

リンク状態型ルーティング方式における 制御負荷低減手法に関する一検討

大浦 優季¹ 吉廣 卓哉^{1,a)}

概要：有線及び無線ネットワークにおける基本的なルーティング方式として、古くからリンク状態型ルーティング方式が活用されている。リンク状態型ルーティング方式は、トポロジの変化時に収束が速い利点があるが、定期的な送信するメッセージのための負荷が大きいことが問題視されており、有線・無線ネットワークのいずれにおいても、メッセージの制御負荷を低減するための工夫がなされてきた。特に無線ネットワークにおいてはメッセージ負荷低減への要求が高く、階層クラスタリングによるトポロジ情報の圧縮等が試みられてきた。しかし、階層化による方法ではクラスタヘッド同士が信頼性の高いメッセージ交換を必要とするため、逆にシステムの複雑性が増し、トポロジ変化への耐性が低くなる問題がある。本研究では、リンク状態型ルーティングにおいて、異なるノードが発行したメッセージをフラッディング中に適応的に統合することにより、経路計算に大きな影響を与えることなく、制御メッセージの負荷を低減する方法を提案する。

1. はじめに

近年の無線通信技術の発達により、ノートパソコンやスマートフォン等の無線通信可能な携帯端末が広く普及している。また、無線通信可能な携帯端末同士が自律的にリンクを確立し、ネットワークの構築を行う MANET (Mobile Ad-hoc NETWORK) が盛んに研究されている。MANET では、各端末が中継器としての機能を持ち、全て無線で通信できるため、有線ケーブルの敷設コストがかからないため、安価にネットワークを構築できるという利点がある。

MANET を対象として経路制御プロトコルとして、常に経路表を作成し維持するプロアクティブ型プロトコルが研究されている。プロアクティブ型プロトコルとしては、OLSR[1] などのリンク状態型プロトコルが主流である。リンク状態型では、ネットワーク内の全てのノードがトポロジを把握し、最短経路を計算することによって、経路表を作成する。このために、各ノードが隣接ノード情報を含んだトポロジ広告メッセージを作成し、定期的にネットワーク全体に広告する。このトポロジ広告メッセージの通信負荷が大きいことが問題である。特に MANET では、局所的にノードが集中することが多く、隣接ノードペア数が組み合わせ的に増大することがあり、通信負荷の問題はより一層深刻である。

メッセージの通信負荷を低減する方法はいくつか提案されている。OLSR では、MPR と呼ばれる仕組みを利用して、メッセージを中継するノードを制限することで負荷を低減する。各ノードは、自分が広告したメッセージを中継するノードとして隣接ノードの中から MPR を選択し、MPR のみがメッセージを再度広告することで、より少ない通信負荷でメッセージをネットワーク全体に広告する。また、自分を MPR に選んだノードへのリンクのみを広告する仕組みにより、最短経路の計算を保証しながらも、広告する情報量を低減する仕組みも備えている。MPR は、有線ネットワーク向けのリンク状態型プロトコルである OSPF[2] にも取り入れられている [3]。TBRPF[4] では、最短経路木に沿って上流ノードのみにメッセージを広告し、また、広告する情報を最短経路木の情報のみに絞ることで、通信負荷を低減する。しかし、これらの成果にもかかわらず、現在でも、リンク状態型の MANET ルーティングプロトコルは通信負荷が高いことが問題となっている。

階層化による通信負荷の低減も複数試みられている [6][7][8][9][10]。例えば、Ros らは OLSR に 2 階層の階層構造を導入し、クラスタ内では OLSR と同様の処理を行い、クラスタ間には上位階層のためのメッセージを追加することで、OLSR の通信負荷を低減する手法 C-OLSR を提案した [6]。Ge らは複数階層の階層構造により、通信負荷を低減する Hierarchical OLSR を提案した [7]。しかし、階層化による方法は、クラスタヘッドの選出や (クラスタ

¹ 和歌山大学システム工学部
930 Sakaedani, Wakayama, 640-8510, Japan
^{a)} tac@sys.wakayama-u.ac.jp

ヘッドを持たない手法の場合でも) クラスタ間処理のためのメッセージの追加により処理が複雑化する。また、クラスタヘッドの故障やクラスタヘッド間の通信障害等によるネットワーク全体への影響が大きくなり、耐障害性においても問題がある。

本研究では、リンク状態型ルーティングプロトコルにおける階層化をしない新たなメッセージ負荷低減手法として、メッセージの適応的統合に基づいた方法を提案する。リンク状態型プロトコルではメッセージのフラッディングは送信元ノード毎に独立であるが、本提案では、フラッディングの際に送信元が異なる複数のメッセージを統合し、広告トポロジを適応的にクラスタ化することで、メッセージの通信負荷を低減する。メッセージを受信した各ノードは、ネットワーク全体のトポロジをクラスタ化したトポロジの要約を把握でき、これを基に最短路計算をすることで経路表を構築する。シミュレーションにより、提案手法のメッセージ負荷削減効果を評価した。また、提案手法により、ほぼ経路ループのない、一貫した経路表が作成できることを確認した。

本論文の構成を以下に示す。2章では基礎としてリンク状態型経路制御法の概要とその問題点を示す。3章では提案手法を説明する。4章ではシミュレーションにより提案手法の評価を行う。5章で本研究をまとめる。

2. リンク状態型ルーティングプロトコルと問題点

2.1 リンク状態型ルーティングプロトコル

リンク状態型ルーティングは、ネットワークにおける代表的な経路制御法の一つである。有線ネットワークにおいても OSPF[2] や IS-IS?が、代表的なプロトコルとして用いられてきた。リンク状態型では、まず、各ノードが Hello メッセージにより隣接ノードを把握し、これを接続しているリンクの集合とする。次に、接続リンクの集合をトポロジ広告メッセージとしてネットワーク内に広告し、フラッディングにより全ノードに届ける。(本稿では、以後、トポロジ広告メッセージのことを単にメッセージと呼ぶことにする。) ネットワーク内の全ノードからメッセージを受信することでネットワークトポロジが把握でき、各ノードが最短路計算をすることにより、経路表を構築する。距離ベクトル型とは異なり、リンク状態型にはトポロジ変化時の経路の収束が速い利点があるが、一方で、メッセージによる通信負荷と経路計算における計算負荷が大きい欠点がある。計算負荷は、近年の計算機の性能向上により問題視されなくなっているが、特に無線ネットワークにおけるメッセージ負荷は現在でも大きな問題である。

2.2 リンク状態型プロトコルにおけるメッセージ負荷

リンク状態型プロトコルにおいては、一定時間毎にメッ

セージをフラッディングする。具体的には、無線ネットワークの場合には、各ノードは、自分が生成したメッセージと、他ノードが生成し自分が中継すべきメッセージの両方を、一定時間毎に隣接ノードにブロードキャストする。このための通信負荷は、各メッセージ毎に必要なヘッダ情報を別にすると、主に広告するリンクの数に依存して大きくなる。リンクは通常は有向であり、両端ノードのアドレスにより表現される。従って、各ノードが生成するメッセージは、生成ノードアドレスと、隣接ノードアドレス集合により定義される。本研究では、メッセージあたりの負荷として、これらのサイズに、シーケンス番号のサイズを加えた値を考える。シーケンス番号は、メッセージ受信ノードがメッセージの鮮度を判断する上で重要な値であり、メッセージ統合においても影響が大きいため、考慮する。メッセージのヘッダに含まれる他の値については、ルーティングの原理を考える上で本質的ではないため、議論の簡単化のため、本研究では考慮しないこととする。

即ち、リンク状態型メッセージの通信負荷は、各ノードの隣接ノードの数(次数)により大きく変動する。例えば、平面を正三角形分割するようなネットワークを考えると、各ノードの次数は6であり、各メッセージはアドレス7つとシーケンス番号を含む。ネットワーク内のノード数を N と書くと、各ノードは $7N$ 個のアドレスと N 個のシーケンス番号の合計サイズとして見積もれる。一般には、市街地などでは、数多くの無線端末が局所的に集中し、それらが互いに接続してネットワークを構築する場合も考えられる。そのような場合には、リンク数が爆発的に増加することが容易に想像できる。MANET は様々な状況で動作することが想像されるため、このような状態であっても、十分に低いメッセージ負荷でネットワークが維持されることが求められる。

3. メッセージ統合による制御負荷低減手法の提案

3.1 問題解決のためのアイデア

経路表は宛先と次ホップノードのアドレスと距離から成るため、適切な次ホップノードを選ぶことさえできれば、データを送信できる。遠くで隣接関係にある複数のノードにデータを送る際、同じノードを次ホップノードとして選ぶ可能性が高い。よって、遠くで隣接関係にある複数のノードを1つのクラスタとして考えてもデータを送信できる。提案手法では、一定の距離より遠くで隣接関係にある複数のノードを1つのクラスタとみなすようにトポロジ広告メッセージを統合する。各ノードがメッセージの統合を行うことで、ネットワーク全体におけるトポロジ広告メッセージによる通信負荷を低減する。

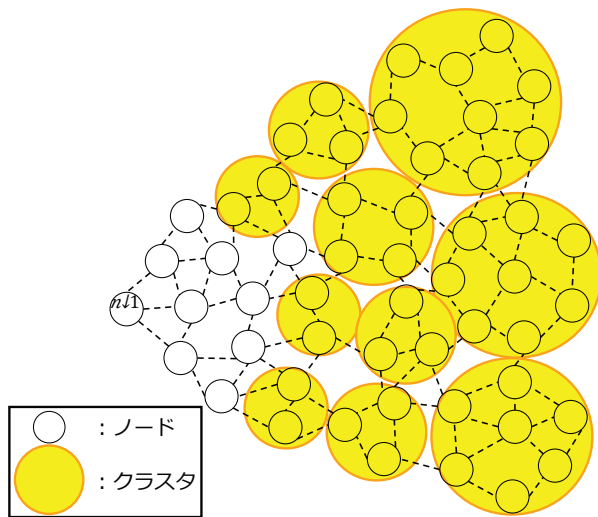


図 1 概要

3.2 提案手法の概要

リンク状態型ルーティング方式(以後、既存手法と呼ぶ)では自身の知り得るトポロジ広告メッセージ全てをそのまま送信するため、受信ノードはグラフ全体を把握するが、メッセージの負荷は大きい。各ノードで自律分散的に経路計算を行う時、経路表に一貫性があればループは発生しない。近くでは正確なトポロジを、遠くでは荒いトポロジを把握できれば、各ノードで自律分散的に経路計算を行っても、宛先に対する次ホップを適切に選べるため経路表は一貫性を持つ。よって、提案手法では近くでは正確なトポロジを、遠くでは荒いトポロジを把握できるように圧縮したグラフを各ノードが広告する。

提案手法では、フラッディングにおいて少しずつメッセージを統合することにより、メッセージ伝搬中にトポロジを圧縮するようにノードをクラスタ化する。近くでは正確に把握する必要があるため1クラスタに1ノードのみ、遠くでは大まかに把握できれば良いため1クラスタに複数ノード含める。距離が遠いほどクラスタに含めるノード数を多くする。提案手法では距離に応じてクラスタに含めるノード数を変更するため、経路計算の後、統合処理を行う。提案手法のイメージを図1に示す。ノード n_1 を基準とすると、遠くにあるクラスタには、近くにあるクラスタよりも多くのノードを含める。複数のノードを1つのクラスタとして考えることで、複数のトポロジ広告メッセージを1つのメッセージにまとめられる。

既存手法で使用するトポロジ広告メッセージは、次の3つのエントリから成る。

- A) 発行ノード：トポロジ広告メッセージを発行するノードのアドレス。
- B) 隣接ノード集合：発行ノードの隣接ノードのアドレス。
- C) シーケンス番号：メッセージの鮮度を表すシーケンス番号。

これに対して提案手法では、複数のノードを1つのクラスタに統合することを考える。これは、発行ノードが異なる複数のメッセージを、一つのメッセージに統合することで実現する。具体的には、提案手法のメッセージは、

提案手法では、複数のメッセージを統合した結果、メッセージは次のようなエントリを持つ。

- a) 主ノード集合：主ノードのアドレスとシーケンス番号の組の集合である。主ノードとは既存手法における発行ノードである。発行ノードを統合し、1つのクラスタとみなす。シーケンス番号には、統合前の値をそのまま用いる。
- b) 隣接ノード集合：主ノード集合を1つのクラスタとみなした時の隣接ノードのアドレスの集合である。隣接ノード集合は、統合前の複数メッセージに含まれる隣接ノード全てから、主ノードを除くことで生成される。

クラスタ情報の主ノード集合には複数のノードをまとめたものをクラスタとみなして含む。このようにすると、ネットワークは、クラスタによるネットワークとなるが、提案手法のメッセージは、隣接ノードとしてクラスタのリストを用いるのではなく、ノードのリストを用いる。これは、メッセージを繰り返し統合する際に、隣接ノードを正確に把握するための措置である。隣接クラスタを管理する方法では、共通に含まれるノードが存在する複数のクラスタを統合する場合に、正確に隣接関係を持てない場合が存在する。

提案手法では、受信時にクラスタ情報の統合処理を行い、送信時には統合処理を行ったクラスタ情報をそのままメッセージとして送信する。提案手法では、距離に応じて統合処理を行うため、経路計算の後、クラスタ情報の統合を行う。提案手法のメッセージ受信時の処理手順を以下に示す。

- 手順1: トポロジ広告メッセージの保存(3.4.1節)
- 手順2: トポロジ広告メッセージ生成のためのクラスタ情報の保持(3.4.2節)
- 手順3: 経路計算(3.4.3節)
- 手順4: クラスタ情報の統合(3.4.4節)

提案手法では、経路計算の結果を基にクラスタ情報の統合を行う。より多くのノードの情報を統合する方が、トポロジ広告メッセージによる通信負荷は低減する。しかし、データを宛先まで届けることができなければ意味がない。今回は、距離に応じて主ノード集合に含むノード数を制限するために、クラスタ主ノード数上限というものを設け、クラスタ情報を統合しすぎることを避ける。クラスタ主ノード数上限については3.5節で述べる。

また、各ノードで自律分散的に経路計算を行うため、距離が遠い順に統合すると各ノードで統合するノードがバラバラになり、クラスタ情報の主ノード集合に含むノードが重複し、トポロジ広告メッセージによる通信負荷をあまり低減できない。よって今回は、クラスタ情報に共通のノード

ドを多く含むクラスタペアから統合するために共通近傍ノード数というものを設け、統合を行う。共通近傍ノード数については 3.6 節で述べる。

3.3 トポロジ広告メッセージ

提案手法では複数のノードを 1 つのクラスタとして考えることで、複数のトポロジ広告メッセージが 1 つのトポロジ広告メッセージに統合され、トポロジ広告メッセージのサイズが小さくなり、通信負荷が低減する。

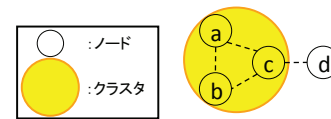
図 2(a) のネットワークを用いて、既存手法と提案手法のトポロジ広告メッセージを具体的に述べる。図 2(a) でクラスタに囲まれているノードは、クラスタの主ノードを表す。図 2(a) のネットワークはノード a, b, c, d の 4 ノードから成り、提案手法ではノード a, b, c を 1 つのクラスタとみなすことにする。既存手法では、図 2(b) のトポロジ広告メッセージが生成される。既存手法のトポロジ広告メッセージでは、発行ノードはトポロジ広告メッセージを発行するノードであり、隣接ノードは発行ノードの隣接ノードの集合である。一方、提案手法では、図 2 (c) のトポロジ広告メッセージが生成される。提案手法のトポロジ広告メッセージでは、発行ノードはトポロジ広告メッセージを発行するノードであり、主ノードは、クラスタに含まれているノード (統合処理がされていない場合は 1 つのノード) の集合であり、隣接ノードは主ノード集合の隣接ノードの集合である。

図 2(b)(c) のトポロジ広告メッセージのサイズを計算してみる。仮に、ノードのアドレスは 4byte、シーケンス番号と主ノード集合や隣接ノード集合に含まれるノードの数は 1byte と設定する。隣接ノードと提案手法における主ノードの数はクラスタ情報によって異なるため、必要である。既存手法においてトポロジ広告メッセージは、ノード a が 14byte、 b が 14byte、 c が 18byte、 d が 10byte となり、合計は 56byte となる。提案手法においてトポロジ広告メッセージは、ノード a, b, c を 1 つのクラスタとみなす時のクラスタ情報 21byte、ノード d が 11byte、送信ノードが 1byte となり、合計は 33byte となる。よって、提案手法では複数のノードを 1 つのクラスタとして考えることで、複数のトポロジ広告メッセージが 1 つのトポロジ広告メッセージにまとめられ、トポロジ広告メッセージのサイズが小さくなり、通信負荷が低減する。

3.4 メッセージ受信時の処理

3.4.1 メッセージの保存

受信したトポロジ広告メッセージを保存する。トポロジ広告メッセージは隣接ノードのみから受信するため、過去にトポロジ広告メッセージを受信したことがある場合、古いトポロジ広告メッセージは破棄する。



(a) トポロジとクラスタ

ノードaの制御メッセージ	
発行ノード(シーケンス番号)	a(100)
隣接ノード	b, c

ノードbの制御メッセージ	
発行ノード(シーケンス番号)	b(45)
隣接ノード	a, c

ノードcの制御メッセージ	
発行ノード(シーケンス番号)	c(220)
隣接ノード	a, b, d

ノードdの制御メッセージ	
発行ノード(シーケンス番号)	d(105)
隣接ノード	c

発行ノード: X	
ノードa, b, cを統合したクラスタ情報	
主ノード(シーケンス番号)	a(100), b(45), c(220)
隣接ノード	d
ノードdのクラスタ情報	
主ノード(シーケンス番号)	d(105)
隣接ノード	c

(b) 既存手法のトポロジ広告メッセージ

(c) 提案手法のトポロジ広告メッセージ

図 2 広告メッセージ

3.4.2 メッセージ生成のためのクラスタ情報の保持

受信したメッセージから、トポロジ広告メッセージ生成のためのクラスタ情報を保持する。異なる隣接ノードから受信した複数のメッセージからクラスタを取り出し、以後の処理に使用するためにデータベース化する。過去に保持したことがあるクラスタ情報は、シーケンス番号を基に削除し、新しいクラスタ情報を保持する。この時、包含関係にあるクラスタ情報は保持し、後の統合処理の際、統合が可能なら統合を行う。また、統合は毎回同じノードペアに対して行われる訳ではないため、過去に 1 度だけ統合されたノードペアのクラスタ情報を破棄せず保持してしまう。よって、クラスタ情報集合から各ノードの最新のシーケンス番号を洗い出し、クラスタ情報の主ノードのシーケンス番号が全て最新より a (今回は 10) よりも古い場合には破棄する。

3.4.3 経路計算

既存手法ではノードをリンクで結んだグラフを入力として Dijkstra のアルゴリズムを用いて各ノードまでの経路を計算するが、提案手法ではクラスタをリンクで結んだグラフを入力として Dijkstra のアルゴリズムを用いて各クラスタまでの経路を計算する。ただし提案手法では既存手法で用いる Dijkstra のアルゴリズムと異なることが 2 点ある。

まず、クラスタ C_x の隣接ノードがクラスタ C_y の主ノード集合の部分集合である時、クラスタ C_x とクラスタ C_y は隣接関係とみなす。次に、クラスタの主ノード集合のノード数に応じて重みを加算する。この 2 点を考慮し経路計算を行うことで、各宛先までの最短経路を求められる。

図 3 の例を用いて経路計算手順を説明する。図 3 でクラスタに囲まれているノードは、クラスタの主ノードを表す。今回、隣接関係にあるクラスタ間の距離は 1 とする。また、主ノード集合のノード数が k ($k > 1$) 個の時、そのクラスタを経由すると、距離に k の平方根の重みを加算する。ただし、クラスタ主ノード集合に含まれるノードが部分的に同じ場合、主ノード集合のノード数から同じノードの数を引

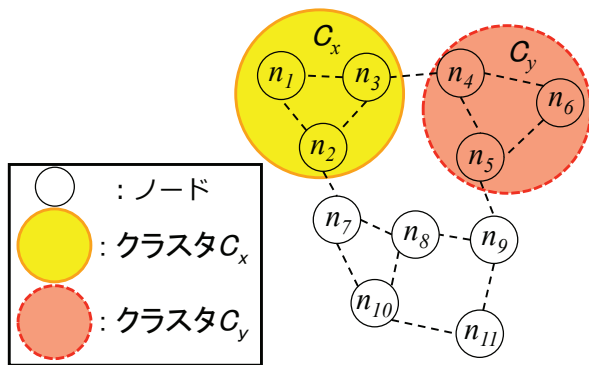


図 3 経路計算の例

いたものの平方根を加算する。図 3 では、クラスタ C_x の隣接ノード n_4 がクラスタ C_y の主ノード集合の部分集合であるため、クラスタ C_x とクラスタ C_y は隣接関係のみならず、この時、ノード n_{11} からクラスタ C_x 内のノードまでの距離は、ノード n_{10} と n_7 を経由する場合は 3、ノード n_9 とクラスタ C_y を経由する場合は $3 + \sqrt{3}$ となる。よって、ノード n_{11} からクラスタ C_x にあるノードまでデータを送る時、次ホップとしてノード n_{10} を選ぶ。

3.4.4 クラスタ情報の統合

提案手法では、一定の距離より遠くで隣接関係にあるノードを統合するために、最初に距離の降順にクラスタ情報を並び替える。その後、自身からの距離に応じて受信したクラスタ情報の統合処理を行うか判断する。クラスタの統合では、次の 4 つの処理を順に行う。

手順 1. 距離の降順にクラスタ情報をソート: 3.4.3 項の経路計算を基に、クラスタ情報を距離の降順にソートする。

手順 2. クラスタ主ノード数上限を超えるクラスタ情報の削除: 近くのノードは正確なトポロジを、遠くのノードは荒いトポロジを把握するために、提案手法では、3.5 節で述べるクラスタ主ノード数上限というものを設け、距離に応じてクラスタ情報に含む主ノードの数を制限する。隣接ノードから受信した統合されたクラスタ情報には、自身の距離では統合すべきではないクラスタが含まれている。よって、クラスタ主ノード数上限を超えるクラスタ情報を削除して、別のノードから受信したバラバラのクラスタ情報を優先する。

手順 3. 遠いクラスタを優先して、クラスタペアの統合を繰り返す: 提案手法では距離が遠ければ遠いほど、クラスタ情報の主ノード集合に含めるノード数が多くなるため、遠いクラスタペアから統合を行う。しかし、遠い順に統合することで、各ノードでバラバラのクラスタを統合してしまう。今回は、クラスタ情報に共通のノードを多く含むクラスタ同士から統合するために 3.6 節で述べる共通近傍ノード数というものを設け、統合を行う。共通近傍ノード数を用いることで、クラ

スタの主ノード集合に含むノードが部分的に同じ場合や、隣接ノードに同じものを選んでいく割合が高いものから統合することができる。共通近傍ノード数が大きいものは、ノード間のリンクが多いと考えられるため、1 つのリンクが切断しても別のリンクから宛先に到達できる可能性が高いため、統合を行っても良いと考える。また、提案手法では、隣接クラスタの中で、共通近傍ノード数が 1 番大きなクラスタを見つけてもすぐには統合を行わず、共通近傍ノード数が 1 番大きなクラスタのさらに隣接クラスタを見て、その中に共通近傍ノード数が 1 番大きな別のクラスタがない場合に統合を行う。今回、あるクラスタに対する隣接クラスタの共通近傍ノード数が同じものが複数ある場合、その中で主ノード集合のアドレスの数値の和が大きいものを優先して統合を行い、各ノードでできるだけ同じクラスタペアを統合することを目指した。

手順 4. 包含関係にあるクラスタ情報の削除: クラスタ C_x の主ノード集合がクラスタ C_y の主ノード集合の部分集合である時、クラスタ C_x とクラスタ C_y は包含関係にあるので、クラスタ C_y のクラスタ情報は保持し、クラスタ C_x のクラスタ情報は削除する。今回、提案手法では 3.6 節で述べる共通近傍ノード数を考慮し、統合を行うため、統合処理の際、包含関係にあるクラスタの統合が行われないことがあるのでこの処理が必要である。

3.5 クラスタ主ノード数上限

より多くのノードを統合してクラスタとする方が、トポロジ広告メッセージによる通信負荷は低減する。しかし、データを宛先まで届けることができなければ意味がないため、クラスタ情報に含める主ノードの数を制限する必要がある。よって、提案手法では、クラスタ主ノード数上限というものを設ける。近くのノードにデータを送信するためには、正確に次ホップを選ばなければならないが、遠くのノードは大まかな方向が分かれば、次ホップを正しく選ぶことができる。これは、データを送信し始めた時は大まかな位置しか分からなくても、データが宛先に近づくにつれて、宛先に近いノードが適切にデータを送信するため可能である。よって、ノードまでの距離が遠ければ遠いほど、クラスタ情報に含む主ノードの数を増加させる。提案手法では、距離に応じて主ノード集合に含めるノード数を変更するために、主ノード数上限関数を設ける。統合処理の際は、3.4.3 項で求めた宛先までの距離に応じて主ノード集合に含めるノード数を制限する。

図 4 のネットワークを用いてクラスタ主ノード数上限を考慮した統合処理を具体的に述べる。図 4 でクラスタに囲まれているノードは、クラスタの主ノードを表す。今回、宛先までの距離が 2 までは主ノード集合に含むノード数の

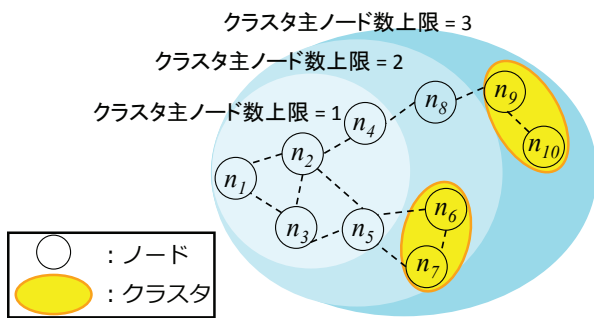


図 4 クラスタ主ノード数上限

上限は 1、それより遠いクラスタは距離から 1 引いた数を主ノード集合に含むノード数の上限とする。ノード n_1 からの距離が 2 であるノード n_4 や n_5 では主ノード数の上限は 1 なので統合処理を行えない。しかし、ノード n_1 からの距離が 3 であるノード n_6 や n_7 では主ノード数の上限は 2 で、統合後の主ノード集合に 2 個まで含まれるため、図のようにノード n_6 と n_7 を統合しクラスタとすることができる。この時、ノード n_2 はノード n_1 からノード n_6 と n_7 を統合したクラスタ情報を受け取るが、ノード n_2 からノード n_6 と n_7 までの距離は 2 で主ノードの上限が 1 なので、そのクラスタ情報は削除する。また、ノード n_1 からノード n_9 までの距離は 4 であるため、統合後の主ノード集合には 3 個まで含めることができる。ノード n_9 と n_{10} の統合後、もう 1 ノードを主ノードに統合することができるが、ノード n_8 を統合する場合、ノード n_1 からの距離が 3 であるため、主ノードに 2 個までしか含めることができない。よって、ノード n_1 では、ノード n_9 と n_{10} にノード n_8 を統合しない。

3.6 共通近傍ノード数

提案手法では自律分散的にノードの統合を行うため、距離が遠い順に統合すると、各ノードで統合するノードがバラバラになり、クラスタ情報の主ノード集合に含むノードが重複し、トポロジ広告メッセージによる通信負荷をあまり低減できない。統合を行う際、共通のノードを多く含むクラスタペアから統合するようすれば各ノードでできるだけ同じノードから統合する。ネットワークにおいてクラスタ性と関係が深い共通隣接ノードというものがある。共通隣接ノードとはノードペアの共通する隣接ノードのことで、共通隣接ノード数が多いノードペアはノード間のリンクが多い。共通隣接ノードでは自身のノードを除いた隣接ノードのみを考慮するが、提案手法では自身のノードも含め共通のノードを多く含むクラスタペアの統合を行いたい。よって本研究では、共通隣接ノードの拡張として、共通近傍ノードを定義する。共通近傍ノードは 2 つのクラスタの主ノード集合と隣接ノード集合に含むノードの論理積とする。

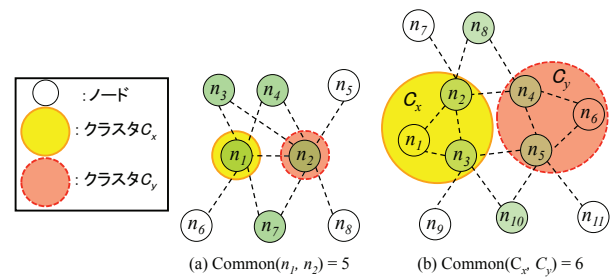


図 5 共通近傍ノード数

図 5(a) のネットワークにおいてノード n_1 と n_2 の共通近傍ノードは、主ノードの n_1 と n_2 、隣接ノードの n_3, n_4 と n_7 の 5 個である。また、図 5(b) でクラスタに囲まれているノードは、クラスタの主ノードを表す時、クラスタ C_x と C_y の共通近傍ノードは、主ノードの n_2, n_3, n_4 と n_5 、隣接ノードの n_8 と n_{10} の 6 個である。

共通近傍ノード数が多いものは、ノード間のリンクが多いと考えられるため、1 つのリンクが切断しても別のリンクから宛先に到達することができる可能性が高いため、統合を行っても良いと考える。また、2.3 節で述べた通り、ノード間のリンクが多いところを統合した方が、広告すべきリンクの数が減るため、トポロジ広告メッセージの低減にも繋がる。また、提案手法では、隣接クラスタの中で共通近傍ノード数が 1 番大きなクラスタを見つけてもすぐには統合を行わず、共通近傍ノード数が 1 番大きなクラスタの隣接クラスタを見て、その中に共通近傍ノード数が 1 番大きな別のクラスタがない場合に統合を行う。今回、あるクラスタに対する隣接クラスタの共通近傍ノード数が同じものが複数ある場合、統合候補の主ノード集合のアドレスの和が大きいものを優先して統合を行い、各ノードでできるだけ同じクラスタペアの統合を行うことを目指した。

3.7 統合処理の例

図 6 を用いて、共通近傍ノードを用いた統合処理を具体的に説明する。図 6 でクラスタに囲まれているノードは、クラスタの主ノードを表す。ノード n_{12} がクラスタ情報としてクラスタ C_x, C_y, C_z とノード n_9, n_{10} と n_{11} を受け取った時の処理を考える。まず、ノード n_{12} から 1 番遠いクラスタ C_x を基準に統合処理を行うとする。クラスタ C_x の隣接クラスタにはノード n_9 とクラスタ C_y がある。クラスタ C_x とノード n_9 の共通近傍ノードは、 n_3 と n_9 の 2 個である。また、クラスタ C_x とクラスタ C_y の共通近傍ノードは n_2, n_3, n_4 と n_5 の 4 個である。よって、クラスタ C_x の隣接クラスタの中で共通近傍ノード数が 1 番大きいクラスタは C_y である (以後、統合候補 A と呼ぶ)。次に、クラスタ C_x の隣接クラスタであるクラスタ C_y の隣接クラスタを見る。クラスタ C_y に対する隣接クラスタにはノード n_{10} とクラスタ C_z がある。クラスタ C_y とノード

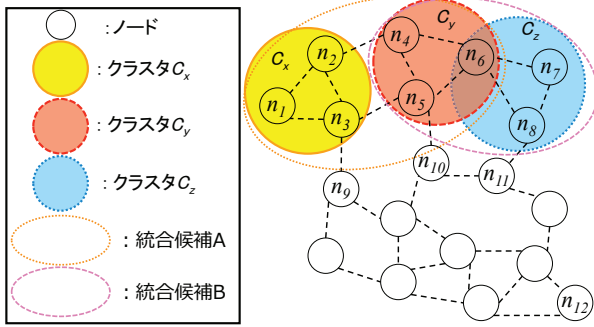


図 6 クラスターの統合

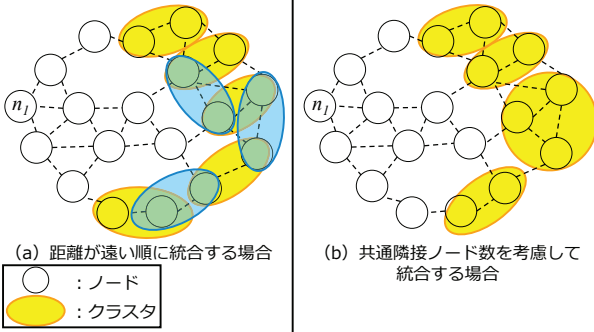


図 7 統合時の問題

n_{10} の共通近傍ノードは、 n_5 と n_{10} の 2 個である。また、クラスター C_y とクラスター C_z の共通近傍ノードは n_4 、 n_5 、 n_6 、 n_7 と n_8 の 5 個である。よって、クラスター C_y との隣接クラスターの中で、共通近傍ノード数が 1 番大きいのはクラスター C_z である（以後、統合候補 B と呼ぶ）。統合候補 A と B では、統合候補 B の方が共通近傍ノード数が大きいので、統合候補 A であるクラスター C_x はクラスター C_y との統合は行わない。後に、クラスター C_y を基準に統合処理を行う時、統合候補 B であるクラスター C_y とクラスター C_z を統合する。この時、クラスター主ノード数上限の関係でノード n_{12} ではクラスター C_y とクラスター C_z を統合することが出来なくても、クラスター主ノード数上限を満たす距離になった時に統合できるため、統合候補 A であるクラスター C_x とクラスター C_y は統合しない。

図 7 を用いて、距離が遠い順に統合する場合と共通近傍ノード数を考慮して統合する場合の違いを述べる。距離が遠い順に統合する場合、図 7 の (a) のように各ノードで統合するノードがバラバラになり、クラスター情報の主ノード集合に含むノードが重複する。また、共通近傍ノード数を考慮して統合する場合、生成されるクラスターは図 7 の (b) のようになり、クラスター情報の主ノード集合に含むノードが重複する可能性が低くなる。クラスター情報の主ノード集合に含むノードが重複すると、トポロジ広告メッセージによる通信負荷をあまり低減できない。よって、共通近傍ノード数を考慮することで重複が減り、トポロジ広告メッセージによる通信負荷を低減できる。

4. 評価

4.1 評価方法

提案手法により通信負荷をどの程度低減できるかを評価する。プログラミング言語 C++ を用いて、シミュレータを自作した。シミュレータでは各ノードが一定時間ごとにメッセージを送信し、それを受信する処理を模倣した。本評価ではメッセージサイズを見積もることが目的であるので、シミュレーションではメッセージの衝突や損失を考慮しない。シミュレータに 4 種類のトポロジを入力し、計算される経路が収束するのに十分な時間だけ実行した。

クラスター主ノード数上限としては線形関数を用いた。具体的には、クラスター主ノード数上限は、宛先までの距離が 2 までは主ノード集合に含むノード数の上限は 1、それより遠いクラスターは距離から 1 引いた数を主ノード集合に含むノード数の上限とした。経路計算の際、隣接関係にあるクラスター間の距離は 1 とした。また、主ノード集合のノード数が k ($k > 1$) 個の時、そのクラスターを経由すると、距離に k の平方根の追加コストを加算した。ただし、クラスター主ノード集合に含まれるノードが部分的に同じ場合、主ノード集合のノード数から同じノードの数を引いたものの平方根を加算した。

評価指標としては、通信負荷の低減効果を見るために、全ノードが一定時間毎に送信するメッセージの合計サイズを用いる。また、クラスター化により宛先までの正確な距離が失われることから、提案手法がどの程度最短路に近い経路を計算できるのかを知るために、最短路を用いた場合の次ホップと一致する次ホップの割合を求めた。

評価に使用した 4 つのトポロジを図 8 に示す。トポロジ A は手作業で作成した。トポロジ B~D は、ノードの次数が一定を超えない制約の下で正方形領域にランダムにノード配置することにより生成した。一定時間ごとに送信されるトポロジ広告メッセージの合計サイズを比較するために、各フィールドのサイズを以下のように設定した。IPv4 (Internet Protocol version 4) を想定し、アドレスは 4byte と設定した。また、シーケンス番号やノード数は 1byte あれば十分であると判断し、1byte と設定した。

- 発行ノード、隣接ノード、主ノード、送信ノード：各 4byte
- シーケンス番号、主ノード数、隣接ノード数：各 1byte

4.2 評価結果

評価結果を表 4.1 に示す。提案手法を用いることで、トポロジ広告メッセージによる通信負荷を低減できることがわかる。ノードが多く密なトポロジの方がより大きく通信負荷を低減できた。また、メッセージ統合の経過を観察することにより、共通近傍ノード数がうまく働き、異なる

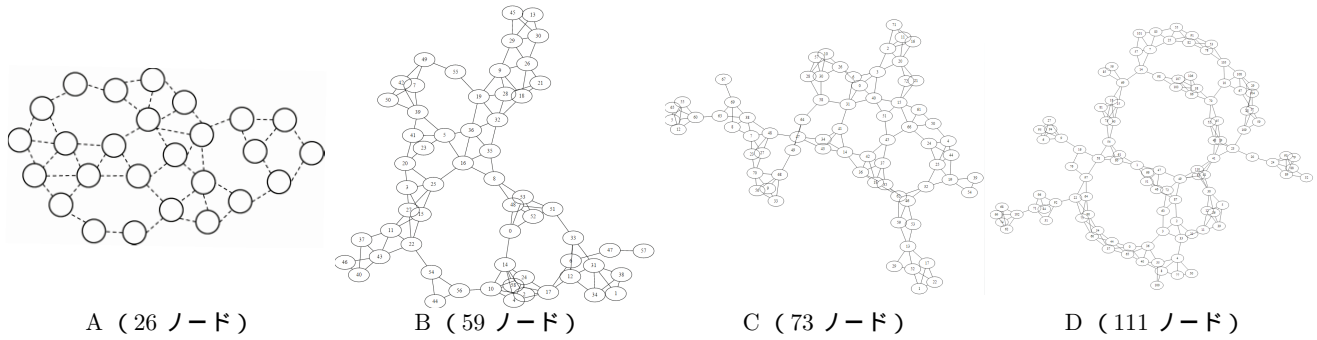


図 8 評価に用いたトポロジ

表 1 評価結果

トポロジ (ノード数)	A (26)	B (59)	C (73)	D (111)
既存手法 (Bytes)	13416	79414	119574	287934
提案手法 (Bytes)	12812	52641	74684	151241
提案 / 既存	95.5%	66.3%	62.5%	52.5%
同一次ホップ割合	98.4%	96.8%	95.7%	95.3%

ノードでも共通のノードペアから順番に統合できていることを確認した。一方、提案手法は必ずしも最短次ホップを選んでいないが、95%以上の高い割合で最短次ホップを選ぶことも明らかになった。さらに、経路ループの存在を調べたところ、トポロジ D において 1 箇所のみ、2 ノードによるループが存在した。

5. おわりに

本研究では、リンク状態型ルーティングプロトコルにおいて複数のトポロジ広告メッセージを統合することで、トポロジ広告メッセージによる通信負荷を低減する手法を提案した。自作シミュレータを用いて、提案手法を評価した。評価の結果、提案手法は、既存手法であるトポロジ広告メッセージを統合せずに定期的送信する場合と比べて、ネットワーク全体で一定時間ごとに送信されるトポロジ広告メッセージの合計サイズが低減できることが明らかになった。また、クラスタ主ノード数上限を設けることで、提案手法は高い割合で最短経路を計算することが分かった。

今後の課題はいくつか挙げられる。提案手法が最短次ホップを選ぶ割合を向上するような計算法の工夫をしたい。関連して、経路ループの生成を完全に防ぐことが実用上は非常に重要であり、このための方法を模索したい。また、本研究ではクラスタ主ノード数上限として線形関数を用いたが、どのような関数が適しているかを明らかにすることも重要な課題である。

参考文献

[1] T. Clausen and P. Jacquet, "Optimized Link State Routing (OLSR)," IETF RFC3626, 2003.
 [2] J. Moy, "OSPF Version 2," IETF RFC2328, 1998.
 [3] E. Baccelli, P. Jacquet, D. Nguyen, T. Clausen, "OSPF

Multipoint Relay (MPR) Extension for Ad Hoc Networks," IETF RFC2328, 1998.
 [4] R. Ogier, F. Templin and M. Lewis, "Topology Dissemination Based on Reverse-Path Forwarding (TBRPF)," IETF RFC3684, 2004.
 [5] D.Oran, Editor, "OSI IS-IS Intra-domain Routing Protocol," IETF RFC1142, 1990.
 [6] F.J. Ros and P.M. Ruiz, "Cluster-based OLSR Extensions to Reduce Control Overhead in Mobile Ad-hoc Networks," In Proc. IWCMC'07, 2007.
 [7] Y. Ge, L. Lamont, "Hierarchical OLSR - A Scalable Proactive Routing Protocol for Heterogeneous Ad Hoc Networks," In Proc. WiMob'05, 2005.
 [8] N. Nikaiein, H. Labiod and C.Bonnet, "DDR-Distributed Dynamic Routing Algorithm for Mobile Ad Hoc Networks," First Annual Workshop on Mobile and Ad Hoc Networking and Computing (MobiHOC), 2000.
 [9] N. Nicaiein, C. Bonnet, N. Nikaiein, "HARP-Hybrid Ad-hoc Routing Protocol," International symposium on telecommunications, 2001.
 [10] B. Guizani, B. Ayeb, and A. Koukam, "A New Cluster-based Link State Routing for Mobile Ad Hoc Networks," In Proc of Communications and Information Technology (ICCIT'12), 2012.