

中継ノードによる代替ルートの構築と動的なルート切替え手法

三原 龍^{†1} 藤田 茂^{†2}
菅原 研次^{†3} 白鳥 則郎^{†4}

現在、アドホックネットワークにおいて複数のルートを保持するルーティングプロトコルがさかんに開発されている。しかし、既存の複数のルート保持を目的とする DSR を拡張したルーティングプロトコルでは、DSR のルートキャッシュの信頼性が低いため、ルートキャッシュの検索とそれに基づくルート再構築にかかるオーバーヘッドが大きいことや、最低限の予備ルートを保持するための通信中のルート探索におけるブロードキャストのオーバーヘッドが大きいなどの問題が残されたままである。これらの問題点は、アドホックネットワーク構築の大きな制限になっている。本研究では、ノードが分散し頻繁に移動する環境においての利用を想定し、中継ノードと隣接ノードにルート構築に有効な情報を持たせ、これを利用してルートメンテナンスを行うことにより、通信における遅延時間を削減することのできる Child Node Routing (CNR) を提案する。CNR では、各ノードがルートキャッシュ以外に Multi Route Table (全有効ルート情報) と Child Node Table (隣接ノード情報) を使用して、中継ノードが動的にルートメンテナンスを行うことにより、通信遅延時間の削減やデータパケット送受信量の改善、パケット伝達率の向上を実現した。

Child Node Routing Protocol in Ad Hoc Networks

RYOU MIHARA,^{†1} SHIGERU FUJITA,^{†2} KENJI SUGAWARA^{†3}
and NORIO SHIRATORI^{†4}

In recent years, routing protocols which holds multiple routes in Ad Hoc Networks have been developed extensively. However, the existing multipath routing protocols based on the extended DSR have many problems, that they have much overhead of searching route caches and rediscovering routes due to low reliability of route caches of DSR, and also much overhead of broadcast for searching routes to secure alternative routes. In this paper, we propose a new multipath routing protocol Child Node Routing (CNR) for Ad Hoc Networks, based on the assumption that nodes are distributed sparsely and moved frequently. The CNR performs Route Maintenance using information recorded in node tables stored in relay nodes and adjacent nodes in Ad Hoc Network. CNR decreases the overhead of Route Maintenance using the information in Multi Route Table (MRT) and Child Node Table (CNT).

1. はじめに

近年、マルチパスを構築する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がさかんに研究されている^{1),2)}。アドホックネットワークは、無線端末

でのモバイル環境において形成されるネットワークであり、このネットワークに対して、従来様々な分野のネットワークで使用されてきたシングルパスを構築するルーティングアルゴリズムを用いると、アルゴリズムによる緩やかなルートアップデートやトポロジ変化への対応によって、多くのオーバーヘッドが生じてしまうという課題があった。また、シングルパスによる通信を行った場合には、必ず有効なルートが確立されていないため、頻繁におこるトポロジ変化や輻輳によってルートが破損するたびに、ルートメンテナンスを行いルートを再構築しなければならない。このため、ネットワークリソースを有効に活用することができず、単位時間内でのデータパケット送受信量が低下してしまうという問題点がある。モバイル環境下において、シングルパスルーティングアルゴリ

^{†1} 千葉工業大学大学院工学研究科情報工学専攻
Graduate School of Computer Science, Chiba Institute of Technology

^{†2} 千葉工業大学情報科学部情報工学科
Department of Computer Science, Chiba Institute of Technology

^{†3} 千葉工業大学情報科学部情報ネットワーク学科
Department of Information and Network Science, Chiba Institute of Technology

^{†4} 東北大学電気通信研究所/情報科学研究科
Research Institute of Electrical Communication/Graduate School of Information Sciences, Tohoku University

ズムにおけるこのような問題点は多大なタイムロスの原因ともなることが指摘されている^{3),4)}。現在、シングルパスルーティングプロトコルでは、代表的なものに Dynamic Source Routing (DSR)⁵⁾、Ad hoc On-demand Distance Vector (AODV)⁶⁾ などが存在する。この DSR や AODV をベースとした、マルチパスを構築するプロトコルがさかんに研究されている^{7),8)}。これらの多くは、通信を行う際に目的ノードまでのルートを複数構築し、パケットを分割して送信する。これは、モバイル環境において、MAC (Medium Access Control) 層に IEEE802.11 (DCF) を使用した、帯域幅 2Mbps という小規模のネットワーク帯域を有効に活用し、またパケットを分割して送信することによりルートチェックを行い、中継ノード消失などによる通信信頼性の低下を防いでいる。しかし、

- (1) ルートメンテナンスの際に、ベースであるプロトコルのルート修復に時間がかかるという問題点が未解決のままである、
- (2) 複数のルートを構築するためにシングルパスルーティングプロトコルに比べ、多くのルート構築要求パケットが送信される、
- (3) サブルートを保持するために、多くのルート探索・メンテナンスのメッセージがネットワーク全体に送信される、

などの問題点がある。これらの問題点を持つマルチパスルーティングは、シングルパスルーティングに比べて通信の信頼性は向上するものの、タイムロスを軽減できるわけではない。また、街中でのアドホックネットワークや車車間通信などの、広域なフィールド上でノードが分散し頻繁に移動する環境下では、前にあげた問題点 (1) は強く解決が求められる課題である。

本稿では、ノードが分散し頻繁に移動する環境における使用を前提とし、中継ノードを用いてマルチパスルーティングにおける動的なルートメンテナンスを行う Child Node Routing (CNR) を提案する。提案する CNR の特徴として、ルート探索時に Multi Route Table (MRT) と Child Node Table (CNT) という 2 つのテーブルを構築する。これらは、全有効ルート情報 (ルート ID, ソース, ホップ数) と隣接ノード情報 (所属ルート ID, ノード ID) を格納する。これにより、中継ノードは CNT を参照することで、隣接ノードの所属ルート ID および所属ルートのソースを隣接ノードに問い合わせることなく知ることができる。隣接ノードの探索や下位ノードに対してのルート探索を行わないことにより、ルート再構築の時間を短縮しネットワーク遅延を抑制することができる。本

提案の利点がある。

以下では、2 章で関連研究として既存のルーティングプロトコルである DSR および、DSR を拡張したマルチパスルーティングプロトコルである Split Multipath Routing (SMR)⁹⁾ について述べ、その問題点を示す。3 章で先に述べた問題点を解決するための、新しいルーティングにおけるルートメンテナンス方式である CNR について述べる。4 章で提案したルートメンテナンス方式の妥当性を示すために CNR の性能評価について述べる。この中では DSR, SMR との比較実験をすることにより提案方式の利点を示した。

2. 既存プロトコルの問題点

DSR は、on-demand ルーティングを行う。このルーティングアルゴリズムでは、中継ノードが目的ノードまでのルート情報を知らなくてよい、近隣ノードとつねに HELLO メッセージを交換しなくてよいという特徴がある。DSR は、送信ノードが ROUTE REQUEST (RREQ) をブロードキャストして、目的ノードか目的ノードまでのルート情報をキャッシュしている中継ノードからの ROUTE REPLY (RREP) を受け取ることによって通信が開始される。ルートがノード消失や輻輳、遅延などで使用できなくなったとき、中継ノードは自分のルートキャッシュ内に存在するリンクダウンした接続を削除し、ROUTE ERROR (RERR) を送信ノードに送る。RERR を受けた送信ノードは、自分のルートキャッシュを検索し代替ルートを探す。これにより、RREQ をフラディングせずに通信を再開できるという利点がある。代替ルートがあればそれによって通信を再開し、ない場合は RREQ を再びブロードキャストする。

しかし、DSR のルートキャッシュは各ノードが通信中に保持する目的地までのルート情報であり、異なった目的ノードとの通信、使用環境が異なる場合にはルート構築の有効性に乏しい。また、同一の目的ノードまでの通信であっても、再度通信を行う前にトポロジが変化することにより中継ノードがキャッシュ内のノードと同一である可能性は低下し、ルートキャッシュ内の情報だけで通信を再開できる可能性もあわせて低下する¹⁰⁾。ルートメンテナンスにおいて、ルートキャッシュに有効ルートが存在しないときにリンクダウンを発見した場合、送信ノードでルートキャッシュの検索を行い、また有効と思われるルートにパケットを送信してしまうといった動作は、モバイル環境において限界のあるネットワーク帯域のリソースを浪費し、タイムロスの原因となる。

これに対し SMR では、複数のルートを構築し通信を行い、かつルートキャッシュを使用しないことにより、ルートが切断されてから再構築するまでのオーバーヘッドを軽減している。しかし、SMR ではすべてのルートが切断されると再度 RREQ のフラッディングによるルート構築が行われるため、制御パケットの冗長的発生によりネットワークスループットが低下してしまうという問題点がある。本研究では、この SMR と同様のルート構築を行い、さらに CNT と MRT を用いてルート再構築を局所的に行うことによりこの問題を解決する。

3. Child Node Routing

本稿で提案する CNR は、on-demand 型のルーティングを行うプロトコルであり、RREQ によって構築されたルートへの RREP はそれぞれの RREQ 内に記述されているルートソースに基づいて返信される。各ノードは、ルートメンテナンスのために MRT と CNT をルート構築の際に作成する。ルートは最大構築数を 3 とし、すべてのルートを通信に使用する。3 本もしくは 2 本のルートが構築できた場合は、データパケットをそれぞれのルートに分配して送信する。

3.1 ルート構築の方式

本稿では、アドホックネットワークをグラフ $G = (V, E)$ とし、以下のように各要素を定義する。

- V : ノード, $|V|$: ノード数
- E : リンク, $|E|$: リンク数 (ホップ数)
- S : 送信ノード, D : 目的ノード, R : ルート
- $R = (S \rightarrow V_1 \rightarrow V_2 \dots \rightarrow V_n \rightarrow D)$

このとき、ルート R において S と D 以外のノードを中継ノードと呼び、2 つのルート R_a と R_b に対し以下の記法を定義する。

- $R_a \cap R_b = \{w \mid w \text{ は } R_a \text{ と } R_b \text{ での重複中継ノード}\}$
- $|R_a \cap R_b|$: 重複している中継ノードの数

CNR では、中継ノードにおける RREQ の受信、および目的ノードによるルート選択を下記の制約に従って行う。中継ノードにおける制約として、中継ノード V_i は、1 度目に受信した RREQ の S からのホップ数 ($|E_1(S, V_i)|$) に対し 2 度目以降に受信した RREQ については、

$$|E_n(S, V_i)| < |E_1(S, V_i)| \tag{1}$$

を満たす場合に限り RREQ を受信する。SMR の制約では、 $|E_n(S, V_i)| \leq |E_1(S, V_i)|$ とされていることにより RREQ の冗長性が問題となっていたが、本稿の制約によりこの問題は改善される。しかし、これに

より構築できるルート数が SMR に比べ減少してしまうという新たな問題点がある。この問題は、SMR では許されていないルート中のノード重複を許すことで改善する。ノードの重複は、アドホックネットワークの平均ホップ数が約 4 または 5 ホップ⁹⁾ であるのをふまえ、

$$|R_a \cap R_b| \leq 1 \tag{2}$$

を満たす場合に限り許可する。これにより、たとえば送信ノードの隣接ノードが 1 つしかないときにもルートを複数構築できるようになる。ただし、2 つ以上のノード重複を許可してしまうと、1 つのノード重複時に比べて、重複ノードが消失しリンクが切れてしまうことによるルートの安定性が低くなってしまうので、これは許可しない。

D が RREQ を受信した場合は、以上の制約に基づいて次のようなプライマリルート R_1 、セカンダリルート R_2 、サードルート R_3 を決定する。

- R_1 : S と D を結ぶ最小ホップ数のルート
- R_2 : R_1 に対して式 (2) を満たす、 S と D を結ぶ最小ホップ数のルート
- R_3 : R_1 および R_2 に対して式 (2) を満たす、 S と D を結ぶ最小ホップ数のルート

3.2 ルート構築アルゴリズム

CNR では、3.1 節で示した式 (1) と式 (2) を満たし、かつ RREQ を受信した際に隣接ノード情報を格納する点が SMR とは異なる。そこで、SMR のルート構築アルゴリズムに対して、中継ノードは隣接ノード情報を RREQ のアドレスシーケンスから読み取りキャッシュする点、中継ノードは式 (1) を満たすように RREQ のフラッディングを行う点、重複ノード数が式 (2) を満たしている場合にはルート候補とする点、という 3 点の変更を行う。図 1 は、中継ノードを N_i としたときの CNR におけるルート構築概念図である。

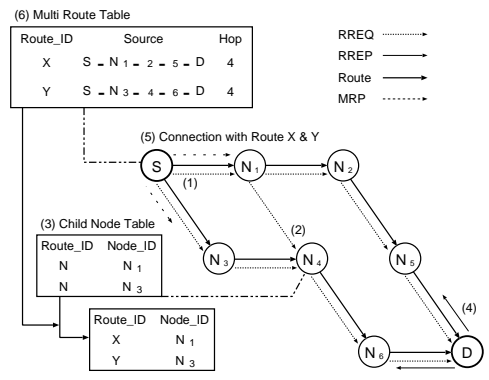


図 1 ルート構築
Fig. 1 Route discovery.

以下に、送信ノード S 、中継ノード N_i 、目的ノード D での RREQ の処理およびルート構築のアルゴリズムを記す。このとき、 req_flag_i は N_i における RREQ の受信判別チェックフラグとし、 rep_flag_i は S における RREP の受信判別チェックフラグとする。

0. 各ノードは次の各変数を初期化する。

$req_flag_i = false$, $rep_flag_i = false$.

1. S は自身からなる RREQ のアドレスシーケンスを含む RREQ を通信可能範囲に存在するすべてのノードにブロードキャストする。

2. 中継ノード N_i が RREQ を受信した場合。

- $req_flag_i = false$ の場合、アドレスシーケンスに格納されている末尾のアドレスをキャッシュし、ルート ID (初期状態: $NULL[N]$) を付加して CNT を作成する。また、自身のアドレスをアドレスシーケンスの末尾に追加し、RREQ をブロードキャストする。このとき、 $req_flag_i = true$ とする。
- $req_flag_i = true$ の場合、3.1 節の式 (1) を満たしている場合は RREQ 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て RREQ をブロードキャストする。式 (1) を満たしていない場合は破棄するが、破棄する場合もしない場合も CNT にアドレスシーケンスに格納されている末尾のアドレスをキャッシュし、ルート ID を付加する。

3. D が RREQ を受信した場合。

- $req_flag_i = false$ の場合、アドレスシーケンスの末尾に自身のアドレスを追加し、 S から D までのアドレスシーケンスをキャッシュする (これを R_1 とする)。このとき、 $req_flag_i = true$ とする。 R_1 をキャッシュした後、 R_2 と R_3 となるルートが構築できるまで、ランダム時間 $([0, 1]s)$ 待機する。待機時間が過ぎたら、キャッシュしてあるすべてのルートのアドレスシーケンス (ルートソース) を、逆参照しながら S に RREP を送信する。
- $req_flag_i = true$ の場合、3.1 節の式 (2) を満たし、 R_2 か R_3 としてルートが構築できる場合は、アドレスシーケンスの末尾に自身のアドレスを追加し、 S から D までのアドレスをキャッシュする。待機時間を過ぎてから受信した RREQ でルートが構築できる場合は、アドレスシーケンスに自身のアドレスを追加して、RREP を S に送信する。

4. S が RREP を受信した場合。

- $rep_flag_i = false$ の場合、RREP にあるルートを使用して通信を開始する。このとき、

$rep_flag_i = true$ とする。また、通信に使用するルートのルートソースをキャッシュし、ルート ID を付加して MRT を作成する。その後、 S は通信に使用しているルートの各中継ノードに MRT を送信する。

- $rep_flag_i = true$ の場合、RREP にあるルートのルートソースを MRT に格納し、通信に使用する。 S は通信に使用しているルートの各中継ノードに MRT を送信する。

通信開始後、MRT を受信した中継ノードは、キャッシュしてあるノードが所属するルートを検索し、ノードのルート ID を所属しているルートのルート ID に変更する。

2 本以上のルートが構築された場合は、ルートのホップ数の逆数の比でパケットを分配する。ルート R_a と R_b が構築された場合のパケットの分配比は、

$$R_a : R_b = |E(R_b)| : |E(R_a)| \quad (3)$$

となる。これにより、 S により送信される全データパケット数が x のとき、SMR ではネットワーク全体での通信回数が、

$$x(|E(R_a)| + |E(R_b)|)/2 \quad (4)$$

となるのに対し、CNR では、

$$2x|E(R_a)||E(R_b)|/(|E(R_a)| + |E(R_b)|) \quad (5)$$

となり、より少ない通信回数でデータパケットを S から D まで送信することができる。これにより、単位時間内でのパケット送信量を増加させることが可能になる。

3.3 ルートメンテナンス

CNR では、主に中継ノードにより代替ルートを構築し、ルートの切断が生じた場合にルート変更を行う。しかし、代替ルートが構築できなかった場合は、 S によってルートの削減もしくは再構築が行われる。

3.3.1 代替ルートの構築

通信が開始されると使用ルート上に存在する D に隣接していない中継ノードは、HELLO メッセージを用いて、他ルートもしくは、近隣に移動してきた新しいノードを中継ノードとする代替ルートの発見を行う。 D はルート上に存在しないノードのみから HELLO メッセージを受信する。このとき、中継ノードの重複はルート構築方式に従うものとする。本稿では、HELLO メッセージの Time To Live (TTL) を実験的に 3 とする。これは、中継ノードにより構築される代替ルートのホップ数をアドホックネットワークの平均ホップ数内に制限するとともに、多量の冗長的な HELLO メッセージの発生を防ぐことを目的とする。図 2 は代替ルート構築の概念図である。

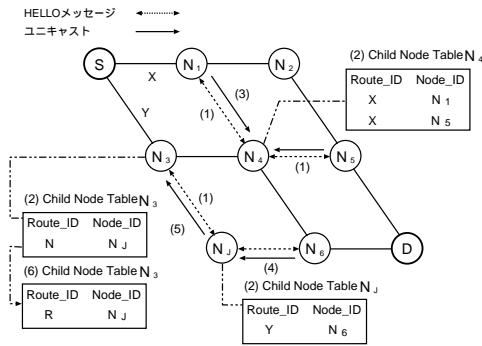


図 2 代替ルートの構築
Fig. 2 Substitution-route discovery.

以下に、中継ノードを N_i, N_j 自身の所属するルート以外に所属する隣接ノードを NR_n , HELLO パケットによって発見された隣接ノードを NR_j としたときの、代替ルート構築のアルゴリズムを示す。

1. 通信が開始されると、 N_i は TTL を 3 として HELLO メッセージをブロードキャストする。
2. N_i の隣接ノードが HELLO メッセージを受信した場合、
 - NR_n が HELLO メッセージを受信した場合、ユニキャストにより N_i に自身のアドレスを送信し、CNT に N_i の情報を格納する。
 - NR_j が HELLO メッセージを受信した場合、CNT に N_1 の情報を格納し、TTL を 1 減らして HELLO メッセージをブロードキャストする。
3. D もしくは N_{i+n} が HELLO メッセージを受信した場合、ユニキャストにより、代替ルートとして使用できることを NR_j に送信する。
4. NR_j がユニキャストによる HELLO メッセージを受信した場合、
 - D もしくは N_{i+n} から受信した場合は、HELLO メッセージのブロードキャストを停止し、一定間隔で D もしくは N_{i+n} とチェックパケットとして HELLO メッセージをユニキャストする。CNT に格納されている D もしくは N_{i+n} のルート ID を r に変更し、 NR_{j-1} に、代替ルートとして使用できることをユニキャストする。
 - NR_{j+1} から受信した場合、HELLO メッセージのブロードキャストを停止し、一定間隔で NR_{j+1} と HELLO メッセージをユニキャストする。CNT に格納されている NR_{j+1} のルート ID を r に変更し、 NR_{j-1} に、代替ルートとして使用できることをユニキャストする。
5. N_i が HELLO メッセージを受信した場合、

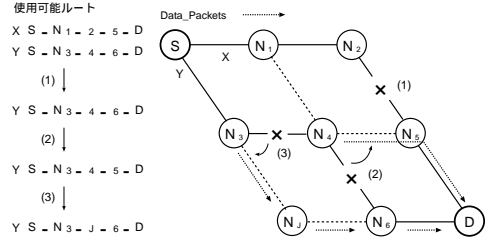


図 3 再構築されたルート
Fig. 3 Reconstructed route.

- NR_n から HELLO メッセージを受信した場合は、CNT に NR_n のアドレスを追加し、ユニキャストにより一定間隔で HELLO を交換する。
- NR_j から HELLO メッセージを受信した場合は、CNT に NR_j のアドレスを追加する。また、ユニキャストによる代替ルート構築のメッセージだった場合は、ルート ID を r に変更し、一定間隔で NR_j と HELLO メッセージをユニキャストし、代替ルートを保持する。

このとき、たとえば、図 2 のルート Y を代替ルートとして N_1 から N_4 へとルートを変更することは、重複ノードが多いためできない。しかし、 N_3 が消失した場合は、 N_1 から N_4 へとリンクすることで代替ルートを構築する。このように代替ルートを中継ノードによって構築することにより、ルートの切替えにかかる時間を、CNT の検索時間のみとすることができる。

3.3.2 ルート切替え

代替ルートを保持している中継ノードはリンク切断が発生した場合にルート切替えを行う。使用する代替ルートは、構築時にホップ数を基準に並べることにより、どのルートを使用するかといったテーブル内の検索において時間がかかるとはならない。図 3 は、代替ルートによるルート再構築の概念図であり、ルート切断にともなう使用可能ルートの推移を表した図である。

以下に、中継ノードを N_i, N_j 自身の所属するルート以外に所属する隣接ノードを NR_n , HELLO パケットによって発見された隣接ノードを NR_j としたときの、ルート切替えのアルゴリズムを示す。このとき、ROUTE UPDATE (RUPD) パケットは、アドレスシーケンスを代替ルートのルートソースに変更し、 S に使用可能ルートのルートソース変更を告知するパケットとする。

1. N_i は、 $(N_i \rightarrow N_{i+1})$ におけるリンク切断を発見した場合、 NR_n もしくは NR_j が CNT に存在するかどうかを検索する。
 - NR_n が存在した場合、ルート構築アルゴリズム

において有効なルートとして使用できる場合は、データパケットのアドレスシーケンスを変更し代替ルートとして使用する。また、 S に RUPD を送信する。

- Nr_j が存在した場合、アドレスシーケンスの N_{i+1} 以降のアドレスを削除し、 Nr_j にデータパケットを送信する。
2. Nr_j がデータパケットを受信した場合、アドレスシーケンスに自身のアドレスを追加して、 Nr_{j+1} にデータパケットを送信する。
 3. D は、 Nr_j がアドレスシーケンスに含まれているデータパケットを受信すると、新しいルートソースを逆参照し RUPD を S に送信する。
 4. S は RUPD を受信すると MRT の内容を変更し MRP を全ルートに送信する。
 5. MRP を受信した N_i は、CNT 中に MRP に含まれないノードを保持していた場合、その情報を削除する。

代替ルートが存在しない場合は、RERR が S に送信されることにより新しくルート構築が行われる。

中継ノード N_i が ($N_i \rightarrow N_{i+1}$) におけるリンク切断を発見したとき、DSR では、各中継ノードがキャッシュを検索する時間を T_c 、キャッシュしている隣接ノード数を x 、隣接ノードへの問合せ時間を T_x 、 S により新しくルートを構築するのにかかる時間を T_r とすると、キャッシュしているノードが D までの情報を保持していない場合、

$$Tr + x(Tc + Tx)|V(S, N_i)| \quad (6)$$

といった遅延時間が生じる。本稿で前提としている環境では、ノードは分散し頻りに移動するので、一度キャッシュした情報を更新しない DSR では、式 (6) で求められる遅延時間がリンク切断のたびにかけられると考えられる。これに対し、SMR ではキャッシュを使用しないので、ルート再構築にかかる時間を同じく T_r とすると、この時間しかかからない。CNR では、CNT に代替ルートが存在すればキャッシュの検索時間のみ、存在しない場合は SMR と同様の処理を行うため、 T_c もしくは T_r という遅延時間になる。これにより、CNR では、DSR、SMR に比べ、リンク切断から通信を再開するまでの遅延時間を減少させることができる。

4. シミュレーションによる評価と考察

シミュレーションでは、CNR の評価を行うために、DSR と SMR との比較を行った。

4.1 シミュレーション環境

シミュレーションは、Network Simulator2 (ns-2.26) を用い、 $1000\text{m} \times 1000\text{m}$ のフィールド内に 20 台のモバイルノードをランダムに配置した。ノードの移動速度は 0m/s から 10m/s の範囲内でランダムに設定しランダムウェイポイントモデルに従って移動する。シミュレーション時間は 200s 、各ノードは IEEE802.11 を使用して通信を行うものとし、通信可能範囲は 250m として帯域幅は 2Mbps としている。また、データパケットのサイズは 512bytes とし、UDP により 0.25s 間隔でパケットを送信する。

このとき、ノードが移動し次に移動を始めるまでのポーズタイムを 0s から 200s まで変化させたときの通信遅延時間、データパケット送受信量、パケット伝達率について測定した。コネクションは 1 つとして、各ポーズタイムにつきそれぞれ 50 パターンのノード配置とコネクションを用意した。

4.2 評価基準

シミュレーションでは、前にあげた通信遅延時間、データパケット送受信量、パケット伝達率について評価した。以下にその定義を記す。

• 通信遅延時間

通信遅延時間は、ルートが切断されてから再構築が行われ再び通信が開始されるまでの時間とする。これは、本稿における提案手法の有効性を示す重要な項目であり、CNT・MRT を利用した中継ノードによるルート切替え、代替ルートの構築による遅延時間の削減を評価する。

• データパケット送受信量

データパケットは、RREQ や RREP などの制御パケットを除いたパケットの送受信量であり、送受信量が多い方が単位時間内での通信においてより多くの情報交換や転送が行われているものとする。

• パケット伝達率

パケット伝達率は、送信したパケット数と受信されたパケット数の割合である。これは、伝達率が高いほどドロップされるパケット数が少ないと考えられ、通信における信頼性が高いといえる。

4.3 評価結果

シミュレーション結果を図 4、図 5、図 6 に示す。各グラフの横軸はノードの移動におけるポーズタイムを表し、縦軸はそれぞれ、通信遅延時間、データパケット送受信量、パケット伝達率を表す。すべての結果において、ノードの移動がない場合、すなわちポーズタイムが 200秒 のときは、それぞれほぼ同等の値に収束

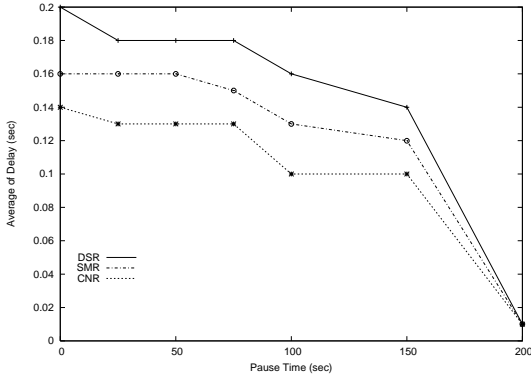


図 4 通信遅延時間の平均
Fig. 4 Average of delay.

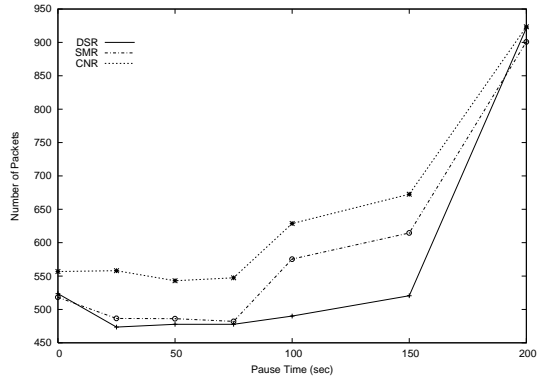


図 5 データパケット送受信量
Fig. 5 Number of packets.

した。

4.3.1 通信遅延時間

ノードがまったく移動しない場合は、遅延時間は DSR, SMR, CNR とも 0.01s で収束した。しかし、ポーズタイムが小さくなるほど、遅延は大きくなっている。DSR では、シングルパスルーティングのために、ポーズタイムが 0s のときは、CNR が約 15 回、SMR が約 22 回というのに比べて約 86 回とルート切断が多い。これにより、通信中に約 86 回の RREQ のフラッディングを行うため、切断から再構築までの遅延が多く発生し、遅延時間は約 $80(T_r + x(T_c + T_x))V(S, N_i)$ となる。これにより最大遅延が 0.2s となり最小遅延時間の 20 倍の遅延が生じてしまう。これに対し SMR では、3.3.2 項で述べたように、ルートキャッシュを使用せずにルート再構築を行うことにより、全体の遅延時間が DSR に比べ約 18.75% 削減できている。しかし、ノードが移動することにより 2 本のルートが切断されてしまうと RREQ をフラッディングすることにより、ポーズタイムが 0s から 50s と小さいときには遅延時間は変化しない。CNR では、代替ルートの構築および中継ノードによるルート切替えを行うことにより、1 度のルート再構築にかかる遅延時間は T_c もしくは T_r となり、全体の遅延時間では DSR に比べ約 37.5%, SMR に比べ約 23.1% 削減することができた。

DSR ではルートが単一であり、かつルートキャッシュにおけるオーバーヘッドが大きい、SMR では複数ルートにおける通信を行うが、ルート切断時に多くの制御パケットが冗長的に発生してしまう、という問題点があった。しかし、CNR ではノードが少ない状況において、隣接ノードとの情報交換や中継ノードによるルート構築・切替え、制御パケットの局所的な発生、送信ノードのルート保持における負荷の軽減、短時間での通信の再開が実現できていると考えられる。

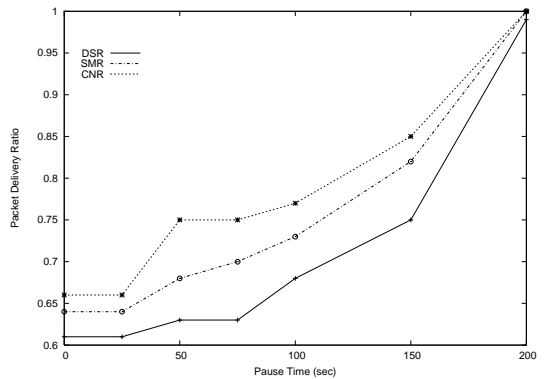


図 6 パケット伝達率
Fig. 6 Packet delivery ratio.

4.3.2 データパケット送受信量

制御パケットを除いたデータパケットの送受信量はノードが移動しない場合は、200s の通信時間中どのプロトコルでも約 900 パケット送受信できた。DSR では、4.3.1 項でも述べたように、多くのルート切断にともない全体的に低い値となっている。SMR はポーズタイムが 0s から 75s までは、ノードの移動にともなうルート切断によりルートが 1 本になってしまうため、DSR と同等の値となっている。これは、2 本作成したルート中どちらかが短時間で切断されてしまうことや、2 本作成するための制御パケットの増加によるものと考えられる。しかし、ポーズタイムが 100s 以上になると、ルートが切断されにくくなるため、DSR に比べて多くのパケットが送受信できるようになる。これにより、平均値では約 43 パケット、21.5 Kbyte 以上多くのデータを送受信できている。CNR では、3.2 節で述べたように式 (3) を用いてパケットを割り振ることにより、式 (4) および式 (5) から分かるように、SMR に比べパケットの送受信できる量を増加するこ

とができる。また、ルートが切断されても RREQ のフラッシングをすることなく通信を再開することができるため、DSR に比べて送受信できるパケット量が約 95.27 パケット (47.635 Kbyte), SMR に比べて約 52.27 パケット (26.135 Kbyte) 増加している。ポーズタイムが 0s のときは、DSR では 523.62, SMR では 518.15, CNR では 556.79 となった。これらの結果により、CNR ではルートメンテナンスが局所的に行われており、また単位時間において多くのパケットが送受信できている。

単位時間におけるデータパケット送受信量はノードが頻繁に移動する環境下では重要な要素である。CNR は、DSR, SMR に比べ送受信量が高く、移動の激しい環境下に最も適しており、4.3.1 項の結果とあわせると、遅延時間の削減とデータパケット送受信量の増加により、既存プロトコルに比べネットワークスループットが向上したと考えられる。

4.3.3 パケット伝達率

ノードが移動しなければ、どのプロトコルもほぼ 100%送信したパケットが目的のノードで受信されていることが分かる。しかし、ノードが移動を始めることにより、ルート切断によってパケットがドロップされてしまうために、パケットの伝達率は低下する。DSR では、つねにノードが移動する環境では約 60%しかパケットが届いていない。これに対し、SMR では 64%のパケットが受信されており、これはルートを 2 本使用することによりルート切断に対処しているためと考えられる。CNR は、68%のパケット伝達率となった。各プロトコルにおいて、ポーズタイム 0s から 25s まで伝達率に変化がないのは、ルートが切断されてしまうことによってドロップするパケット数に変化が生じないためと考えられる。これは通信遅延時間、データパケット送受信量においても同様のことがいえる。全体の平均値では、DSR が 67%, SMR が 73%, CNR が 77%となった。この結果により、CNR はパケットドロップの発生が少なく、信頼性の高い通信が可能であるといえる。

また、データパケット送受信量と関連づけて見ると、CNR がより多くのパケットを正確に目的とするノードに届けていることが分かる。これを「通信性能」とし、「データパケット送受信量 × パケット伝達率」で表すと、DSR が 360.17, SMR が 423.81, CNR は 487.28 となり DSR, SMR と比べ最も通信性能の高いプロトコルといえることができる。

4.3.4 オーバヘッド

通信遅延時間やデータパケット送受信量、パケット

伝達率において、CNR は DSR や SMR に比べて結果が向上したことは、今までに述べた。これを裏付ける結果として、ルート構築およびメンテナンスにおけるオーバヘッドとして、各プロトコルにおける RREQ の発生回数および、これにともなうネットワーク全体でブロードキャストされる RREQ のパケット数を、最もルート切断が多いポーズタイムが 0s の場合での平均値で比較すると、

- DSR: RREQ 発生数 86, RREQ パケット数 334
- SMR: RREQ 発生数 22, RREQ パケット数 659
- CNR: RREQ 発生数 15, RREQ パケット数 308

となった。これにより、シングルパスルーティングはルート切断が起こりやすいのに比べ、マルチパスルーティングはルートが切断されにくいことが分かる。しかし、本稿の冒頭でも述べたように、SMR などのマルチパスルーティングは複数の経路を作成するだけであり、ルートのメンテナンスを行わないため、ネットワーク中で冗長な RREQ パケットが送受信される。これは、結果からも分かるとおり、ルートキャッシュを使用してルートメンテナンスを行う DSR に対し、RREQ 発生数が DSR の約 1/4 であるにもかかわらず、SMR の RREQ パケット数は 2 倍である。これに対し、本提案手法である CNR は、ルート構築の際に中継ノードに制約を設けたことで、冗長な RREQ を抑制できたと考えられる。また、CNT と MRT を用いたルートメンテナンスにより、SMR に比べてルート切断の回数も削減できたと考えられる。以上より、CNR はルート構築およびルートメンテナンスにおけるオーバヘッドが最も少ないプロトコルだといえる。

4.4 ノードの高機能化にともなうトレードオフ

CNR では、CNT や MRT を作成し中継ノードでのルートメンテナンスを行うため、ノードの高度化にもなってハードウェア面での要求の増加や、HELLO パケットによる通信コストによって消費電力などの負荷の増加が考えられる。以下では、4.3 節において示した結果などを基に、これらの負荷の増加について考察する。

4.4.1 通信・計算コストによる消費電力

CNR では、4.3.4 項で示した結果から分かるとおり、まず RREQ パケットは既存のプロトコルよりも削減される。また、HELLO メッセージを使用することにより、制御パケットが増加するが、すべてのノードにおいて HELLO メッセージの交換が行われる table-driven 型ルーティング^{11),12)} に比べ、メッセージ数は少ない。また、CNT と MRT を利用したメンテナンスを行うことにより通信遅延時間が削減されているこ

とから、各テーブルの検索・計算コストも DSR のルートキャッシュ程度と考えることができるため、CNR の通信および計算による消費電力は、実用可能範囲と考えることができる。

4.4.2 中継ノード高度化によるメモリサイズ

CNR では、中継ノードの高度化により保持するデータ量が、on-demand 型ルーティングプロトコルと比べ増加する。しかし、増加するデータ量は、MRT に格納する最高 3 本までのルートソースであり、隣接ノード情報は DSR のルートキャッシュと同等としているため、大幅なメモリサイズの増加を必要としない。ルートが長く、またノードが密集している環境で使用する場合は、隣接ノード数も増加するため中継ノードが保持するデータ量は増加する。このような場合では、無線端末のメモリサイズ以上を必要とする可能性はあるが、本稿で前提としているノードが分散し頻繁に移動する環境においては、既存の無線端末のメモリサイズに対して格納するデータ量が著しく大きくなってしまふことはないと考えられる。

4.4.1 項の消費電力およびメモリサイズの問題に関して、ノード数が多くまた密集しているような環境では、CNR は実用可能なレベルではない。このような環境における使用のためには改善すべき点が多く、課題とその解決方針については次章で述べる。

5. おわりに

本研究では、ルートメンテナンスに注目し、ルート構築時に CNT・MRT を作成し、また HELLO パケットによる隣接ノード探索を行うことにより、局所的なルートメンテナンスを行う CNR を提案した。CNR は、ユースケースとしてノード数が少なく分散している状況を想定しており、シミュレーションの結果より、DSR、SMR に比べ遅延時間の削減によるネットワークスループットの向上、データパケット送受信量の増加による単位時間内でのデータ通信量の増加、パケット伝達率の向上を実現したといえる。これにより、CNR では災害時などにおいて、分散された救助隊や被害者における緊急ネットワークの構築において信頼性の高い通信を実現することができると考えられる。

しかし、提案した CNR において考慮していないノードが密集している環境においての使用では、中継ノードや隣接ノード間で、多量の冗長的な HELLO メッセージが送受信されてしまい、ネットワークスループットが低下してしまうと考えられる。また、HELLO メッセージがいくつかのルート間で干渉してしまう可能性も高く、中継ノードが代替ルートを構築するより

送信ノードによるルート再構築によるオーバーヘッドのほうが小さくなると考えられる。よって、今後の課題として以下の項目が考えられる。

- (1) ノードのグループ化による制御パケットの減少
- (2) 密・疎状態による最適なルーティング切替え

- 隣接ノード数を指標とした密度算出
- グループ内外における通信手法選択

グループ化の研究^{13),14)}はさかんに行われているため、今後の大きな課題は(2)の2つの項目になる。1つ目の密度算出は、RREQ のフラッディングの際に、破棄する RREQ を受信した場合でも隣接していることだけを返信することにより密度を把握し、通信中はルート切断やノード消失を検知した時点で HELLO メッセージをブロードキャストすることにより、ノード密度の算出が行えると考えられる。2つ目のグループ内外での通信は、グループ内は一定のノード密度を保っているため HELLO メッセージを使用せず、グループどうしの通信において、互いのゲートウェイノード間によって HELLO メッセージを交換しあうことにより、冗長的に HELLO メッセージが発生することを防ぐことができると考えられる。このようにしていくことで、ノード密度に対応したアドホックネットワークのためのルーティングが実現できると考えられる。

参考文献

- 1) Park, V.D. and Corson, M.S.: A Highly Adaptive Distributed Routing Algorithm for Mobile Wireless Networks, *Proc. INFOCOM*, pp.1405–1413 (1997).
- 2) Raju, J. and Garcis-Luna-Aceves, J.J.: A New Approach to On-demand Loop-Free Multipath Routing, *Proc. IEEE 13 International Conference on Computer Communications and Networks (ICCCN)*, pp.522–527 (1999).
- 3) Broch, J., Maltz, D.A., Jonson, D.B., Hu, Y.-C. and Jetcheva, J.: A Performance Comparison of Multi-Hop Wireless Ad Hoc Network Routing Protocols, *Proc. ACM/IEEE Mobile Computing and Networking (MOBICOM)*, pp.85–97 (1998).
- 4) Johansson, P., Larsson, T., Hedman, N., Mielczarek, B. and Degermark, M.: Scenario-based Performance Analysis of Routing Protocols for Mobile Ad-hoc Networks, *Proc. ACM/IEEE Mobile Computing and Networking (MOBICOM)*, pp.195–206 (1999).
- 5) Johnson, D.B. and Maltz, D.A.: The Dynamic Source Routing in Ad Hoc Wireless Networks, Internet-Draft, draft-ietf-manet-dsr-

- 09.txt (2003).
- 6) Perkins, C.E., Royer, E.M. and Das, S.R.: Ad Hoc On-Demand Distance Vector (AODV) Routing, Internet-Draft, draft-ietf-manet-aodv-08.txt (2001).
 - 7) Lee, S.-J. and Gerla, M.: AODV-BR: Backup Routing in Ad hoc Networks, *Proc. IEEE Wireless Communications and Networking Conference (WCNC)* (2000).
 - 8) Nasipuri, A. and Das, S.R.: On-Demand Multipath Routing for Mobile Ad Hoc Networks, *Proc. IEEE 13 International Conference on Computer Communications and Networks (ICCCN)*, pp.64-70 (1999).
 - 9) Lee, S.J. and Gerla, M.: Split Multipath Routing With Maximally Disjoint Paths in Ad hoc Networks, *Proc. IEEE 13th International Conference on Computer Communications and Networks (ICCCN)*, pp.3201-3205 (2001).
 - 10) 溝口和寛, 古庄伸一, 北須賀輝明, 中西恒夫, 福田 晁: 再接続時間を短縮したアドホックネットワークルーティングプロトコルの提案と評価, *信学技報*, NS2003-10-16, Vol.103, No.10, pp.1-4 (2003).
 - 11) Adjih, C., Clausen, T. and Jacquet, P.: Optimized Link State Routing Protocol, Internet-Draft, draft-ietf-manet-olsr-08.txt (2003).
 - 12) Ogier, R.G., Lewis, M.G., F.L.T. and Bellur, B.: Topology Dissemination Based on Reverse-Path Forwarding (TBRPF), Internet-Draft, draft-ietf-manet-tbrpf-08.txt (2003).
 - 13) 西澤正稔, 萩野浩明, 原 隆浩, 塚本昌彦, 西尾章治郎: 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).
 - 14) 太田 賢, 町田基宏, 大辻清太, 杉村利明: アドホックネットワーク上のコミュニティのためのグループ通信プロトコル, *情報処理学会論文誌*, Vol.42, No.7, pp.1801-1809 (2001).

(平成 16 年 4 月 5 日受付)

(平成 16 年 10 月 4 日採録)



三原 龍

1981 年生。2003 年千葉工業大学工学部情報工学科卒業。同大学院博士前期課程情報工学専攻在学中。スケーラブルな高信頼性マルチキャストプロトコルについて研究。現在、移動端末における通信に興味を持ち、アドホックネットワークにおけるルーティングの研究に従事。



藤田 茂 (正会員)

1968 年生。1997 年千葉工業大学大学院博士後期課程情報工学専攻期間満了退学。同年同大学助手(工学部情報工学科)。現在、同大学講師(情報科学部情報工学科)。博士(工学)。エージェント、マルチエージェントシステム、センマンティックウェブ、分散処理、知能ソフトウェア工学、人工知能応用システム等の研究に従事。1995 年度電子情報通信学会学術奨励賞受賞。電子情報通信学会、人工知能学会各会員。



菅原 研次 (正会員)

1950 年生。1973 年東北大学工学部通信工学科卒業。同年富士通株式会社入社。1975 年東北大学大学院工学研究科修士課程入学。1980 年同大学院博士課程中退。同年千葉工業大学電子工学科助手。現在千葉工業大学情報ネットワーク学科教授。工学博士。分散人工知能、知識型設計方法論、CAI、ソフトウェア再利用、やわらかいシステム、サイバー工学に興味を持つ。1992 年日本工業教育協会功績賞、1994 年情報処理学会山下記念研究賞受賞。電子情報通信学会、人工知能学会、ソフトウェア学会、IEEE 各会員。



白鳥 則郎 (正会員)

1946 年生。1977 年東北大学大学院博士課程修了。1984 年同大学助教授(電気通信研究所)。1990 年同大学教授(工学部情報工学科)。1993 年同大学教授(電気通信研究所)。情報通信システム、ソフトウェア開発環境、ヒューマンインタフェースの研究に従事。1993 年本会マルチメディアと分散処理研究会主査。本会 25 周年記念論文賞受賞。本会副会長、IEEE フェロー、電子情報通信学会、人工知能学会各会員。