

## MANETにおける高信頼クラスタベース ルーティングアルゴリズムの提案と実装・評価

鳴海寛之<sup>†</sup> 白石陽<sup>††</sup> 高橋修<sup>††</sup>

高移動・高通信頻度のモバイルアドホックネットワークにおける従来のリアクティブ型プロトコルを用いた通信では、通信経路が長経路となった場合、中継ノードの移動による経路切断や経路構築要求メッセージのネットワーク全体へのフラディングが悪影響を及ぼしパケット到達率が低下することが分かっている。一般的に、ノード移動への対応能力を向上させる手法としてクラスタリングが用いられるが、従来のアルゴリズムには長経路通信が発生するという問題がある。そこで本研究では、長経路通信自体の発生を抑制するというアプローチのもと、クラスタリングおよびルーティングのためのアルゴリズムを提案し、通信信頼性の向上を図る。さらに、本論文では提案方式を代表的なリアクティブ型プロトコルであるDSR上に実装し、計算機シミュレーションによりその有効性を検証評価する。その結果、ノードが高速移動する(最大ノード移動速度 15.0m/s)場合、パケット到達率が既存プロトコルであるDSRよりも約43%、CBRPよりも約19%向上した。また、データパケット到達遅延時間においても、提案方式は既存プロトコルよりもノード移動速度の変化の影響を受けにくいことが分かった。これらの結果から、本提案方式は従来よりも高い通信信頼性を有することが示された。

### Proposal and Evaluation of Reliable Cluster-Based Routing Algorithm for MANET

HIROYUKI NARUMI<sup>†</sup> YOH SHIRAIISHI<sup>††</sup>  
OSAMU TAKAHASHI<sup>††</sup>

Existing reactive routing protocols in Mobile Ad-hoc Network (MANET) decrease the packet delivery ratio in the high mobility and traffic environments. This is because the temporary congestion occurs by the flooding of control packets for the route establishment and by the route break in the long route communication. Generally, clustering is an effective method to improve the capacity to deal with the node mobility. However, the existing algorithms have the above problem in the long route communication. In this paper, we propose clustering and routing algorithms that rein

the emergence of the long route communication. Furthermore, we implement our proposed method on DSR that is known as representative reactive routing protocol and evaluate the effectiveness by the computer simulation. As a result, in the high mobility environment (15.0m/s maximum node velocity, pose time 0.0s), the proposal method improves PDR (Packet Delivery Ratio) about 43% better than DSR and about 19% better than CBRP. In addition, for packet arrival delay, we show the proposed method is less subject to node velocity than existing protocols. By these results, we show our proposed method provides the high communication reliability than the existing routing algorithms.

#### 1. はじめに

近年、無線通信技術の発達と移動無線端末(ノード)の小型化・高性能化に伴い、モバイルアドホックネットワーク(MANET: Mobile Ad-hoc Network)に関する研究が活発に行われている。MANETは、基地局などの既存インフラに依存せずにノードが即席で形成する自律分散型のネットワークである。MANETではマルチホップ通信と呼ばれる通信方式を採用しており、通信経路上に位置するノード(中継ノード)がデータを転送することによってエンド-エンド間の通信が実現される。MANETにおけるルーティングプロトコルとして、DSR(Dynamic Source Routing)<sup>[1]</sup>などに代表されるリアクティブ型プロトコルがある。これらのプロトコルでは、データ通信時に利用する通信経路が長経路であるほど、経路構築・修復処理やパケット転送処理によってネットワークに負荷がかかり、パケット到達率やルーティングオーバーヘッドなどの通信信頼性が低下することが分かっている。さらに、高移動頻度環境において複数のノードが同時に通信を行うような状況では、長経路通信がもたらす悪影響によって輻輳が生じやすくなり、著しく通信信頼性が低下する。そのため、高移動・高通信頻度のネットワークにおいて安定した通信を実現するためには、長経路通信の発生を抑制することが効果的である。高移動頻度環境への対応能力を向上させる代表的な手法としては、クラスタリングを用いたクラスタベースルーティングが挙げられる。一般的なクラスタベースルーティングでは、クラスタリングによって経路構築・修復処理が効率化され、ルーティングオーバーヘッドの削減やパケット到達率の向上などが実現できる。そのため、非クラスタベースのアドホックルーティングと比べてノード移動に強いという特徴がある。しかし、長経路通信が発生するという問題は依然として残されたままである。

そこで本稿では、長経路通信を複数の短経路通信に分割するというアプローチによって、従来よりも高移動・高通信頻度のネットワークにおいて高い通信信頼性を提供する新たなクラスタベースルーティングアルゴリズムを提案する。さらに提案方式をDSR上に実装し、計算機シミュレーションによって既存プロトコルと比較評価を行うことにより、提案方式の有効性を示す。

#### 2. MANETにおける長経路通信

高移動頻度のネットワークにおける長経路通信の信頼性を評価するため、2点間固定ビットレート通信のシミュレーションをネットワークシミュレータ ns-2<sup>[2]</sup>を用いて行った。シミュレーションパラメータは表1のように設定し、ルーティングプロトコルには代表的なアドホックルー

<sup>†</sup> 公立はこだて未来大学大学院  
Graduate School of Future University Hakodate

<sup>††</sup> 公立はこだて未来大学  
Future University Hakodate

ティングプロトコル（非クラスタベース）である DSR を用いた。そして、ランダムに選択した 2 ノード間で固定ビットレート通信を 1200 秒間発生させた。

表 1 シミュレーションパラメータ (1)

Simulation Parameters	
ネットワークサイズ	1000mx 1000m
ノード数	50, 100, 150, 200
シミュレーション時間	1200秒
ノード移動速度	2.5 m/s
ノード移動モデル	Random Waypoint
ポーズタイム	0.0秒
無線伝搬範囲	半径250m
パケットサイズ	512 Bytes
送信キューのサイズ	500/パケット
パケット送信間隔	10 packets/s

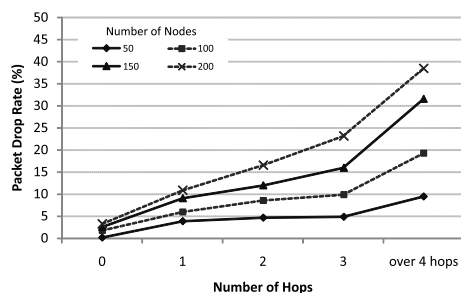


図 1 ホップ数に対するパケット到達率

シミュレーション結果から得られたホップ数とデータパケットのドロップ率の関係を図 1 に示す。このシミュレーションにより、エンド・エンド間の通信が長経路である場合ほど、パケットドロップ率が増加するという結果が得られた。特に、200 ノードの場合は 4 ホップ以上の長経路通信のドロップ率が約 40%にも達することが分かった。このように、MANET では長経路通信時ほどパケットのドロップ率が高くなり、さらにノード数の多いネットワークほどさらにドロップ率が増加する傾向にある。これは 2つの要因が考えられる。1つは、ルーティングプロトコルが経路構築を行う際、エンド・エンド間が長経路であるほど経路構築のための制御メッセージが広範囲にフラッディングされる可能性が高く、コリジョンが発生しやすくなり正常な通信ができなくなってしまうためである。もう 1つは、高移動頻度の環境ではパケットを転送する際に中継ノードが移動することによって経路切断が発生し、次の転送者にパケットを転送できずにドロ

ップしてしまうためである。

このように、高移動頻度の MANET において安定した通信を行うためには、長経路通信の発生を抑制することが重要な解決要素となる。

### 3. 関連研究

#### 3.1 クラスタベースルーティング

高移動頻度のネットワーク環境下において通信信頼性を向上させるための代表的な手法として、クラスタリングが広く用いられている[2][11]-[14][16]。クラスタリングを用いたルーティング手法をクラスタベースルーティングと呼び、これまでに CBRP (Cluster Based Routing Protocol)[4]をはじめとする様々なクラスタベースのルーティングプロトコルが提案されている[5]-[8][15]。これらのプロトコルで用いられる一般的なクラスタリングの概念図を図 2 に示す。MANET におけるクラスタリングでは、物理的に近いノード同士がクラスタヘッド (CH: Cluster Head) を中心としてクラスタを形成する。CH は自クラスタに現在所属しているノードの情報 (ノードリスト) を集中管理する。異なるクラスタの境界に位置するノードは境界ノード (GN: Gateway Node) として、クラスタを跨ぐ通信時の橋渡しとしての役割を担う。CH 及び GN 以外のクラスタに参加しているノードはメンバノード (MN: Member Node) と呼ぶ。

ネットワークを複数のクラスタに分割することにより、経路探索時に発生する制御メッセージ (RREQ: Route Request<sup>a</sup>及び RREP: Route Reply<sup>b</sup>)を大幅に抑制することが可能となる。既存のアドホックルーティングプロトコルにおける経路探索では、1つの経路を構築するために多くのノードが関与する必要があり、RREQ がネットワークの広範囲にフラッディングされてしまう。これに対し、クラスタベースルーティングではクラスタを形成することによりネットワーク上の各 CH にのみ RREQ を送信するだけで経路を構築できる。図 3 は一般的なクラスタベースルーティングにおける経路構築方法を示したものである。送信元ノード S が発行した RREQ はまず自クラスタの CH に転送される。各 CH は自身が管理するノードリストを参照し、宛先ノードが含まれていなければ RREQ を破棄もしくは隣接するクラスタに転送する。ノードリストに宛先ノードが存在する場合は、宛先ノードに RREQ を転送する。RREQ を受信した宛先ノード D は、RREQ が転送されてきた経路の逆経路を使用して送信元ノード S に RREP を返信する。このように、任意のノードに対する経路を素早く形成することができるため、クラスタベースのルーティングプロトコルは既存のアドホックルーティングプロトコルに比べてノード移動への対応能力が高いということが知られている。

a) Route Request: 経路探索の際に送信元ノードを基点として周囲に送信される経路要求メッセージ。  
 b) Route Reply: 経路探索処理によって構築された宛先ノードまでの経路が付加された経路応答メッセージ。

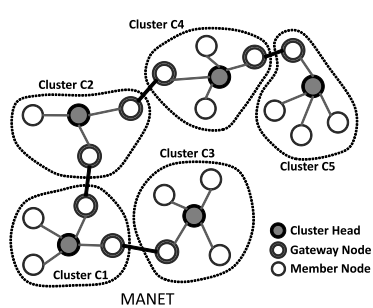


図 2 MANETにおけるクラスタリング

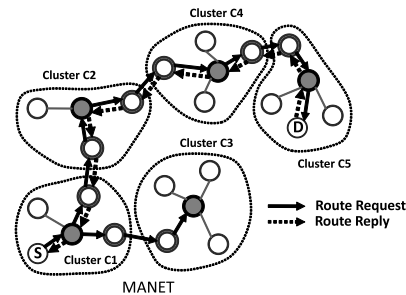


図 3 一般的なクラスタベースルーティングにおける経路探索

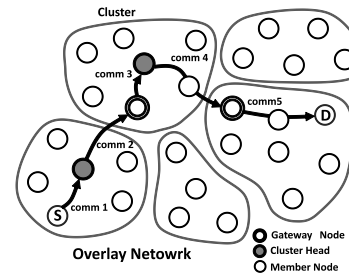
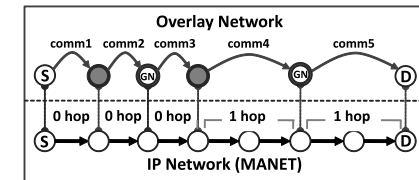
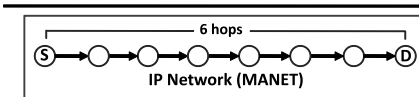


図 4 提案方式



(a) 提案方式による通信分割



(b) 従来のルーティング

図 5 従来のルーティングと提案方式

しかし、送信元ノードと宛先ノードとの物理的距離が大きい場合などは、従来のクラスタベースルーティングアルゴリズムにおいても構築される通信経路は多くの中継ノードを介す長経路となってしまふ。図 3 の例では、送信元ノード S と宛先ノード D の間には 10 ホップの長経路が構築される。このような経路を利用してパケットを送信した場合、中継ノードの移動による経路切断が発生しパケット到達率が低下するなどといった通信信頼性の低下を引き起こしてしまう。CBRP など一部の階層型ルーティングアルゴリズムには既存のリアクティブ型プロトコルが備えるような経路修復処理と同等の機能を備えているものもあるが、長経路通信の発生自体を抑制するというアプローチはみられない。そこで本研究では、長経路通信の発生自体を抑制し、新鮮な短経路を組み合わせることで通信を行うアプローチを提案し、高移動・高通信頻度のネットワークにおける通信信頼性の向上を図る。

## 4. 提案方式

### 4.1 提案方式の基本構成

本提案方式はオーバレイネットワーク上にクラスタを形成し、任意の宛先ノードが所属するクラスタまで中継ノードがローカルにキャッシュした短経路を組み合わせることでパケットを転送する(図 4) [10]。このとき、クラスタ間の境界に位置している一時的な境界ノード (Gateway Node) を宛先としてそれぞれ指定することにより、長距離通信であったとしてもクラスタ間の転送毎に通信が区切られるため、長経路通信の発生を抑制することが可能となる。図 5 の例では、従来のルーティングプロトコルでは 6 ホップ必要とする通信(図 5(b))が、提案方式では 0, 0, 0, 1, 1 ホップの 5 つの短経路通信の組み合わせによって構成される(図 5(a)) ことを表している。

### 4.2 クラスタリング

提案方式では、MANET 上にオーバレイネットワークを形成し、クラスタリングを行う[9]。各クラスタにはクラスタ ID と呼ばれる固有の識別子が割り当てられる。各クラスタの CH は定期的に自クラスタ ID を含んだビーコンメッセージをクラスタ内にブロードキャストすることで、各メンバーノードは自身が所属しているクラスタを認識する。従来のクラスタリング同様、CH は自クラスタに所属している全ノードの情報 (ノードリスト) を管理する(図 6)。

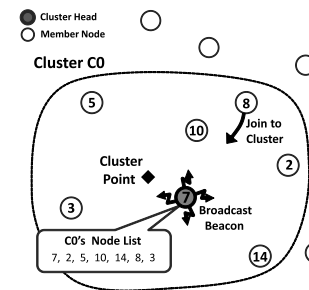


図 6 提案方式におけるクラスタリング

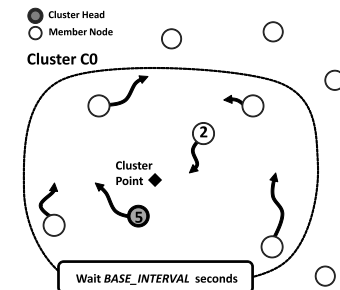


図 7 CH 交替処理 (1)

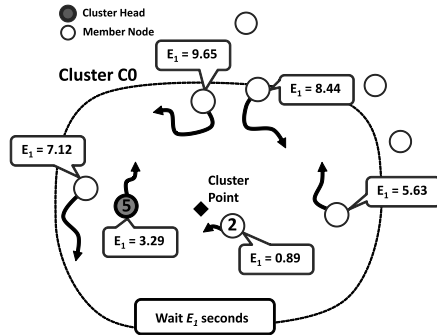


図 8 CH 交替処理 (2)

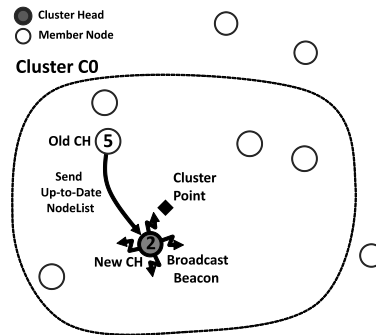


図 9 CH 交替処理 (3)

CH が移動する場合、新たな CH が動的に選出される。提案方式では、CH の交代処理を効率的に行うために、GPS などの位置情報を用いてクラスタの中心点を各クラスタに割り当てる。このクラスタの中心点をクラスタポイントと呼ぶ。そして各ノードがそれぞれのクラスタポイントに対する“近さ”を表す評価値を算出することにより、最適な CH を選出する。クラスタ数を  $n$ 、ノードの平均移動速度を  $v_{avg}$ 、クラスタポイントとの物理距離を  $d$  としたとき、評価値  $E_n$  は次の計算式(1)によって求められる。  $C$  は定数である。

$$E_n = C \times d + \sqrt{v_{avg}} \log_{10}(v_{avg} + 1) \quad (1)$$

各ノードは、自 CH からのビーコンを受信すると、一定時間 ( $BASE\_INTERVAL$  秒) の待ち時間の後に自クラスタのクラスタポイントに対する評価値を算出する (図 7)。そしてさらに求めた評価値である  $E_n$  秒の待ち時間を各ノードが消化する (図 8)。最終的に、最も評価値の低い (距離と速度を考慮した上でクラスタポイントに最も近い) ノードが最初に待ち時間を消化することになり、そのノードが新 CH としてビーコンの送信を行う (図 9)。旧 CH は新 CH からのビーコンを受信すると、自身がこれまで管理していたノードリストを新 CH に転送し、自身はメンバーノードとなる。これらの処理を定期的に行うことによって、CH を含む全ノードが移動する環境下においても、クラスタを維持することが可能となる。

### 4.3 クラスタ間の隣接情報の収集

本提案方式では CH がクラスタ内にブロードキャストするビーコンメッセージを利用して各クラスタの隣接情報を収集し、ネットワーク上に存在するクラスタ同士の相対的な位置を把握することでルーティング時の経路決定に活用する。CH は自身のクラスタ ID をセットしたビーコンをブロードキャストする。異なるクラスタからのビーコンを受信したノードは、自身がクラスタ間の境界にいと判断し、境界ノード (Gateway Node) としてビーコン送信元 CH に対して

“Neighbor Feedback” メッセージを送信する。

例えば、ノード A がクラスタ C2 に属していた場合、A が C1 の CH に返信する Neighbor Feedback には、これまでローカルに保存していた隣接クラスタ情報に加え、“C2 は C1 に隣接している”という旨を隣接クラスタ情報としてセットする。この Feedback によって、各クラスタの CH は自クラスタに隣接するクラスタの情報を収集することができる (図 10)。そして、クラスタ C1 の CH が次回ビーコンをクラスタ内にブロードキャストする際は、ビーコンに“C1 に隣接するクラスタは C2”であるという情報を追加し、さらにこれまで収集した最新の隣接クラスタ情報をセットして送信する。この処理を各 CH が行う事により、最終的にネットワークに参加している全ノードが全てのクラスタの隣接クラスタ情報を知ることができる (図 11)。

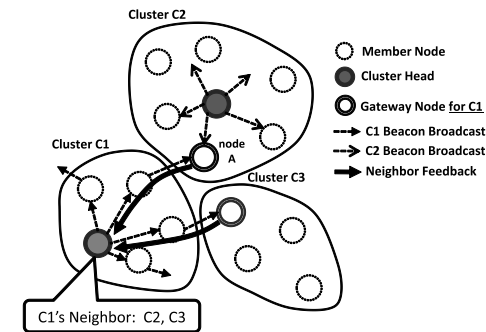


図 10 Neighbor Feedback による隣接クラスタ情報の収集 (1)

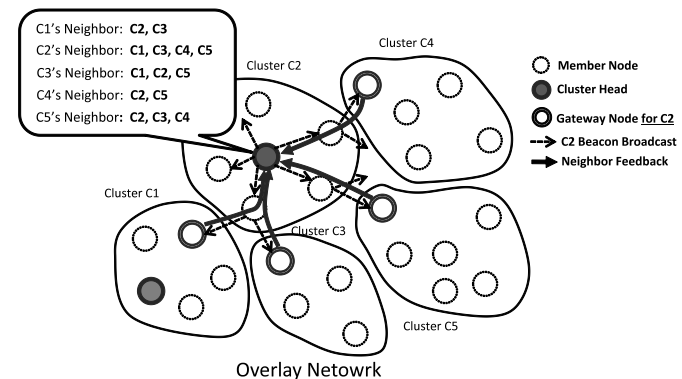


図 11 Neighbor Feedback による隣接クラスタ情報の収集 (2)

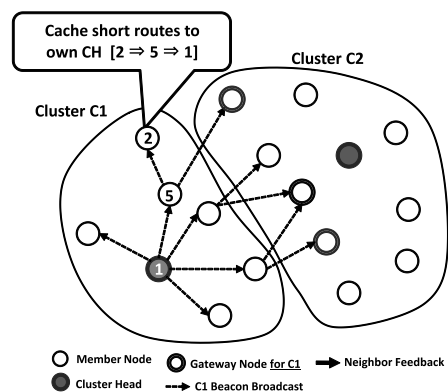


図 12 短経路の生成 (1)

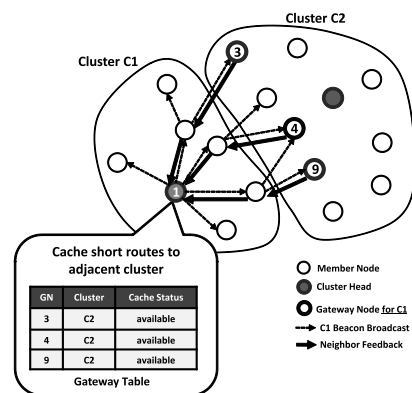


図 13 短経路の生成 (2)

#### 4.4 短経路の生成

効率的な通信分割を行うため、提案方式では隣接クラスタ情報の収集時に短経路を生成する。短経路の生成方法は 2 種類あり、それぞれビーコン受信時と Neighbor Feedback 受信時に生成される。図 12 および図 13 は短経路の生成の例を示したものである。クラスタ C1 の CH であるノード 1 は定期的にビーコンをブロードキャストする。そしてビーコンはクラスタ内に伝搬されていく。このとき、ビーコンを受信したクラスタ C1 の各ノードは、それぞれビーコンが転送されてきた経路の逆経路をローカルに保存する。この短経路をルーティングの際に使用する (図 12)。また、クラスタ C2 に所属しているノード 3, 4, 9 もクラスタ C1 からのビーコンを受信する。するとノード 3, 4, 9 は自身が境界ノードであると判断し CH に Neighbor Feedback メッセージを返信する。そして Neighbor Feedback を受信したノード 1 は、Feedback が転送されてきた経路の逆経路をローカルに保存する。これで隣接クラスタへの短経路を生成したことになる (図 13)。ただし、CH は複数の境界ノードからの Neighbor Feedback を受信する可能性がある。そこで CH は Gateway Table というテーブルに隣接クラスタへの短経路を保存して管理し、ルーティングの際にそれらを効率的に利用する。

##### 4.4.1 短経路の維持・更新

CH が Gateway Table 内で管理する短経路の情報は定期的に更新される。CH は定期的に境界ノードに対して PING リクエストを送信し、境界ノードの生存状態を確認するとともに、保持していた短経路を更新する (図 14)。もし PING 応答が返信されてきた場合はこれまで保持していた短経路 (自身から境界ノードまでの経路) の情報を更新する。一定時間 PING 応答が返って来なかった場合は、短経路のキャッシュが利用不可であるという目印をテーブル内のエントリーに付加する。これにより、ルーティングの際に利用可能な短経路のみを使用することができ、

より信頼性の高い転送が可能となる。

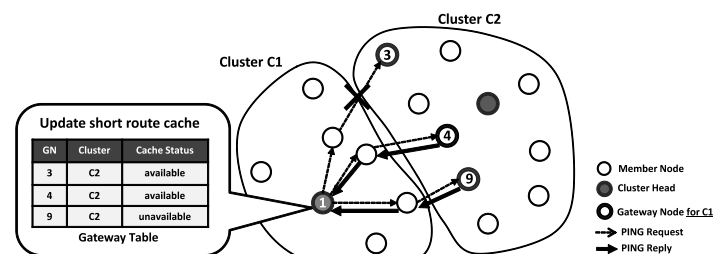


図 14 短経路の維持・更新

#### 4.5 ルーティング

Neighbor Feedback によって収集した各クラスタの隣接クラスタ情報を元に、各ノードは自クラスタから現在存在する任意のクラスタへの経路を決定することができる。各ノードは、自クラスタを基点として隣接クラスタ情報を整理し、クラスタマップと呼ばれるルーティングテーブルを生成する。クラスタマップを参照することで、自クラスタからみた各クラスタとの相対的な距離を求め、任意のクラスタに属する宛先ノードにデータパケットを送信するためにどのクラスタを経由すべきかを判断することができる。図 15 は、クラスタ C0 の CH が収集した隣接クラスタ情報をもとにクラスタマップを生成した例を示している。クラスタマップを参照することで、クラスタ C6 に属しているノードにデータパケットを送りたい場合にはクラスタ C3 を経由してパケットを送ればよいことが分かる。

クラスタマップを用いて宛先ノードが現在所属しているクラスタまでの経路を決定すると、短経路を組み合わせたパケット転送が可能となる。あるノードが特定の宛先ノードに対してクラスタを跨いでデータパケットを送信する際は、ビーコン受信時にキャッシュした経路を用いて自クラスタの CH までパケットを転送する。CH は次クラスタの境界ノードに対してデータパケットを転送する。宛先ノードが所属するクラスタに到達するまで、CH と境界ノード間で転送が繰り返される。最終的に、宛先ノードのクラスタの境界ノードが実際の宛先ノードにパケットを転送し、ルーティングが終了する (図 4, 図 5)。

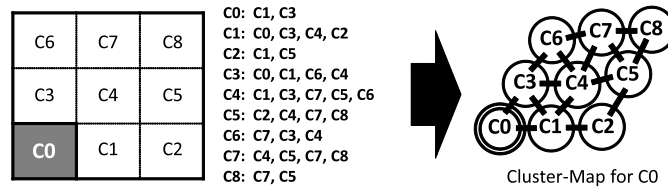


図 15 隣接クラスタ情報からのクラスタマップ生成

表 2 シミュレーションパラメータ (2)

ネットワークパラメータ		CBRPの設定	
ネットワークサイズ	1000m x 1000m	Packet Salvaging	有効
ノード数	100	プロミスキャスモード	有効
シミュレーション時間	300秒	提案方式の設定	
通信時間	180秒	クラスタ数	4
ノード最低移動速度 $V_{min}$	1.0m/s	クラスタポイント (x, y)	(300, 300), (700, 300), (300, 700), (700, 700)
ノード最大移動速度 $V_{max}$	1.0, 2.5, 5.0, 7.5, 10.0, 12.5, 15.0 m/s	BASE_INTERVAL	2秒(非定常状態), 7秒(定常状態)
ポーズタイム	0.0秒	C	0.03
ノード移動モデル	Random Waypoint	NeighborFeedback返信可能数	2
無線伝播半径	250m	PING送信間隔	3秒
DSRの設定			
Packet Salvaging	有効		
プロミスキャスモード	有効		
Ring Zero Search	有効		
Flow State オプション	無効		

## 5. 実装と評価

### 5.1 シミュレーション条件

今回、提案方式を代表的なリアクティブ型プロトコル（非クラスタベース）である DSR の上位に実装し、DSR と代表的なクラスタベースルーティングプロトコルである CBRP との性能比較を ns-2 により行った。主なシミュレーションパラメータを表 2 に示す。DSR の Flow State 拡張は RFC4728<sup>[1]</sup>においてオプション機能として定義されているが、本稿では使用しない。移動モデルには Random Waypoint モデルを使用する。各ノードはランダムな目的座標に  $V_{min}$  から  $V_{max}$  のランダムな移動速度でランダム時間直進し、また新たな目的座標を新たに選択し移動する、という動作を繰り返す。このとき、ポーズタイムは 0.0 秒としているため、各ノードは停止すること無く絶えず移動し続ける。つまり、本実験において構築されるネットワークは高移動頻度のネットワークであるといえる。ネットワーク上のノードのうち、初期位置が各クラスタポイントに最も近いノードが各クラスタの初期 CH として動作する。評価値  $E_n$  の算出には式(1)を使用する。

以上のシミュレーション条件のもと、同時に 3 ペアの固定ビットレート通信（パケットサイズ 100Bytes, パケット送信間隔 10 packets/s）をランダムに発生させ、30 秒間通信する。そし

て 30 秒経過後は新たに 3 ペアをランダムに選択し、30 秒間通信する、という通信パターンでシミュレーションを行った。これらの通信はシミュレーション開始 60 秒後から開始し、シミュレーション終了 60 秒前まで継続して発生させる。

### 5.2 結果と考察

ノード移動速度に対するパケット到達率を図 16 に示す。最大移動速度が 15.0m/s とノードが高速に移動する環境下でも提案方式は 90%に近いパケット到達率を維持していることが分かる。しかし、CBRP は最大移動速度が上昇するにつれてパケット到達率が徐々に低下していくことが分かる。DSR の場合、パケット到達率は急激に低下し、最大移動速度が 7.5m/s 以降の値は下げ止まっていることが分かる。結果として提案方式のパケット到達率は、最大移動速度が 15.0m/s のとき、CBRP と比較して最大で約 19%の向上がみられた。DSR と比較して最大で約 43%の向上がみられた。

#### (1) 提案方式と DSR の比較

ノード移動速度が高速になるにつれて DSR のパケット到達率が急激に低下する要因として、以下の 2 点が考えられる。

- i) 長経路通信時の中継ノード移動によるドロップ
- ii) 長経路初期構築・再構築のための経路探索による輻輳の発生

今回のシミュレーションでは、常に 3 本の固定ビットレート通信が発生する。このとき、ある通信が長経路通信となり、通信中に経路切断が発生すると、経路切断を検出したノードはパケットの Salvaging を試みる。ここで Salvaging に失敗すると、送信元ノードは再度 Route Discovery を実行して宛先ノードまでの経路を構築し直さなければならない。こうした処理が複数の通信において発生すると、ネットワーク全体に経路構築のための制御メッセージがフラッディングされ、ネットワーク全体が輻輳状態に陥り易くなる。そして一度このような状態が発生してしまうと、玉突き衝突のように延々と輻輳状態が続いてしまう。この間に転送しようとしたほとんどのパケットは、有効な経路を構築できなかった（タイムアウトした）として、またはコリジョンの発生によりパケットを正しく転送できなかったとして、パケットをドロップしてしまう。特に、ノードが高速に移動する場合ほど、経路は構築しにくくなるため、経路探索の再実行の回数が増加してしまい、結果として輻輳状態が長く続き、より多くのパケットをドロップするという悪循環に陥ってしまう。このように、高通信頻度のネットワークにおいては、長経路の構築処理がネットワーク全体の通信信頼性を低下させる大きな脅威となる。次に、ノード移動速度に対するルーティングオーバーヘッドを図 17 に示す。また、ノード移動速度に対する通信経路の平均ホップ数を図 18 に示す。移動速度が高速になるにつれて、DSR のルーティングオーバーヘッドが大幅に増加することが分かる。これは、平均ホップ数の値が示すように、長経路の構築や修復のための制御メッセージのフラッディングが行われることで輻輳が発生し、結果として大量の制御メッセージがネットワーク上に氾濫してしまったことが原因と考えられる。オーバーヘッドの増加量から、DSR ではノード移動速度が 5.0m/s 以上になると輻輳が発生しやすくなっていると考えられる。

これに対し、本提案方式はビーコン送信および Neighbor Feedback 返信の仕組みによってあ

あらかじめキャッシュした新鮮な短経路を利用するため、1回の転送の際に使用する経路の平均ホップ数が常に1ホップ以下となっている。つまり、経路切断が発生した場合でも他の有効な短経路に切り替えるだけでよく、新規の経路探索処理はほとんど実行されないため、DSR と比べてルーティングオーバーヘッドを大幅に削減したという結果につながったと考えられる。

## (2) 提案方式と CBRP の比較

CBRP のパケット転送方式は DSR と同様ソースルーティング方式である。そのため、ノード移動速度が高速になるにつれて CBRP のパケット到達率が低下する要因は、DSR の場合と基本的に同じであるといえる。しかし、CBRP はクラスタベースルーティングであるため、発生するルーティングオーバーヘッドが DSR と比べて少ない。そのため、ノード移動速度が高速になっても輻輳自体は発生せず、パケット到達率の大幅な低下は回避できたと考えられる。しかし、CBRP のクラスタリング処理では、各ノードによる定期的な HELLO メッセージのブロードキャストによってトポロジデータベースを生成し、これをルーティングの際に利用している。ノード移動速度が高速になった場合、既に自身の1ホップ圏外に移動してしまったノードがトポロジデータベースに登録されたままになるなど、各ノードが保持するデータベース内の情報に乱れが生じやすくなる。トポロジデータベースに乱れが生じている際に、経路切断が発生すると、Packet Salvaging (Local Repair) によるパケット救出を試みたとしても代替経路が実際には使用不可能であるといったエラーが生じる。こうしたエラーによって、正しく転送することができずに最終的にパケットをドロップしてしまうことが、パケット到達率の低下につながったと考えられる。

一方、図 18 の提案方式の1転送の平均ホップ数からみてとれるように、提案方式では長経路通信が基本的に発生しない。さらに、隣接クラスタへの複数の短経路情報の中から最も新鮮な情報を元にパケットの転送を行う。そのため、より確実にパケットを転送することが可能となり、ノードが高速に移動する場合でも高いパケット到達率を維持できたと考えられる。

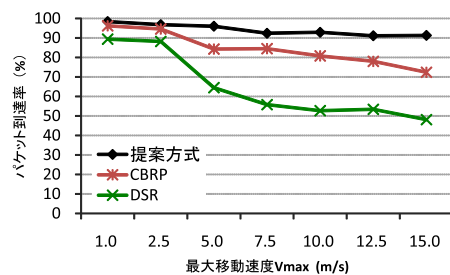


図 16 ノード移動速度に対するパケット到達率

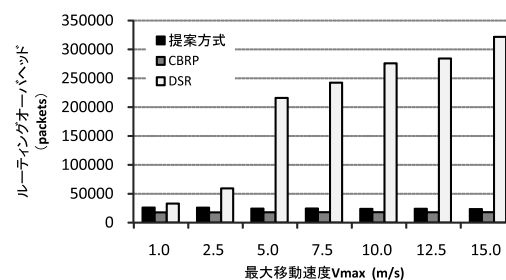


図 17 ノード移動速度に対するルーティングオーバーヘッド

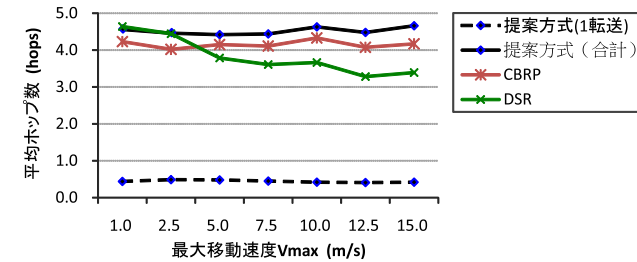


図 18 ノード移動速度に対する通信経路の平均ホップ数

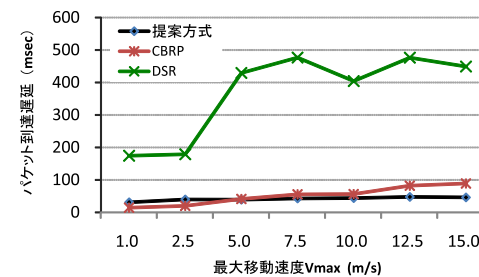


図 19 ノード移動速度に対するパケット到達遅延時間

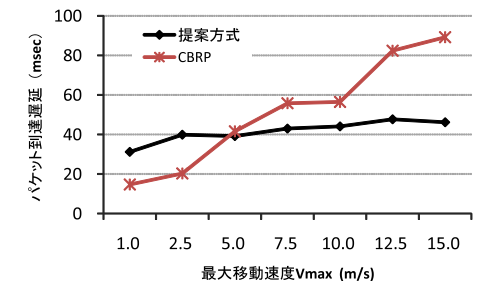


図 20 ノード移動速度に対するパケット到達遅延時間 (提案方式, CBRP)

## (3) パケット到達遅延時間に関する比較

ノード移動速度に対する受信したデータパケットの到達遅延時間の変化を図 19 に示す。このグラフからみてとれるように、DSR の遅延時間が他の方式と比較して圧倒的に大きく、ほとんど正常な通信が行われていないことが分かる。これは受信できたパケットのうちほとんどが Salvaging 機能によって救出されたパケットであったためと考えられる。最大移動速度が 1.0m/s および 2.5m/s の場合でも、DSR のパケット到達遅延時間は 170 ミリ秒を超える値となっており、通信信頼性は非常に低いといえる。

次に、提案方式と CBRP のパケット到達遅延時間を比較する (図 20)。ノード移動速度が低速である場合には、CBRP の遅延時間が提案方式の約半分となっていることが分かる。しかし、ノード移動速度が高速になると、遅延時間は提案方式の 2 倍の値となることが分かる。つまり、CBRP の遅延時間はノード移動速度に依存しているといえる。これは、長経路通信時の経路切断による Salvaging 機能が影響していると考えられる。ノード移動速度が低速の場合は長経路通信

であっても経路切断は発生しにくいいため、遅延時間を低く抑えることができる。しかし、ノード移動速度が高速になると長経路通信の経路切断の発生数が増加する。そしてパケットの再送処理や経路の再構築などの遅延が、パケット到達遅延の増加につながったと考えられる。

一方、移動速度が低速である場合に提案方式の遅延時間が CBRP の 2 倍に近い値となった理由の 1 つとして、実質のホップ数が CBRP のホップ数よりも大きかったためであると考えられる。実際に、図 18 の提案方式の合計ホップ数から分かるように、最大ノード移動速度に関わらず、平均ホップ数の値は提案方式の方が大きい。そのため、ノード移動速度が低速の場合に提案方式の遅延時間が CBRP よりも大きくなってしまったと考えられる。しかし、提案方式の遅延時間はノード移動速度に依存せず、常に 30~50 ミリ秒の間で推移している。これは、短経路通信による確実な転送の効果だけでなく、提案方式におけるクラスタリング処理の方針が好影響を与えていると考えられる。提案方式では、クラスタポイントによってクラスタの物理的な位置をある程度固定することができるため、一度収集した隣接クラスタ情報の変更される回数が少ない。そのため、送信元ノードが生成したクラスタパスを修復する必要はほとんどなく、経路制御に乱れが生じにくい。そのため、クラスタパスの修復等による遅延時間もほとんど生じないため、遅延時間に差が出にくくなったと考えられる。以上のことから、提案方式はノード移動速度に関わらず、非常に安定した通信を提供しているといえる。

## 6. おわりに

高移動・高通信頻度のアドホックネットワークにおける通信信頼性を向上させるため、本研究では長経路通信がネットワークに与える悪影響について着目し、長経路通信の発生を抑制するというアプローチを提案した。提案方式では、クラスタベースルーティングを採用することでルーティングオーバーヘッドの発生を最小限に抑え、なおかつクラスタリング処理の際に各ノードが短経路を生成・保持することにより、長経路通信が発生するような状況下でも新鮮な短経路を組み合わせて高信頼な通信を行う手法を提案した。提案方式をネットワークシミュレータ上に実装し、既存のアドホックルーティングプロトコルである DSR と、代表的なクラスタベースルーティングプロトコルである CBRP と比較評価を行った結果、DSR と比較して最大約 43%、CBRP と比較して最大約 19%の packets 到達率の向上がみられた。これらのシミュレーション結果から、提案方式は他のプロトコル比べてノードの移動速度に影響を受けにくく、ノードが高速に移動する環境下でも高い packets 到達率と安定した packets 到達遅延時間を提供することを示した。

現状の提案方式では、パケットを転送する際は必ず各クラスタの CH を経由しなければならない。これは CH への負荷という観点からみると必ずしも望ましいことではない。そこで、通信経路の最適化処理を今後提案方式に組み込むことが望ましい。また、クラスタリングに関しては、クラスタのメンバ数などに応じて動的にクラスタポイントを設定したり解除したりできるようなロードバランシング機構を組み込むことを予定している。

## 参考文献

- [1] The Dynamic Source Routing Protocol (DSR) for Mobile Ad Hoc Networks for IPv4. RFC4728, <http://www.ietf.org/rfc/rfc4728.txt>, February 2007
- [2] J.Y.Yu and P.H.J.Chong, "A survey of clustering schemes for mobile ad hoc networks," IEEE Communications Surveys & Tutorials, vol.7, no.1, pp.32-48, 2005
- [3] The Network Simulator version 2 (ns-2), <http://www.isi.edu/nsnam/ns/>
- [4] M.Jiang, J.Li, and Y.C.Tay, "Cluster based routing protocol (CBRP)," functional specification, IETF Internet Draft, MANET working group, draft-ietf-manet-cbrp-spec-01.txt, Aug. 1999.
- [5] 小田亮太郎, 大田知行, 角田良明, "モバイルアドホックネットワークにおける経路情報の分散管理手法を用いた階層ルーティング," 電子情報通信学会技術報告, Vol.107, no.193, pp.7-12, 2007 年 8 月.
- [6] Thomas Zahn and Jochen Schiller. "DHT-based Unicast for Mobile Ad Hoc Networks". In Proc. of IEEE Percom. pp.179-183, March 2006.
- [7] R.Oda, T.Ohta, and Y.Kakuda. "An Autonomous Clustering-Based Hierarchical Routing Using Decentralized Route Information Management Scheme for Mobile Ad Hoc Networks," ICDCS Workshops 2008: pp.569-574, 2008.
- [8] T.Zahn and J.Schiller. "MADPastry: A DHT Substrate for Practicably Sized MANETs," In Proc. of ASWN, June 2005.
- [9] 鳴海寛之, 白石陽, 高橋修, "Cluster-by-Cluster ルーティングにおけるクラスタヘッド選出手法に関する検討," 情報処理学会 マルチメディア, 分散, 協調とモバイル (DICOMO 2009) シンポジウム論文集, pp. 1074 - 1083, July 2009.
- [10] H.Narumi, Y.Shiraishi and O.Takahashi, "A Reliable Cluster-based Routing Algorithm for MANET", International Workshop on Informatics (IWIN) 2009, pp.44-51, September 2009.
- [11] K.Lee and C.Hwang, "Hierarchical Configuration Scheme for Scalable Ad Hoc Networks," In Proc. of ICCSA, LNCS 3981, pp.816-826, 2006.
- [12] T.Ohta, S.Inoue, Y.Kakuda, and K.Ishida. "An Adaptive Mutihop Clustering Scheme for Ad Hoc Networks with High Mobility," IEICE Trans, Vol.E86-A No.7, pp.1689-1697, June 2003.
- [13] F.Garcia, J.Solano, and I.Stojmenovic, "Connectivity based k-hop clustering in wireless networks," Telecommun. Syst., vol. 22, pp. 205-220, 2003.
- [14] G.He, S.Zhou, and S.Liu. "A K-hop Based Clustering Algorithm in Ad Hoc Networks," In Proc. of NAS 2008, pp.151-152, 2008.
- [15] 中川裕士, 石田賢治, 大田知行, 角田良明, "MANET における位置と移動情報を用いた階層型ルーティング方式," 電子情報通信学会 信学技報, IN2005-118, pp.49-54, November 2005.
- [16] F.D.Tolba, D.Magoni and P.Lorenz, "A Stable Clustering Algorithm for Highly Mobile Ad Hoc Networks," In Proc. of ICSNC 2007, pp. 11-16, August 2007.