型地雷除去システムのようなアクセスポイントの設置が困難な状況におけるネットワーク構築手段としても有効である。ここでは、ネットワークに専用のルータは接続されておらず、移動コンピュータがメッセージのルーティングを行なう。これによって、移動コン

ピュータのみからなる、より柔軟性の高いコンピュータネットワークを構築することが可能となる。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として、AODV [3] や DSDV [4] などが提案されている。これらのプロトコルでは、すべての移動コンピュータの無線信号到達範囲の大きさは同一かつ一定であると仮定している。すなわち、2台の移動コンピュータ間の接続はすべて双方向であるという前提に基づいた設計がなされている。しかし、無線通信に用いられる電磁波は、反射、回折、透過を行なうため、無線信号到達範囲の大きさはすべての移動コンピュータで同一であるとは限らない。また、各移動コンピュータの無線信号到達範囲の大きさも一定とは言えず、時間とともに変化する。そのため、2台の移動コンピュータ間に片方向接続が存在する場合がある。本論文では、ネットワーク内に複数の片方向接続が存在する場合においても、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経路検出可能なルーティングプロトコルLBSR (Loop Based Source Routing) を提案する。また、FreeBSD3.3+PAOへの実装について述べる。

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なうためには、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、以下の2つに分類される。

[プロアクティブ型]

通信要求の有無に関わらず、各移動コンピュータが他のすべての移動コンピュータへ至る経路の次ホップをルーティングテーブルに保持する方法であり、DSDV [4] がこれにあたる。この方法では、経路情報が常時最新に維持されているため、通信開始時に時間オーバーヘッドを必要としない。しかし、ネットワークポロジの変化をただちにすべての移動コンピュータのルーティングテーブルに反映させる必要があるため、多数のメッセージを交換しなければならない。

[リアクティブ型]

移動コンピュータの位置が経時的に変化するアドホックネットワークにおいては、ネットワークポロジの変化を常時ルーティングテーブルに反映させる通信オーバーヘッドが大きい。そこで、通信開始時に送信元移動コンピュータが送信先移動コンピュータまでの通信経路を探索し、検出された経路を用いてメッセージをソースルーティングする方法がオンデマンド型である。通信要求が発生してからデータの配送が開始されるまでの遅延が大きいという問題点があるが、AODV [3] や DSR [5] をはじめとして、現在までに研究開発された多くのプロトコルが本手法を用いている。

一方、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移動コンピュータ間の接続が必ずしも双方向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ることを仮定している。そこで、本論文では、アドホックネットワーク内に片方向接続が複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。片方向の接続もメッセージの配送経路に用いることによって、経路検出可能性を高めることができる。オンデマンド型経路探索において、片方向接続を用いない従来のルーティングプロトコルと片方向接続も用いる提案プロトコルとの相違は以下の通りである。

[片方向接続を用いないルーティング (図 1)]

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から、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。しかし、片方向接続を用いるプロトコルに比べ、経路検出の成功確率が低い。

[片方向接続を用いるルーティング (図 2)]

それぞれの移動コンピュータが送信する無線信号の到達範囲は同一でなく、また一定でもないが仮定する方法である。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。経路は送信先で検出されるため、この経路を用いてメッセージをソースルーティングするためには、経路情報を送信元に伝えるための送信先から送信元までの経路を検出することが必要である。

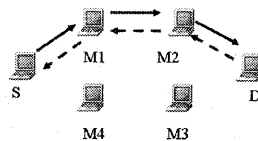


図 1: 片方向接続を用いないルーティング

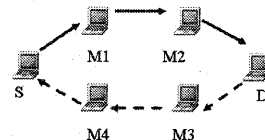


図 2: 片方向接続を用いるルーティング

3 従来手法

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。フラッディングとは、message diffusion protocol [6] 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 の経路探索アルゴリズムを示す。

[DSR による経路探索 (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 の無線到達信号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。

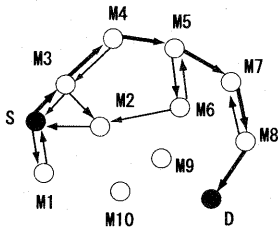


図 3: $R_{S \rightarrow D}$ の探索

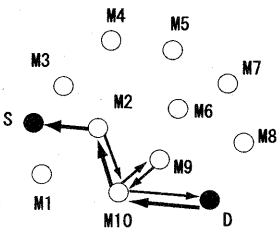


図 4: $R_{D \rightarrow S}$ の探索

4 提案プロトコル

4.1 LBSR(図 5)

移動コンピュータ間のすべての接続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。 $R_{S \rightarrow D}$ を反転させた $R_{D \rightarrow S}$ 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、片方向接続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。 S から D までの経路 $R_{S \rightarrow D}$ を S が得るためには、 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 に戻るループ経路を探索する途中で検出される 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利用することにより、マルチキャストの利用を削減し、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。

LBSR では、経路探索時に2種類のメッセージ RREQ

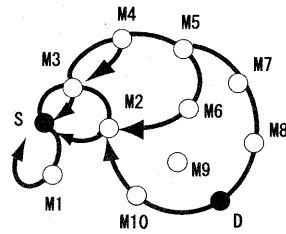


図 5: ループ経路構築

と LOOP を用いる。RREQ 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。LOOP には、 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。LOOP 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 へ至る経路を通知する。また、各移動コンピュータ M_i は、2種類の経路キャッシュを持つ。RREQ キャッシュには、送信元 S から M_i までの経路(アドレスのシーケンス)が格納される。また、LOOP キャッシュには、 M_i から S までの経路が格納される。RREQ キャッシュ、LOOP キャッシュへの経路追加は、それぞれ RREQ メッセージ、LOOP メッセージの受信時になされる。LOOP メッセージは、LOOP キャッシュの経路情報を用いてユニキャストされる。

[LBSR による経路探索]

1. S 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 M_i に RREQ をブロードキャストする。
2. M_i または D が RREQ を受信したならば、以下のいずれかの処理を行ってから、RREQ に含まれる経路を RREQ キャッシュに保存する。
 - 2-1. RREQ キャッシュが空であるならば、ただちに、RREQ を自身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ブロードキャストする。ただし、RREQ をブロードキャストする前に自身のアドレスを RREQ に含まれるアドレスシーケンスの末尾に加える。
 - 2-2. M_i の RREQ キャッシュが空でないならば、LOOP キャッシュに含まれる経路に沿ってユニキャストで RREQ を送信する。 M_i は RREQ をユニキャストする前に自身のアドレスを RREQ に含まれるアドレスシーケンスの末尾に加える。LOOP キャッシュが空である場合には、LOOP を受信して LOOP キャッシュに経路が格納されるまで待つ。このとき、RREQ は LOOP にビジーバックされる。
 - 2-3. D の RREQ キャッシュが空でないならば、この RREQ を破棄する。
3. S が D をアドレスシーケンスに含めない RREQ を受信したならば、RREQ に含まれるアドレスシーケンスを用いてソースルーティングされる LOOP をユニキャストで送信する。

- M_i が LOOP を受信したならば、これに含まれるアドレスシーケンスから S への経路を取得し、これを LOOP キャッシュに保存する。そして、LOOP に含まれるアドレスシーケンスを用いて、LOOP をユニキャストで送信する。
- S が D のアドレスをシーケンスを含む RREQ を受信することで $R_{S \rightarrow D} + R_{D \rightarrow S}$ が得られる。以降、 S は RREQ を受信しても LOOP を送信しない。□

5 LBSR の実装

オンデマンド型ソースルーティングプロトコルである DSR は、FreeBSD3.3+PAO のカーネルに実装されている [7]。ここで、DSR は、ネットワーク層の IP とトランスポート層の TCP、UDP の中間に位置している。受信したメッセージの種類は、IP ヘッダのプロトコルフィールドの値により判別可能であり、これを処理するための関数が呼び出される。また、すべてのメッセージの送信時には、送信先をブロードキャストアドレスとした IP ヘッダと IEEE802.11 ヘッダが DSR によって作成される。つまり、この実装においては、すべての DSR メッセージ (RREQ と RREP に加えて DATA も) はブロードキャストによって配送される。

本論文では、FreeBSD3.3+PAO を対象として、[7] の方法に基づき、IP と TCP、UDP の中間に LBSR を実装した。メッセージの種類は、IP ヘッダのプロトコルフィールドの値によって行なっている。ただし、ブロードキャストとユニキャストの両方を利用している点が [7] の実装とは異なっている。LBSR の構成を図 6 に示す。受信した LBSR メッセージの IP ヘッダに含まれるプロトコルフィールドの値 (RREQ:55、LOOP:57、DATA:60) に対応して、それぞれのメッセージを処理する関数が実行される。RREQ を受信したならば、RREQ キャッシュを参照する。RREQ キャッシュが空であるならば、RREQ のブロードキャスト (フラッディング) を行なう関数 `rreq_forward` が呼び出される。RREQ キャッシュが空でないならば、`loop_output` 関数により、ループ経路に沿ったユニキャストによる RREQ の配送を行なう。いずれの場合も、RREQ に格納された送信元からの経路を RREQ キャッシュに格納する。LOOP を受信したならば、LOOP に格納された経路情報を LOOP キャッシュに格納し、`loop.out` 関数で LOOP をループ経路に沿ってユニキャストで配送する。DATA を受信したならば、`rt.input` 関数でメッセージの宛先アドレスを調べる。自身のアドレスであれば、上位層にデータ部を渡す。それ以外の場合、DATA の経路情報にしたがってユニキャストでメッセージを送信する。上位層からの送信要求を処理する場合は、IP によって `rt.output` 関数、`route.discovery` 関数が起動される。キャッシュに経路が保持されているならば、これを用いてソースルーティングを行なう。保持されていないならば、RREQ のフラッディングによる LBSR 経路探索を開始する。

5.1 メッセージフォーマット

LBSR で用いる 3 種類のメッセージ、RREQ、LOOP、DATA のフォーマットについて説明する。

[RREQ メッセージ (図 7)]

RREQ は、送信元 S から送信先 D へのループ経路を探索するためにフラッディングされるメッセージである。それぞれの経路探索に用いられるメッセージ群を

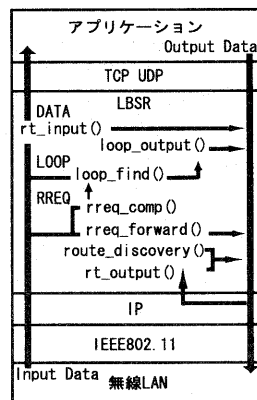


図 6: LBSR の構成

区別するために、 S によって ID (識別子) が与えられる。ネットワーク内では、 S の IP アドレスとこの ID との 2 項組によって、経路探索に関連するメッセージ群を識別している。本実装では、8 ビットの整数値をインクリメンタルかつサイクリックに使用している。また、RREQ を送信する際、IP ヘッダに含まれるプロトコルフィールドには 55 の値が設定され、送信元アドレスには、RREQ を送信する各移動コンピュータ自身の IP アドレス、送信先アドレスには、ブロードキャストアドレスが設定される。RREQ はブロードキャストで送信される。

ID: 経路探索 ID。同一送信元 S からの RREQ を一意に識別するために用いられる。

Length: RREQ メッセージ全体の長さ。byte 単位で表す。

Option: オプション。0 が格納される。

Padding: パディング。常に 0 の値である。

Source address: 送信元 S の IP アドレス。

Target address: 送信先 D の IP アドレス。

Addr_num: この RREQ のアドレスシーケンスに格納されている IP アドレスの数。

Option1,2: オプション。どちらにも 0 が格納される。

Address[Addr_num]: アドレスシーケンス。RREQ が経由した各移動コンピュータの IP アドレスが順に格納される。

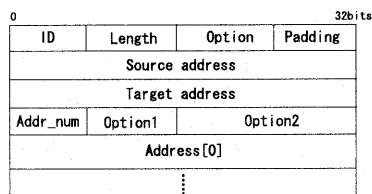


図 7: RREQ と LOOP

[LOOP メッセージ (図 7)]

LOOP は、送信元 S が発見したループ経路を各移動コンピュータの LOOP キャッシュに設定するための

メッセージである。LOOPに含まれるIDは、対応するRREQと同じ値であり、RREQと同様に、送信元Sのアドレスとの2項組によって、関連する一群のメッセージを識別する。LOOPを送信する際、IPヘッダに含まれるプロトコルフィールドには57の値が格納され、送信元アドレスには、LOOPを送信する移動コンピュータのIPアドレス、送信先アドレスには、LOOP内のアドレスシーケンスから取り出された次ホップの移動コンピュータのIPアドレスが設定される。LOOPはユニキャストで送信される。

ID: 経路探索ID。同一送信元SからのLOOPを一意に識別するために用いられる。対応するRREQメッセージのIDと同じ値である。

Length: LOOPメッセージ全体の長さ。byte単位で表わす。

Option: 通常は0である。送信元Sが、送信先Dを含んだループ経路 $R_{S \rightarrow D} + R_{D \rightarrow S}$ を検出した場合に、Sがこのフィールドに1を格納して、LOOPを送信する。このLOOPを受信した各移動コンピュータは、優先的なLOOPとして扱う。

Padding: パディング。常に0である。

Source address: 送信元SのIPアドレス。

Target address: 送信先DのIPアドレス。

Addr_num: ループ経路に属する移動コンピュータの数。

Option1,2: オプション。どちらも0が格納される。

Address[Addr_num]: アドレスシーケンス。検出されたループ経路に属する移動コンピュータのIPアドレスが順に格納される。

[DATAメッセージ(図8)]

DATAは、アプリケーションデータを配送するためのメッセージであり、送信元Sから送信先Dへソールルーティングされる。IPヘッダに含まれるプロトコルフィールドの値は60である。また、IPヘッダの送信元アドレスはDATAを送信する移動コンピュータのIPアドレス、送信先アドレスには、DATA内のアドレスシーケンスから取り出された次ホップの移動コンピュータのIPアドレスが設定される。DATAはユニキャストで送信される。

Protocol: プロトコル番号。上位層に渡すアプリケーションデータのプロトコル番号の値が入る。

Length: DATAメッセージ全体の長さ。byte単位で表す。

Segment Left: ルーティングする経路の残りの移動コンピュータのアドレス数。経路上にある各移動コンピュータは、受信したメッセージを転送する際にこの値を1減らす。

Padding: パディング。常に0である。

Source address: 送信元SのIPアドレス。

Target address: 送信先DのIPアドレス。

Address[Addr_num]: アドレスシーケンス。SからDまでの経路上にある移動コンピュータのIPアドレスが順に格納されている。

5.2 キャッシュ

各移動コンピュータは、メッセージの送信元Sと経路探索IDの2項組のそれぞれに対して、RREQキャッシュとLOOPキャッシュを持つ。

[RREQキャッシュ(図9)]

RREQを受信した場合に送信元Sからこの移動コ

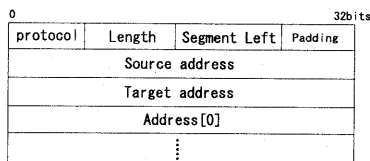


図8: DATA

ンピュータまでRREQが配送された経路が格納されるキャッシュである。ループ経路はLOOPキャッシュとRREQキャッシュによって検出されるが、新たなループ経路が発見された場合や、一定時間使用されなかった経路はRREQキャッシュから削除される。RREQキャッシュの構造は以下の通りである。

Flag: 送信先Dへのループ経路が検出された場合に設定されるフラグ。フラグが設定された後に受信した送信元Sと経路探索IDの等しいRREQパケットは破棄される。

ID: 経路探索ID。

Src: 送信元SのIPアドレス。

Dst: 送信先DのIPアドレス。

Rt_list: RREQに含まれる経路情報が格納される。LOOPを受信した場合、LOOPメッセージの経路情報とRt_listに格納された経路情報とを用いて新規ループ経路の検出を行う。

addr_num: 受信したRREQメッセージが経由した移動コンピュータの数に1を加えた値。

address[addr_num]: 受信したRREQが経由した移動コンピュータのIPアドレスに自身のIPアドレスを追加したアドレスシーケンス。

Next_rt_list: 送信元Sと経路探索IDの2項組が等しいRREQの受信によって検出された複数の経路情報を格納するためにRt_listを連結リストで管理する。このために用いるポインタ。

Next_cache: 送信元Sと経路探索IDの2項組が異なる、すなわち異なる経路探索の情報をリストで管理する。このために用いるポインタ。

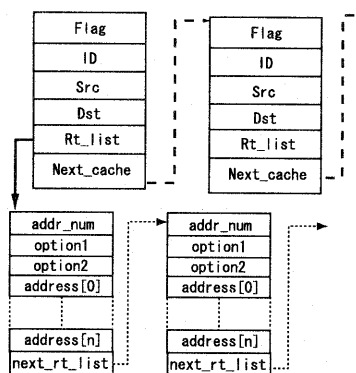


図9: RREQキャッシュ

[LOOP キャッシュ(図 10)]

送信元 S 以外の各移動コンピュータは、IP ヘッダの送信先アドレスが自身の IP アドレスに等しい LOOP を受信したならば、自身から送信元 S までの経路を LOOP キャッシュに保持する。LOOP キャッシュの構造は以下の通りである。

Flag: 送信先 D へのループ経路である場合に設定されるフラグ。フラグが設定された経路と同じ送信元 S と経路探索 ID を持つ LOOP パケットは破棄される。

ID: 経路探索 ID。

Src: 送信元 S の IP アドレス。

Dst: 送信先 D の IP アドレス。

Rt_list: 最初に受信した LOOP メッセージに含まれる経路情報が格納される。2 回目以降に受信した LOOP は、ここに格納された経路に沿って、 S までユニキャスト配送される。

addr_num: 自身から送信元 S までの経路に属する移動コンピュータの数。

address[addr_num]: 自身から送信元 S までのアドレスシーケンス。

Next_cache: 送信元 S と経路探索 ID の 2 項組が異なる、すなわち異なる経路探索の情報をリストで管理する。このために用いるポインタ。

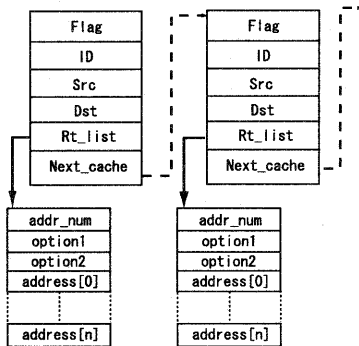


図 10: LOOP キャッシュ

これまでに主要機能の実装を終了し、4 台の移動コンピュータによる経路探索、データ配送の正常動作を確認した。なお、片方向接続を含む場合の動作についても確認済みである。

6 LBSR と DSR の比較

DSR では、片方向接続が存在するアドホックネットワークにおいて、 S から D への経路 $R_{S \rightarrow D}$ を探索する RREQ のフラッディングと D から S への経路 $R_{D \rightarrow S}$ を探索する RREP のフラッディングの双方を必要とする。これに対し、LBSR ではループ経路を探索する RREQ のフラッディングとループ経路の検出を通知する LOOP のユニキャスト配送によって実現される(図 11)。フラッディングにおいては、メッセージがすべての隣接移動コンピュータ間の接続を用いて配送されることから、RREQ の数は同じである。DSR における RREP はフラッディングされるため、RREQ と同数のメッセージを要する。一方、LBSR で

は、RREP とは異なりユニキャスト配送されることから、ループ経路上にある接続のみが用いられる。しかし、同一の接続を用いて複数の LOOP が配送されることがある。実際にネットワーク上で送受される制御メッセージ数、およびプロトコル処理コストについては、実装システムを用いて評価を行なう予定である。

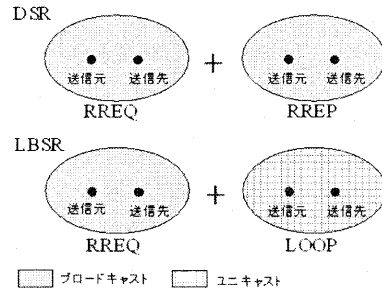


図 11: DSR と LBSR における経路探索メッセージ

7 まとめと今後の課題

本論文では、アドホックネットワーク内に複数の片方向接続が存在する場合においても、経路構築可能な新たなルーティングプロトコル LBSR を提案し FreeBSD3.3+PAO に実装した。本方式では、経路探索時の制御メッセージのフラッディングを削減している。今後は、実装システムを用いた性能評価、キャッシュ管理機構と経路再構築機構についての拡張を行なう。

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); HIPER-LAN," ETSI Functional Specifications (1995).
- [3] Perkins, C.E. and Royer, E.M., "Ad-hoc On-Demand Distance Vector Routing," Proc. of IEEE Workshop on Mobile Computing Systems and Applications, pp. 99-100 (1999).
- [4] Perkins, C.E. and Bhagwat, P., "Highly Dynamic Destination-Sequenced Distance-Vector Routing (DSDV) for Mobile Computers," ACM SIGCOMM'94, pp. 234-244 (1994).
- [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-01.txt (1998).
- [6] Moses, Y. and Roth, G., "On reliable message diffusion." Proc. of the 8th ACM Symposium on Principles of Distributed Computing, pp. 119-128 (1989).
- [7] <http://www.monarch.cs.cmu.edu/dsr-impl.html>.