

自己安定クラスタ構造を用いたアドホックネットワーク ルーティング方式

小林 基成[†] 森田 正範[†] 高橋 直久[‡] 片山 喜章[‡] 和田 幸一[‡]

[†](株)NTT ドコモ 総合研究所 〒239-8536 神奈川県横須賀市光の丘 3-5

[‡]名古屋工業大学 〒466-8555 愛知県名古屋市昭和区御器所町

E-mail: [†]{kobayashimot, moritamasa}@nttdocomo.co.jp, [‡]{naohisa, katayama, wada}@nitech.ac.jp

あらまし アドホックネットワークにおいては、構成する端末数が増加するとルーティングのための制御パケットが大量にアドホックネットワーク内に流れ、通信の帯域を圧迫する。制御パケット量を削減するため、クラスタリングの手法がよく用いられるが、端末の移動や、無線の不安定さによるリンク切断に伴うトポロジ変更への追従が難しい。本稿では、クラスタ構成に自己安定アルゴリズムを適用することによりトポロジ変更への追従性を向上させ、制御パケットの削減と、トポロジ変更への対応を両立する方式を提案し、コンピュータシミュレーションによる性能評価を示す。

キーワード アドホックネットワーク、クラスタリング、ルーティング、スケーラビリティ

Ad-hoc Network Routing Method on Self-Stabilizing Clustering

Motonari KOBAYASHI[†] Masanori MORITA[†]

Naohisa TAKAHASHI[‡] Yoshiaki KATAYAMA[‡] and Koichi WADA[‡]

[†] Research Laboratories, NTT DoCoMo 3-5 Hikarinooka, Yokosuka, Kanagawa, 239-8536 Japan

[‡] Nagoya Institute of Technology Gokisyo-tyou, Syouwa-ku, Nagoya, Aichi, 466-8555 Japan

E-mail: [†]{kobayashimot, moritamasa}@nttdocomo.co.jp, [‡]{naohisa, katayama, wada}@nitech.ac.jp

Abstract In ad-hoc networks, when there are large number of nodes, a large amount of control signal flood and consume a lot of communication band width. Clustering methods are proposed to reduce the amount of control signal, however it is difficult to adjust to the topology changes due to node movement and wireless instability. In this paper, we propose a routing method on self-stabilizing clustering. It both reduces the amount of control signal and quickly adapts to the topology changes. We also present simulation based evaluation to show the efficiency of our scheme.

Keyword Ad-hoc Networks, Clustering, Routing, Scalability

1. はじめに

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、AODVにおけるRREQ、OLSRにおけるTCメッセージのように基本的にフラッディングを用いた情報交換によりルート制御を行っている。一般にアドホックネットワーク内で送信される制御パケットは端末数の2乗に比例して増加することが知られている[1][2]。そのため、アドホックネットワークの規模が拡大し、端末数が増加すると制御パケットが大量にアドホックネットワーク内に流れ、通信の帯域を圧迫し、通信品質が低下してしまう。この問題を軽減するため、OLSRでは、フラッディング時に自身のパケットを中継する端末をMPRとして選定することで、TCメッセージを中継する端末を限定している。しかし、OLSRでは、多くの端末が隣接端末とのリンク情報をフラッ

ディングしたり、重複したリンク情報をフラッディングしたりと非効率な部分が存在する。そこで我々は、制御パケットの交換をさらに効率的に行い、オーバヘッドの小さいルーティングを実現することを目的とし、根付き木構造に構成されたクラスタを用いたルーティング方式(木構造クラスタルーティング)[3]を提案した。しかし、現実的な無線環境や端末が移動する環境において通信品質が劣化することがわかった。その一つの原因として、移動による頻繁なトポロジ変更があげられる。木構造クラスタルーティングでは、木を構成するリンクが切断すると、そのリンクに接続された子端末を根とする部分木をすべて初期化し、再度クラスタリングする。そのため、トポロジ変更時、再びクラスタが構成されるまでの時間が長く、通信不可能な状態が長く続くこととなる。もう一つの原因として、Gray

Zone 問題[4]などによる不安定なリンクの形成がある。現実の無線環境においては、パケットは確率的に受信に成功する。受信成功率の低い端末間にリンクが形成されると必然的に EtoE のパケット損失率も劣化することとなる。この問題の影響は OLSR においても起こるが、木構造クラスタリングでは、パケットは木に沿って伝送されるため、木を構成しているリンクに不安定なリンクが含まれた場合、多くの通信に影響を与えることになる。そこで、制御パケット量の削減しつつ、移動の影響によるパケット損失率を改善するため、クラスタリングの手法として移動や不安定なリンクによるトポロジ変更を「故障」として扱い、任意の「故障」状態から正常な状態への自動回復が保証される自己安定クラスタリングに着目した。自己安定化クラスタリングを用いることにより、移動や、不安定なリンクにより発生するトポロジ変化時にすばやく再クラスタリングし、パケット損失を防止することが期待できる。

本稿では、自己安定クラスタリングを用いたルーティング方式(自己安定クラスタリング)を提案する。そして、制御パケット量と通信品質としてパケット損失率をコンピュータシミュレーションにより評価した結果を示す。評価の結果、制御パケット量を低く抑えつつ、現実的な無線環境を要因とする品質劣化を改善することができたが、端末の移動に対する効果は少なかった。その原因を明らかにし、今後の研究方針を示す。

2. 自己安定クラスタ構造を用いたルーティング方式

新たに提案する自己安定クラスタリングは、以下の処理から構成される。

1. クラスタヘッドの決定
2. ゲートウェイの決定
3. ルーティング

クラスタの構造を図 1 に示す。「クラスタヘッドの決定」は、自己安定的にクラスタヘッドを決定する。クラスタヘッド同士は隣接しない、クラスタヘッド以外の端末はクラスタヘッドに隣接する特徴を持つ。「ゲートウェイの決定」は、隣接するクラスタのクラスタヘッド間を接続するゲートウェイを決定する。クラスタヘッド間のゲートウェイの数は 1 または 2 となる。

「ルーティング」は、基本的には端末が他の端末と情報交換し、ネットワークのトポロジを把握する。そして、トポロジ情報に基づいて Dijkstra アルゴリズムなどを用いて宛て先端末までの経路を計算するものである。経路計算のために必要なトポロジを把握するために端末間のリンク情報を送信する。本方式においては、クラスタの構造を利用し、経路計算に必要な最小限の

リンク情報のみを送信することで、制御パケット量を抑える。以下において各処理について詳細に説明する。

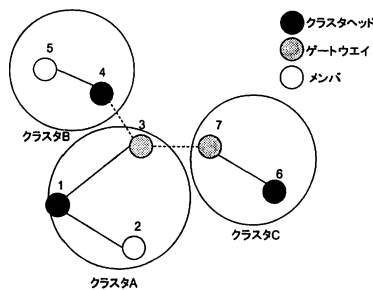


図 1 クラスタ構造

2.1. クラスタヘッドの決定方法

本方式に用いるクラスタの構成アルゴリズムは、Self-stabilizing DMAC[5]をベースとして、トポロジ変更によるクラスタ構造の変化がおきにくく改良した方式を用いる。Self-stabilizing DMAC では、トポロジ変更があったとき、自動的に再クラスタリングされてある状態に収束することが保証されるが、少しのトポロジ変化で多くの端末の状態が変更される問題がある。本方式では、以下のアルゴリズム[7]を用いることにより、トポロジ変更時の端末の状態変化回数を削減している。

- ・ 自身の状態がクラスタヘッドのとき
 - 周囲のクラスタヘッドの中に weight が自身より大きいクラスタヘッドが存在すると自身の状態をメンバとし、周囲の端末の中で weight が最大の端末を親とする
- ・ 自身の状態がメンバのとき
 - 周囲にクラスタヘッドが存在しなければ自身の状態をクラスタヘッドにする
 - 自身が親に指定しているクラスタヘッドが周囲に存在しなくなれば、周囲の端末の中で weight が最大の端末を親とする

ここで、weight とは、端末固有の重みであり、上記アルゴリズムでは weight が大きい端末が優先的にクラスタヘッドとなる。

2.2. ゲートウェイの決定手法

本方式では、あるクラスタヘッドはすべての隣接クラスタのクラスタヘッドに接続するためのゲートウェイを決定する。隣接クラスタとは以下のように定義される。

- ・あるヘッドからみて、自身のクラスタに属する隣接端末の隣接端末が属するクラスタ
- または、
- ・あるヘッドからみて、自身の隣接端末が属するクラスタ

図 1 では、クラスタ A とクラスタ B は隣接し、クラスタ

タヘッド間の距離は 2hop である。また、クラスタ A とクラスタ C は隣接し、クラスタヘッド間の距離は 3hop である。しかし、クラスタ B とクラスタ C はクラスタヘッド間の距離は 3hop であるが、間にクラスタ A の端末があるため隣接しないことになる。

クラスタヘッドは、ゲートウエイを決定すると Hello パケットを用いて該当ゲートウエイに通知する。メンバは少なくとも 1つのクラスタヘッドにゲートウエイ指定されている場合、ゲートウエイとして動作するようになる。また、ゲートウエイはどのクラスタヘッドからもゲートウエイに指定されなくなるとメンバに戻る。

図 1 では、端末 1 はクラスタ B、クラスタ C へのゲートウエイとして端末 3 を指定し、端末 4 はクラスタ A へのゲートウエイとして端末 3 を指定、端末 6 はクラスタ A へのゲートウエイとして端末 7 を指定する。つまり端末 3 と端末 7 がゲートウエイとなる。図 2 にクラスタヘッドがゲートウエイを決定するアルゴリズムを示す。

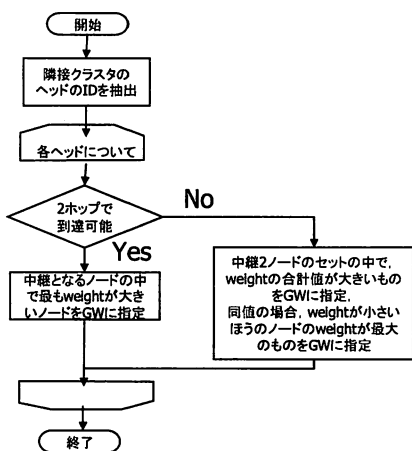


図 2 ゲートウエイ決定アルゴリズム

隣接クラスタヘッドまでのゲートウエイを決定するとき、候補が複数存在する場合、いずれかに一意に決定する必要がある。クラスタヘッド間が 2hop、つまり間にゲートウエイが 1つの場合、該当ゲートウエイの weight が最も大きいものに決定する。また、クラスタヘッド間が 3hop、つまり間にゲートウエイが 2つの場合、2つのゲートウエイの weight の合計が最も大きなものに決定する。同値の場合は、2つのゲートウエイのうち小さいほうの weight が最も大きいものに決定する。このようにゲートウエイを決定することで、任意のクラスタヘッド間のゲートウエイを一意に決定でき、往復の経路が対称となる。

2.3. Hello パケットの構成

本方式のクラスタヘッド及びゲートウエイを決定

するために各端末は以下の情報を保持する必要がある。

- ・隣接端末の状態(クラスタヘッド、GW、メンバ)
- ・2Hop トポロジ
- ・2Hop 内の端末の Weight
- ・2Hop 内の各端末の所属クラスタのクラスタヘッドの ID

これらの情報は、Hello パケットによる隣接端末との情報交換によって得る。そのため、各端末は Hello パケットに以下の情報をのせて送信する。Weight は情報量削減のため、ID で代用することとする。

- ・自身の状態(state)
- ・自身の属するクラスタのクラスタヘッドの ID(Head Address)
- ・隣接端末との間のリンクの状態、ゲートウエイ指定情報(LinkCode)
- ・隣接端末の ID(Neighbor Address)
- ・隣接端末が属するクラスタのクラスタヘッドの ID(Neighbor Head Address)

図 3 にパケットフォーマットを示す。LinkCode は隣接端末との間のリンクが双方向であるか片方向であるか、双方向の場合、ゲートウエイとして指定しているかどうかを示す。そしてリンクコードごとに隣接端末の ID のリストを記述する。リンクコードが双方向を示すものは、隣接端末の ID に加えて隣接端末の属するクラスタのクラスタヘッドの ID を並べて記述する。Link Message Size は各リンクコードのフィールドの大きさを示す。

本方式の Hello パケットは OLSR や木構造クラスタールーティングと比較して、隣接端末のクラスタヘッドの ID の情報を含む大きくなる。本方式の Hello パケットでは双方向リンクを持つ端末の ID にクラスタヘッドの ID を付加するため、多くの隣接端末との間のリンクが双方向である状況を想定すると、Hello パケットサイズは、OLSR、木構造クラスタールーティングの約 2 倍となる。

state	
Head Address	
Link Code	Link Message Size
Neighbor Head Address	
Neighbor Address	
Neighbor Head Address	
Neighbor Address	
.....	
Link Code	Link Message Size
Neighbor Address	
Neighbor Address	
.....	
Link Code	Link Message Size
.....	

図 3 Hello パケットフォーマット

2.4. ルーティング

2.4.1. リンク情報の送信

全端末がネットワークのトポロジを把握するため、クラスタヘッドは以下のリンク情報を定期的に変送する。

- ・自身と自身を親に指定しているメンバの間のリンク
- ・自身と自身がゲートウェイとして指定している端末との間のリンク
- ・隣接クラスタのクラスタヘッドまで 3hop の場合における中継ゲートウェイ間のリンク

図 4 にクラスタヘッドがリンク情報として送信するリンクの選択方法を示す。ゲートウェイ間のリンクはクラスタヘッドが直接保持していないので、どのクラスタヘッドがリンク情報として送信するか決定する必要がある。本方式においては、該当の2つゲートウェイのうち Weight の大きい方のゲートウェイに隣接しているクラスタヘッドのなかで Weight が最大のクラスタヘッドが該当ゲートウェイ間のリンクをリンク情報として送信する。図 1 においては、クラスタ A のクラスタヘッド 1 は端末 1-2 間、端末 1-3 間のリンク情報を送信する。クラスタ B のクラスタヘッド 4 は、端末 4-5 間、端末 4-3 間のリンク情報を送信する。クラスタ C のクラスタヘッド 6 は、端末 6-7 間、さらにゲートウェイ間リンクである端末 7-3 間のリンク情報を送信する。このように、クラスタを構成するすべてのリンクがクラスタヘッドによって重複なく送信される。リンク情報のパケットフォーマットを図 5 に示す。

Advertised Neighbor Main Address は、隣接端末の ID で、Advertised 2hop Neighbor Address は 2hop 端末の ID である。パケットは複数のフィールドに区切られ、各フィールドの長さは Link Message Size が示す。1つ目のフィールドには、送信すべきゲートウェイ間リンクを持たない隣接端末の ID のリストを記述する。図 1 の端末 1 では、端末 2 および端末 3 の ID を記述する。これにより、クラスタヘッドとそれに隣接するメンバやゲートウェイ間のリンクの存在が表現できる。2つ目以降のフィールドには、送信すべきゲートウェイ間リンクを持つ隣接端末を Neighbor Main Address とし、また 2hop 端末の ID を Advertised 2hop Neighbor Address とし記述する。これにより ID が Neighbor Main Address の端末と ID が Advertised 2hop Neighbor Address の端末との間のリンクの存在を表現できる。

2.4.2. リンク情報の中継

クラスタヘッドが送信するリンク情報をネットワーク全体へ送信するため、クラスタヘッド、ゲートウェイは他のクラスタヘッドから送信されたリンク情報を中継する。クラスタの性質より、すべての端末はクラスタヘッドであるか、またはクラスタヘッドに隣接

しているため、すべてのクラスタヘッドが1度ずつリンク情報を中継すればすべての端末にリンク情報を送信することが保証できる。また、隣接するクラスタのクラスタヘッド間はゲートウェイによって接続されているので、あるリンク情報をクラスタヘッドとゲートウェイが中継することでネットワーク中のすべての端末にリンク情報を送信することを保証できる。

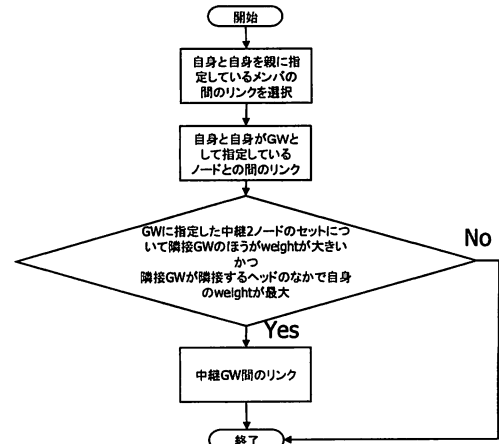


図 4 リンク選択アルゴリズム

Link Message Size
Advertised Neighbor Main Address
Advertised Neighbor Main Address
.....
Link Message Size
Advertised Neighbor Main Address
Advertised 2hop Neighbor Address
Advertised 2hop Neighbor Address
.....
Link Message Size
Advertised Neighbor Main Address
Advertised 2hop Neighbor Address
.....

図 5 リンク情報パケットフォーマット

3. シミュレーション評価

提案自己安定クラスタルーティング方式の制御パケット量、パケット損失率(PER)について、ネットワークシミュレータ Qualnet[6]を用いて OLSR 及び、木構造クラスタルーティングと比較評価を行った。

制御パケットは提案方式においては、リンク情報、OLSR においては TC メッセージとした。また、前提とする無線環境のモデルは、現実的なモデルとして IEEE802.11b 及び、単純化した無線モデルをシミュレートした。IEEE802.11b モデルは、bit error rate(BER)特性を元にパケット受信の成功を判定する。単純化無線モデルでは、Signal-to-Noise Ratio (SNR)がある値以上なら受信成功とするモデルである。送信端末からあ

る距離以内に存在する端末は全てのパケットの受信に成功し、それ以外のところでは全てのパケットを受信できない。端末の移動モデルについては Random Way Point とし、pose time は 0s、移動速度は 1-5/m/s とした。データトラフィックは、移動時の評価においては、特定の 2 端末間でペイロード長 512byte のパケットをシミュレーション時間中、1 秒間に 1 つ送信するものとし、受信できなかったパケットの割合をパケット損失率 (PER) として評価した。静止時のデータトラフィックについては、ある端末から他の全ての端末へ一定時間ごとに送信するものとした。制御パケット量の評価については、データパケットとの衝突による損失を起ささないよう、データトラフィックはない条件で評価した。表 1 にシミュレーション諸元を示す。

表 1 シミュレーション諸元

シミュレーション時間	3600秒
エリアサイズ	1500m×1500m
PHY / MAC	*IEEE802.11b(2Mbps) *Abstract(単純無線)
移動モデル	*静止、 *Random Way Point -1m/s~5m/s (平均2.5m/s) -Pose time=0(s)
ノード数	10、30、50、100、150、200
シミュレータ	Qualnet 3.8

図 6 にネットワークに参加している端末数を変化させたときの制御パケット量を示す。木構造クラスタリング(Cluster)と比較して自己安定クラスタリング(NewCluster)は制御パケット量が増加しているが、OLSR と比較すると非常に少ないことがわかる。

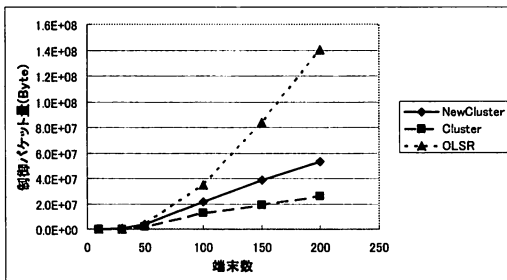


図 6 制御パケット量

図 7 に単純無線環境、端末静止状態でのパケット損失率を示す。自己安定クラスタリングと木構造クラスタリングはほぼ同値で、ネットワーク中の端末がほぼ連結する 100 台以上においては、ほとんどパケット損失は起こっていないことがわかる。OLSR は端末数が増加すると、制御パケットの量が提案 2 方式に比べて大きいため、帯域を圧迫し、パケット損失が発生する。自己安定クラスタリングは、木構造クラスタリングに比べ、制御信号量が増加し

ているが、200 台程度の規模のネットワークではパケット損失率に影響はない。

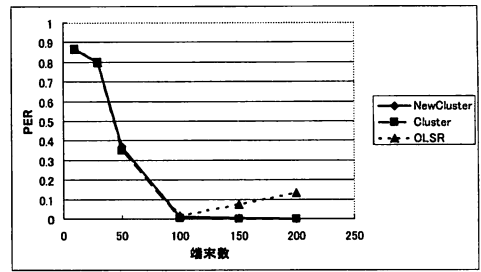


図 7 パケット損失率(単純無線, 静止)

図 8 に無線モデルが IEEE802.11、端末静止状態でのパケット損失率を示す。不安定リンクの影響により、各方式とも図 7 と比較して特性が劣化している。木構造クラスタリングと自己安定クラスタリングを比較すると、ネットワーク中の端末がほぼ連結する 100 台以上では、木構造クラスタリングのパケット損失率が約 0.3 なのに対して、提案方式は約 0.2 に抑えることに成功している。

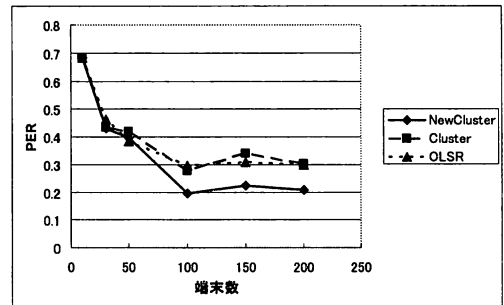


図 8 パケット損失率(IEEE802.11, 静止)

図 9 に単純無線環境、端末移動状態でのパケット損失率を示す。これも移動によるトポロジ変化の影響で、各方式とも図 7 と比較して特性が劣化している。木構造クラスタリングは OLSR よりパケット損失率が高くなっていたが、自己安定クラスタリングにおいても改善は見られない。

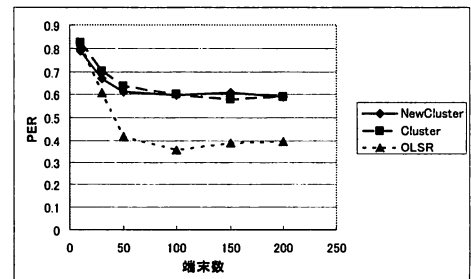


図 9 制御パケット量(単純無線, 移動)

4. 考察

4.1. 制御パケット量について

制御パケット量について考察する。リンク情報の送信頻度を同様とすると、制御パケット量はリンク情報(または TC メッセージ)にて送信するリンクの数に比例する。また、あるリンク情報をネットワーク全体に送信するための中継回数に比例する。リンク数に関しては、木構造クラスタルーティングの場合、木を構成するリンクを送信するので、(ネットワーク中の端末数-1)の数のリンクを送信することとなる。これはネットワーク中のすべての端末の連結のために必要な最も少ない数のリンクである。自己安定クラスタルーティングでは、すべての隣接するクラスタのクラスタヘッドをゲートウェイで接続する構成を取るため、木構造に構成するのに比べて必要なリンクの数が多くなる。

4.2. パケット損失率について

自己安定クラスタルーティングは、木構造クラスタルーティング、OLSR と比較して実無線環境でのパケット損失率の改善ができた。木構造クラスタルーティングでは、パケットは木に沿って送信されるため、木を構成するリンクに不安定なリンクが含まれた場合多くの通信に影響を与えてしまう。自己安定クラスタルーティングでは、すべての隣接クラスタへゲートウェイで接続されるため、クラスタヘッドによるメッシュ構造となる。よって、あて先端末まで複数の経路を確保しており、あるリンクが不安定になってもその影響は一部の通信に及ぶのみである。一方 OLSR では、より多くの経路を確保できるが、制御パケットが多く、制御パケットとデータパケットの衝突が発生するためパケット損失率は劣化する。

以上のように、実無線環境においてパケット損失率は改善できたが、移動環境における効果はなかった。この原因として、クラスタ決定までのステップが多く、時間がかかることが考えられる。木構造クラスタルーティングでは、端末の移動により木を構成するリンクが切断された場合、切断端末を根とする部分木の構造をすべて初期化する。その後またクラスタリングされていくというステップであった。このようにすると、特に根となる端末付近のリンクが切断されたとき多くの端末が初期化され、再びクラスタ構造ができるまで時間が必要である。自己安定クラスタルーティングの場合は、あるリンクが切断するによって端末の状態が変化したとき、切断点から数 hop 範囲の端末の状態を変更する。しかし、状態を変更する範囲の大きさは保証されておらず、即正常な状態に回復できる場合もあるが、最悪の場合ネットワーク全体の端末の状態を変化させる必要がある。端末の状態変化は Hello パケットによって隣接端末に伝達されていくため、再び正し

くクラスタリングされるまで時間がかかる。OLSR の場合は、MPR を決定するために、2 hop neighbor を参照している。よって、あるリンクが切断されたとき、切断部から 2hop 以内の端末が MPR を変更することとなる。つまりリンク切断に対して状態を変更する端末の範囲が限定的であり、短時間でリンク切断に対応でき、提案方式 2 方式に比べてパケット損失率が低くなっていると考えられる。

5. むすび

本稿では、自己安定的クラスタ構築手法、自己安定的クラスタ間ゲートウェイ構築手法を用いたルーティング方式を提案した。自己安定クラスタルーティングは、木構造クラスタルーティングに比べて制御パケット量が増加したが、OLSR に比べると非常に少なく抑えることができ、かつ実無線環境でのパケット損失率を改善することができた。しかし、端末が移動する環境においては、移動によるトポロジ変化への再クラスタリングが追従できずパケット損失率を改善することはできなかった。

今後の課題として、移動環境でのパケット損失率の改善をめざし、高速収束可能なクラスタ構造の検討を行う。また、さらにパケット損失率を改善するため、マルチパスルーティングの適用などにより、把握している複数の経路を有効に活用する検討を行う。

文 献

- [1] Naserian, M., Tepe, K.E., Tarique, M., "Routing overhead analysis for reactive routing protocols in wireless ad hoc networks", IEEE International Conference on Wireless And Mobile Computing, Networking And Communications, 2005.(WiMob'2005)
- [2] Sucec, J., Marsic, I., "Hierarchical routing overhead in mobile ad hoc networks", Mobile Computing, IEEE Transactions on Volume 3, Issue 1, Jan-Feb 2004 Page(s):46 - 56
- [3] 小林基成, 森田正範, 高橋直久, 片山善章, 和田幸一, "木構造クラスタを用いたアドホックネットワークルーティングプロトコルの評価", 信学技法 NS2006-20, 2006 年 4 月
- [4] Henrik Lundgren, Erik Nordstrom, Christian Tschudin, "Coping with Communication Gray Zones in IEEE 802.11b based Ad hoc Networks", The Fifth International Workshop on Wireless Mobile Multimedia (WoWMoM 2002), September 2002
- [5] C.Johnen, L.H.Nguyen. "Self-stabilizing clustering algorithm for ad hoc network", AlgoSensors2006, June 2006
- [6] Scalable Network Technologies, Inc., QualNet network simulation software, <http://www.scalable-networks.com/>
- [7] 宮永 慎太郎, 片山 喜章, 和田 幸一, 高橋 直久, "MANET 上でのクラスタを利用した通信路構築自己安定アルゴリズムについて", 情報処理学会アルゴリズム研究会 111-2, 2007 年 3 月