メッシュネットワークにおける分散ルーティング及び OpenFlow による負荷分散の提案

久保田祥太[†1]　　佐藤文明[†2]

近年，無線通信技術の発展により，無線メッシュネットワークが注目を集めている．無線メッシュネットワークは，低コストで広範囲にネットワークが構築でき，拡張性と柔軟性に優れている利点を持つ．しかし，中央での管理機能を持たない構造のため，通信経路の探索や制御を行うルーティング技術が必要とされる．また，規模が大きくなるとホップ数の増加により通信速度の低下や，複数のフローの存在によりフロー干渉が起きる問題などがある．この問題に対し，本稿では SDN 標準の 1 つである OpenFlow 技術を使用して，効率的な経路を構築する方法を提案する．OpenFlow とは，通信ネットワークを構成するネットワーク機器を一つの制御装置で集中管理し，複雑な転送制御や柔軟なネットワーク構成の変更を可能とする技術である．この技術を使用して，経路探索時に負荷を考慮する経路構築を行い，負荷分散を可能にする．また，OpenFlow ネットワークには中央コントローラが存在するため，集中型アルゴリズムが可能となり，従来の分散ルーティングアルゴリズムに比べて優れたパフォーマンスとフレームワークを提供する．しかし，大規模な無線メッシュネットワークの適用が少ないため，本研究では大規模な無線メッシュネットワークで評価を行った．シミュレーションによって評価した結果，低負荷の経路を構築することができ，OpenFlow の有効性を確認した．

1. はじめに

近年、デジタル家電、PC、移動端末などの様々な端末から構成され、将来ユビキタスネットワークを実現する要素技術として、広帯域で、固定の有線ネットワークに依存せず、柔軟なネットワーク構成が可能な無線メッシュネットワーク技術が注目されている[1][2]。

WMN(Wireless Mesh Networks－無線メッシュネットワーク)は、無線 LAN のアクセスポイント同士をメッシュ状に結び、それらの間で自律的に伝送路を構築する無線のバックボーンのことである。WMN は低コストで広範囲にネットワークが構築でき、拡張性と柔軟性に優れている利点を持つ。しかし中央での管理機能を持たない構造のため、通信経路の探索や伝送制御を行うルーティング技術が必要とされる。加えて、規模が大きくなるとホップ数の増加により通信速度の低下や、複数のフローの存在によりフロー干渉が起きる問題などがある。

また、WMN で使用されている HWMP(Hybrid Wireless Mesh Protocol)[3]では、通信遅延のリンクメトリックに基づいてリンク選択が行われる。そのため、大きな通信遅延を有するリンクは、経路選択プロセスによって回避されるが、ネットワークの状況によっては避けることができず負荷が集中する可能性がある。この課題に対して、負荷を分散しつつ経路を構築する LA-HWMP(Load Aware Hybrid Wireless Mesh Protocol)[4]という方法が提案されている。これにより、ネットワーク内に負荷を均等に分けることが可能になる。しかし、LA-HWMP は負荷変動に対して動的に対処することはできないという課題がある。

負荷変動に対応する方法として、SDN(Software Defined Networking)標準の 1 つである OpenFlow[5][6]技術を使用して、負荷分散を行う方法が提案されている[7]。OpenFlow とは、通信ネットワークを構成するネットワーク機器を一つの制御装置で集中管理し、複雑な転送制御を行い、柔軟にネットワーク構成を変更できる技術である。また、OpenFlow ネットワークには中央コントローラが存在するため、集中型アルゴリズムが可能で、従来の分散ルーティングに比べて、優れた性能とフレームワークを提供する。しかし、OpenFlow を使用した動的負荷分散の従来研究では、比較的小規模なネットワーク環境での評価に限定されている。そのため、本研究では大規模なネットワークでの評価を行う。また、OpenFlow を用いた無線ネットワークの制御では、中央コントローラの障害によりネットワーク停止を招くことや、中央コントローラとノード間の通信量が多くなりやすい問題がある。

本稿では、負荷をメトリックとした分散ルーティングと OpenFlow に基づく経路再構築を用いた WMN におけるハイブリッド型負荷分散方式を提案する。これによって、OpenFlow の中央コントローラが障害を生じた場合でも、分散ルーティングによって経路が維持されるようにしている。また、中央コントローラとノード間の通信を減らすため、制御情報は送信元ノードが経路を作った時間でコントローラに通知することとした。また、大規模な WMN においてシミュレーション評価を行い、有効性を評価する。

この論文の構成として、2 章で関連研究である無線メッシュネットワークと OpenFlow について述べる。3 章において提案手法である、負荷分散と OpenFlow に基づく経路再構築の詳細について述べる。4 章ではシミュレーションによる評価を述べ、5 章では本論文の成果をまとめる。

†1 東邦大学大学院 理学研究科 情報科学専攻
Toho University, Graduate School of Science, Department of Information Science.
†2 東邦大学 理学部 情報科学科
Toho University Faculty of Science, Department of Information Science.

2. 関連研究

2.1 無線メッシュネットワーク

メッシュネットワーク[8][9]とは、無線通信機能を持った端末同士がデータを送受信することで形成されるネットワークである。メッシュネットワークはメッシュルータとメッシュクライアントから構成される。各ノードはルータのような役割を持ち、各ノードのデータは複数のノードを順に経由するマルチホップの通信によって目的のノードへと運ばれる。そのため離れている端末同士であってもネットワークに参加している場合において、全ての端末からネットワークに接続することができるので、通信範囲を大幅に拡大することができる。また、ネットワーク内で障害が発生しても、他の端末との通信経路を利用することにより、障害が発生した端末を迂回して通信を行うことができる。

図 1 に無線メッシュネットワーク構成の一例を示している。無線メッシュネットワークにおける、ノードは以下の 4 種類がある。

- MP(Mesh Point)無線メッシュネットワーク機能を構成するために必要なメッシュ機能を持ったノード。本論文では、メッシュルータと呼ぶ。
- STA(Station)はメッシュ機能をもたない従来の無線端末。
- MAP(Mesh Access Point)は MP の機能とメッシュ機能を持たない STA の接続を収容する機能を実装したノード。
- MPP(Mesh Portal Point)は無線メッシュネットワークと外部のネットワークとのゲートウェイの役割をするノード。

無線メッシュネットワークは、これら 4 種類のノードを適宜配置することでネットワークを構築する。

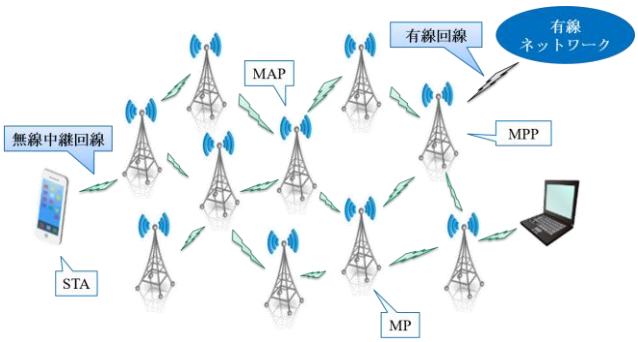

図 1 無線メッシュネットワーク構成の一例

2.2 SDN

SDN[10]とはソフトウェアでネットワークを制御することにより、ネットワークの開発速度の向上と高度化を実現させるコンセプトである。SDN を用いると物理的に接続されたネットワーク上で、別途仮想的なネットワークを構築することで、ネットワークの物理的な制約から離れて、目的に応じたネットワークを柔軟に構築しやすくなる。その結果、負荷の変動に応じて動的にネットワークの構成を変更するといった、制御が可能になる。下記に SDN 構成の一例を示す。

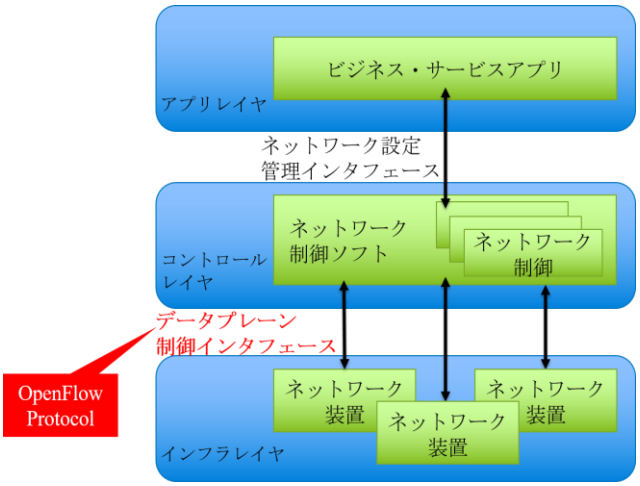

図 2 SDN の構成

2.3 OpenFlow

OpenFlowは、SDNに基づく新しいスイッチングプロトコルである。従来のスイッチと OpenFlow スイッチ(OFS)の構成における大きな違いは、図3のように、ネットワーク全体の経路制御をOpenFlowコントローラと呼ばれる機器上のソフトウェアで集中管理し、OpenFlowスイッチではデータ転送機能のみを実行する。また、物理ネットワーク・仮想ネットワークの両方をコントローラで集中管理することによって、既存のネットワークで実施していた各スイッチでの経路制御の設定が不要となり、ネットワークの単純化と運用及び、管理の負荷の大幅な削減を実現する。

また、コントローラによるネットワークの集中管理により、物理ネットワーク・仮想ネットワーク構成の動的な最適化が可能となる。このことにより、OpenFlowを利用することで様々なサービス形態に合わせて自由にかつ一元的に管理できるようになる。

現状の一般的なスイッチでは、これらは全て一台のハードウェアに組み込まれている。OpenFlow ではスイッチの制御を行うためのネットワークであるコントロールプレーンと、データパケットを転送するためのネットワークであるデータプレーンを分離し、コントロールプレーンでは OpenFlow プロトコルによってスイッチの制御を可能にする。

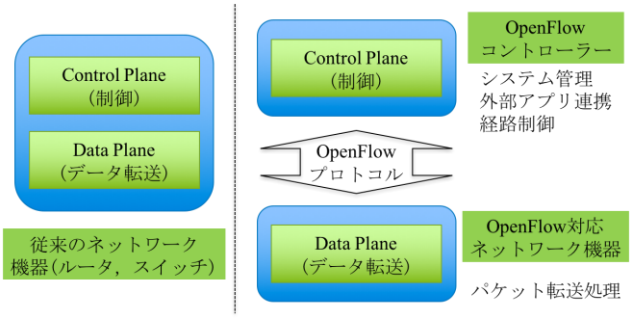

図 3 従来のネットワーク機器と OpenFlow スイッチの構成

2.4 分散ルーティングによる負荷分散方式

分散ルーティングでは、メトリックにノードの負荷情報や隣接するノード情報を含めることによって、負荷の軽い経路や干渉の少ない経路を検索することが可能である。OLSR(Optimized Link State Routing)では、制御パケット上の周辺ノード数の情報を交換して、各ノードの干渉状況を推定する方法が提案されている[11]。この方法では、干渉状況をリンクメトリックの一部として組み込むことによって、干渉を回避する経路を選択することが可能である（図４）。LA-HWMP は、通信帯域の利用率からリンクの負荷レベルを算出し、閾値を超えないリンクを選択し、負荷を分散させる[12]。

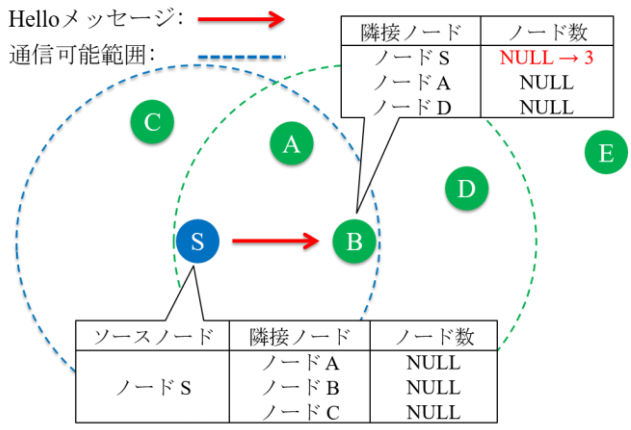

図 4 干渉ノード数の計上方法

分散ルーティングでは、負荷を考慮した経路を選択することは可能であるが、ネットワーク全体の負荷変動に対して動的に対処する研究は少ない。

[14]では経路構築時に経由するノードの負荷情報を制御パケット RREQ に付加して、宛先が複数の制御パケットから最も軽負荷の経路を選び、応答 RREP を送信元に返して経路を作る。更に、データ送信時に経路上の中間ノードがデータパケットに負荷情報を付加し、その情報から宛先が輻輳状態を検知して送信元に RREQ パケットを送出して経路を作り変える方式を提案している。しかし、この方式は輻輳により経路が利用不可能になった場合の再構築を想定しており、負荷のバランスをリアルタイムに維持しようとする方法ではない。また、ネットワーク全体の負荷の状態を考慮する方式でもない。従って、負荷を集中的に管理し、動的に経路を変更する方式を検討する必要がある。

2.5 OpenFlow に基づく負荷分散

OpenFlow を用いて、ネットワークの負荷分散を行う研究が行われている[7][13]。OpenFlow に基づく無線ネットワークの負荷分散[7]では、2 層構成を採用している。第 1 層は B.A.T.M.A.N.(Better Approach To Mobile Ad-hoc Networking)を使用し、第 2 層は OpenFlow を使用するメッシュネットワークである(図 5)。この研究では、メッシュルータに実現されている負荷測定クライアントが負荷の状況を OpenFlow コントローラに通知する。OpenFlow コントローラは、予め決定された経路に従ってトラフィックフローを再転送することによって負荷分散する。しかし、この研究では、6 つのノードからなる 2 つの経路の切り替えでの評価に限定されている。

また、OpenFlow による有線ネットワークの負荷分散の研究 [13]では、実験環境が 3 つの OpenFlow スイッチと 3 つの PC で構成されている(図 6)。したがって、大規模な WMN に対して負荷分散を行うためには、従来の研究では十分ではない。本研究では、大規模な WMN に対して負荷分散を行う新たな経路選択法を提案する。

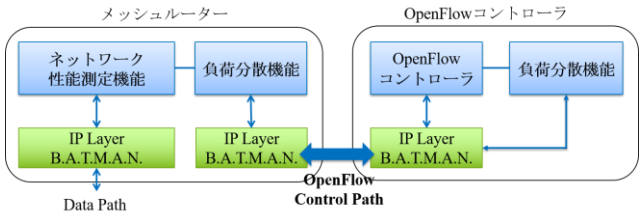

図 5 OpenFlow に基づく負荷分散の構成

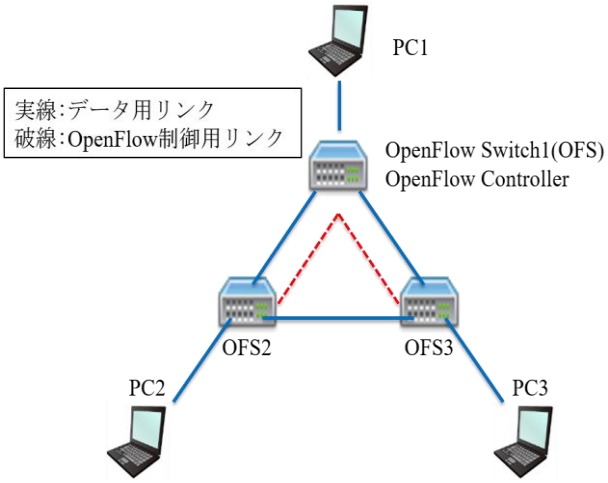

図 6 ネットワーク環境

3. 提案方式

3.1 提案方式の構成

本研究では、負荷を考慮した分散型アルゴリズムである LA-DSR(Load Aware Dynamic Source Routing)と、集中型アルゴリズムの OpenFlow に基づく経路再構築機能をもつ、ハイブリット型負荷分散方式を提案する(図 7)。これによって、OpenFlow の中央コントローラが障害を生じた場合や、無線の経路が輻輳した場合でも、分散ルーティングによって経路が維持されるようにしている。また、中央コントローラとノード間の通信を減らすため、制御情報は送信元ノードが経路を作った時間でコントローラに通知することとした。

OpenFlow コントローラとメッシュルータ間の通信は、データプレーンと同じ DSR(Dynamic Source Routing)を使用して構築する。メッシュルータには、負荷を考慮した DSR 機能とルータの負荷を管理するフローカウンタがあり、フローカウンタに負荷の量が蓄積される。フローの経路は OpenFlow コントローラに通知され、必要に応じて再構成されたフロー情報が、コントローラからルータに送信される。OpenFlow プロトコルは、コントローラとルータ間のフロー情報を交換するために使用される。また、コントローラへのリンクが切断された場合、ルータは DSR 機能を使用して自律的にルートを構築する。

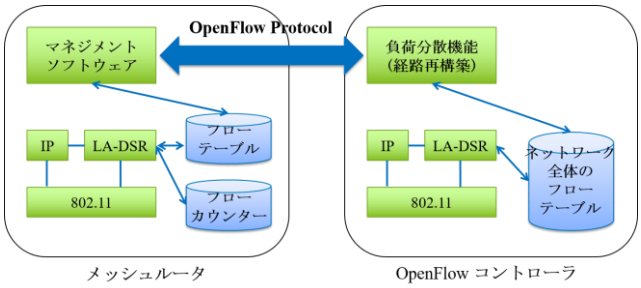

図 7 ハイブリット型負荷分散の構成

3.2 負荷に基づく分散型ルーティング方式

本研究では、Reactive 型ルーティングの DSR プロトコルを採用する。DSR はその単純な機能であるため、それらを容易に変更することができる。DSR は 2 つの制御パケットを使用して経路の構築を行う。送信ノードは、宛先ノードを発見するために RREQ(Route REQuest)パケットをフラッディングし、宛先ノードから送信ノードに RREP(Route REPly)パケットが返される。RREQ には、通過するノードの情報が記録され、経路は RREP によって送信ノードに戻る。基本的に、最も早く到着した RREQ の経路を返すことによって、最短ルートが見つけられる(図 8)。

LA-DSR では、RREQ パケットを転送する際に、負荷の大きさに応じた遅延を挿入する。このため、RREQ パケットは負荷の小さいノードをすばやく通過し、宛先ノードに到達して経路として選択されやすくする(図 9)。RREP パケットが通過するノードでは、フロー情報をフローカウンタに蓄積する。RREP を受信した送信ノードは、フロー情報をフローテーブルに記録し、OpenFlow コントローラに通知する。

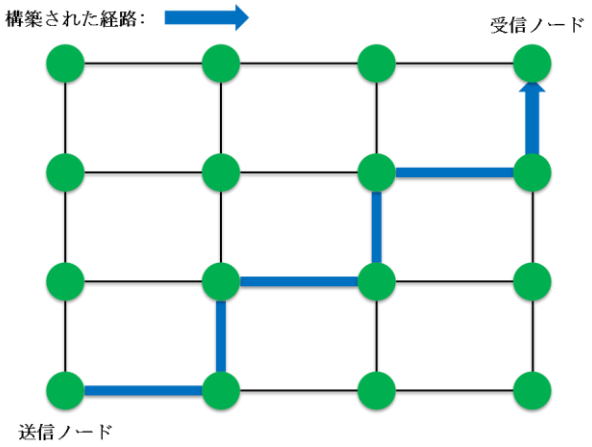

図 8 従来方式の経路

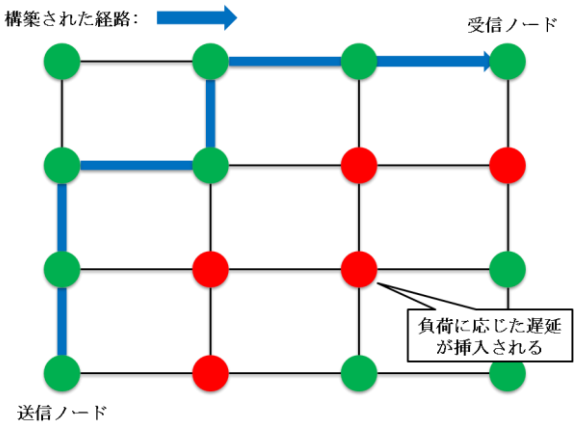

図 9 遅延に応じて構築した経路

3.3 OpenFlow に基づく経路再構成法

OpenFlow コントローラはメッシュルータの負荷分散を行うため、収集したフロー情報に基づいて経路の再構築を行う。フローは送信ノード、宛先ノード、フロータイプ、経路情報を保持している。コントローラは、メッシュルータからフロー情報を受信すると、経路変更を試みる。大規模な無線メッシュネットワークでは各フローの最適な経路を選択することが難しいため、以下のような準最適なヒューリスティック方式を採用する。

最初に、最大負荷のメッシュルータを検出し、最大負荷のリンクを通過するフローの 1 つを選択する(図 10)。次に、フローが削除されたと仮定した場合のネットワークの負荷状況を計算する。その負荷の状況において、負荷をリンクのコストとしてダイクストラ法を適用して、変更後の各経路を計算する。経路変更が適用されたときに最大負荷値が減少した場合、その変更が採用される(図 11)。負荷が軽減

されなければ、経路変更は採用されない。変更が採用された場合、経路情報は変更された経路に含まれる各ノードに通知される。経路再構築のフローチャートを(図 12)に示す。

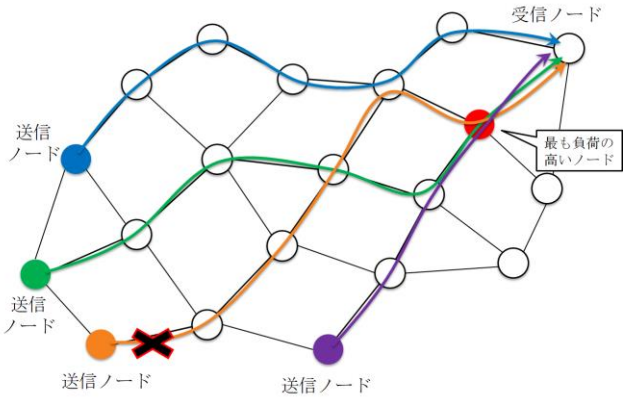

図 10 フローの除去

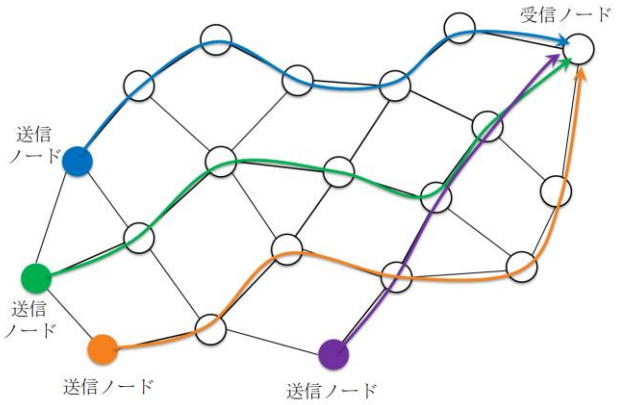

図 11 経路変更後

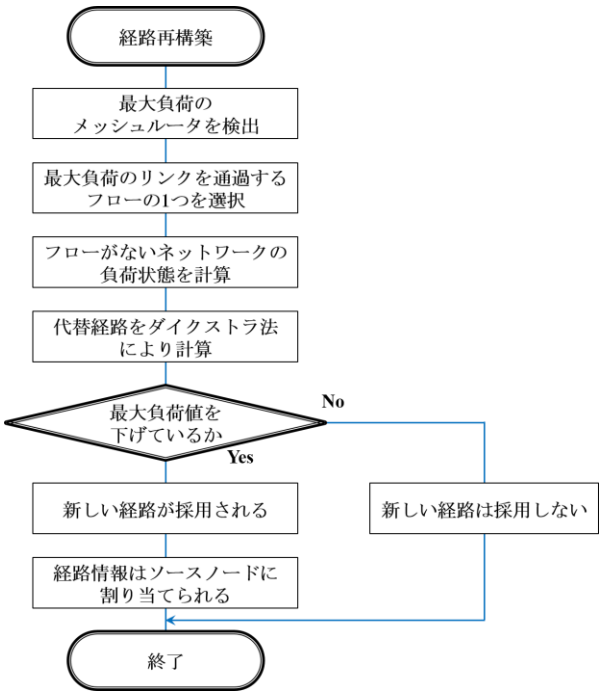

図 12 経路再構築のフローチャート

4. シミュレーション

4.1 シミュレーション条件

本論文では、提案方式の有効性を示すために、従来方式と提案方式 1 と提案方式 2 を比較するシミュレーション評価を行った。シミュレーションはイベントドリブン型のシミュレーションを開発して評価を行なった。ここで従来方式とは、負荷分散を行わず最短経路を使用する方式とする。また、経路探索時に負荷を考慮する負荷分散を加えたものを提案方式 1、これに負荷の高いノード(メッシュルータ)を避け、経路を動的に変更する、ダイクストラアルゴリズムを加えたものを提案方式 2 とする。シミュレーションに用いた通信環境を表 1 に示す。ノード数及びフロー数は可変とした。

表 1 シミュレーションパラメータ

フィールドサイズ	1000[m] × 1000[m]
ノード数	36, 64, 100
ノード配置	格子状
ノード間隔	100[m]
ノード移動	なし
無線範囲	200[m]未満
送信・受信ノード数	10, 10

評価では、従来方式をベースラインとして性能評価を行う。評価指標には、フロー数に対する経路の最大負荷値の平均とホップ数の平均である。ここで、負荷値とはそのノードを通過するフローに流れるデータの総和を示す。このシミュレーションでは、フローに流れるデータ量は同一としており、フロー数に定数を掛けた値とした。

通信要求は、8 個のノードが定期的にネットワークの隅から、ネットワークの隅にある 8 個のノードにデータを送信する。そして途中から 2 個のノードがネットワークの中央からデータを送信し、ネットワーク中央の負荷を高める。

4.2 シミュレーション結果

10×10 ノード配置による環境での経路の最大負荷値の平均を図 13 に示す。図 13 より、各送信ノードが送信するとき、従来方式に対して提案方式 1 は最大で 46 ポイント、提案方式 2 は最大で 53 ポイント削減している。

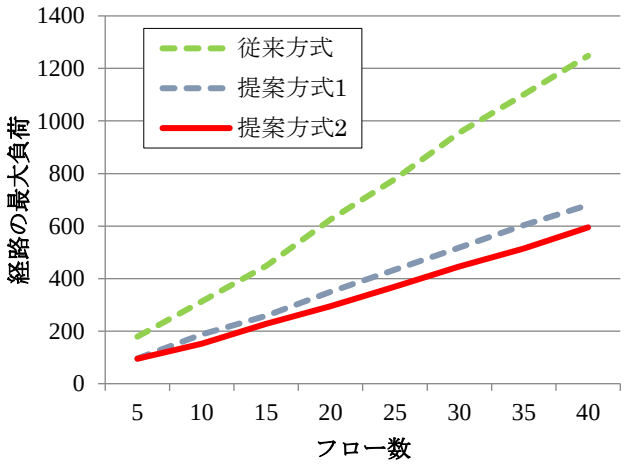

図 13 フロー数に対する最大負荷値の変化
ノード数：100、ノード配置：10 × 10

次に、10×10 ノード配置でフロー数に対する、各経路が受信ノードに到達するまでに要したホップ数の平均を図 14 に示す。ホップ数は増加していないため、それほど遅延が増加していないことが判明した。しかし、フロー数が多くなると、遅延が提案方式 1 よりも増加していることも示した。

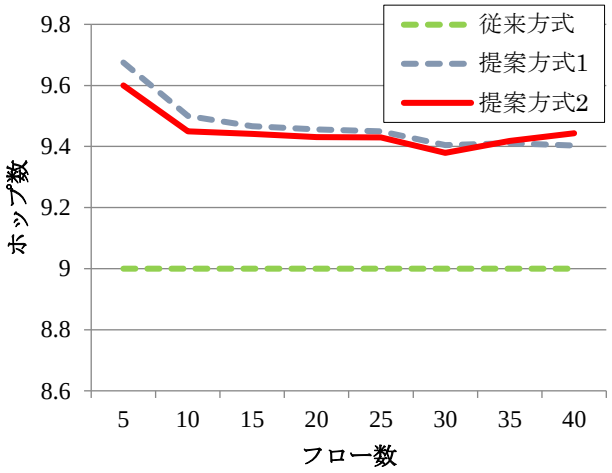

図 14 フロー数に対するホップ数の変化
ノード数：100、ノード配置：10 × 10

次に、ノード配置を 8×8 ノードに変更した環境で、フロー数に対する経路の最大負荷値の平均を示す。この時も、電波到達範囲は 200m 未満でノード間隔は 100m である。

図 15 より、各送信ノードが送信するとき、従来方式に対して提案方式 1 は最大で 45 ポイント、提案方式 2 は最大で 52 ポイント削減している。図 13 と比べて、全体ノード数が減少しても、提案方式 1 と 2 の最大負荷値の削減できているため、ノード数が少ない環境でも最大負荷値を改善することができた。

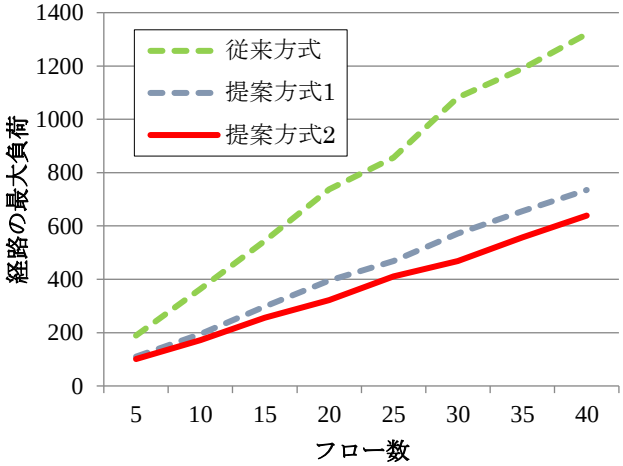

図 15 フロー数に対する最大負荷値の変化
ノード数：64、ノード配置：8 × 8

次に、8×8 ノード配置でフロー数に対する、各経路が受信ノードに到達するまでに要したホップ数の平均を図 16 に示す。図 16 より、全体ノード数を減少させた影響を受け、提案方式 1 よりも遅延が増加している。

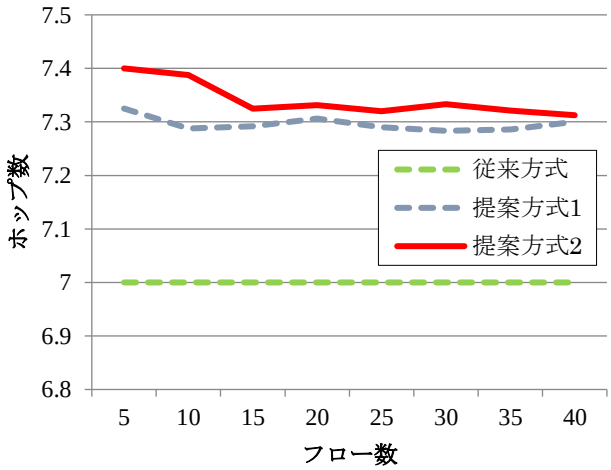

図 16 フロー数に対するホップ数の変化
ノード数：64、ノード配置：8 × 8

次に、ノード配置を 6×6 ノードに変更した環境で、フロー数に対する経路の最大負荷値の平均を図 17 に示す。この時も、電波到達範囲は 200m 未満でノード間隔は 100m である。

図 17 より、各送信ノードが送信するとき、従来方式に対して提案方式 1 は最大で 48 ポイント、提案方式 2 は最大で 52 ポイント削減している。図 13, 図 15 と比べて、全体ノード数が減少しても、提案方式 1 と 2 の最大負荷値の削減できているため、ノード数が少ない環境でも最大負荷値を改善することができた。

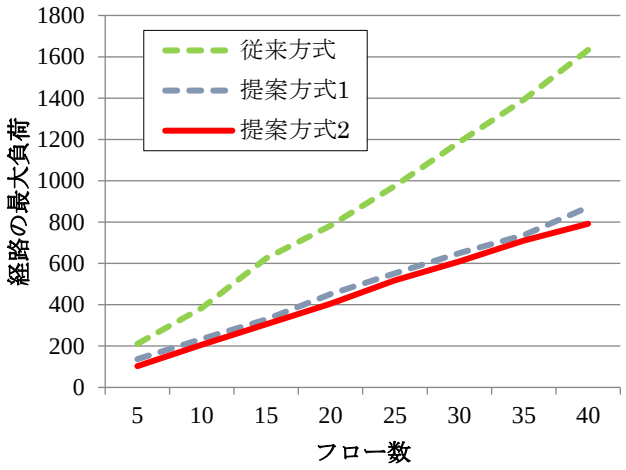

図 17 フロー数に対する最大負荷値の変化
ノード数：36、ノード配置：6 × 6

最後に、6×6 ノード配置でフロー数に対する、各経路が受信ノードに到達するまでに要したホップ数の平均を図 18 に示す。図 18 より、フロー数が増加するほど、遅延が減少している。

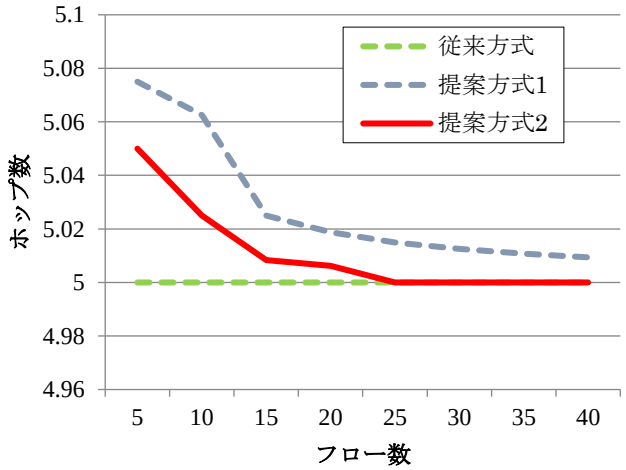

図 18 フロー数に対するホップ数の変化
ノード数：36、ノード配置：6 × 6

5. まとめ

本論文では、大規模な無線メッシュネットワークにおいて、OpenFlow によるノードの負荷を分散するルーティング方式を提案した。従来研究では、ノードの負荷やリンクの干渉をメトリックにして負荷や干渉を分散するルーティング方式が提案されてきたが、経路を利用している際に負荷が動的に変動する場合には対応していない。また、OpenFlow による動的な負荷変動に対応する負荷分散ルーティングの研究もおこなわれているが、その規模は比較的小さいものであった。そのため、本研究ではメッシュルータの負荷をメトリックにした負荷分散ルーティングと OpenFlow を利用した経路再構築を組み合わせたルーティングを提案した。比較的大規模な WMN を想定したシミュレーションの結果、従来方式と比較して経路を増大させることなく、負荷分散ができることを示した。

今後の課題は、Ryu と Mininet を使用した現実的な環境に近い環境において、文献[7]や[14]との比較評価を行うことである。評価環境が実環境に近づくことで、経路の干渉の影響や、パケットロス率、通信遅延などの詳細な評価が可能となる。また、無線メッシュネットワークの管理では、コントローラとメッシュルータ間の通信とデータ通信が無線ネットワークを共有しているため、制御情報のコントローラへの通信が影響を受ける問題がある。この問題の影響を今後評価していく必要がある。

謝辞

本研究の一部は、JSPS 科研費 16H02813 の助成を受けたものである。また本研究の一部は東北大学電気通信研究所における共同プロジェクト研究の支援によって行われた。

参考文献

1) Y. Matsumoto, J. Hagiwara, A. Fujiwara, H. Aoki, A. Yamada, S. Takeda, K. Yagyu, and F. Nuno, “A Prospective Mesh Network Based Platform for Universal Mobile Communications Services”, 信学総大, B-5-245, 2006.

2) 青木 秀憲, 竹田 真二, 柳生 健吾, 山田 暁, “IEEE802.11s 無線 LAN メッシュネットワーク技術”, NTT DoCoMo テクニカル・ジャーナ Vol. 14 No.2, 2006.

3) Xudong Wang, Azman O. Lim, “IEEE 802.11s wireless mesh networks: Framework and challenges, Journal of Ad Hoc Networks”, Elsevier, Vol:6, Issue:6, pp 970-984, 2008.

4) A.Robertsingh, D.Devaraj, R.Narmathabanu, “Development and Analysis of Wireless Mesh Networks with Load-balancing for AMI in Smart Grid,” 2015 Intl. Conference on Computing and Network Communications (CoCoNet'15), pp.106-111, 2015.

5) N. McKeown, T. Anderson, H. Balakrishnan, G.Parulkar, L. Peterson, J. Rexford, S. Shenker, J. Turner, “OpenFlow:Enabling Innovation in Campus Networks”, SIGCOMM, Volume:38, pp.69-74, 2008.

6) Open Network Foundation. OpenFlow – Open Network Foundation, https://www.opennetworking.org/
sdn-resources/openflow, 2015.

7) F. Yang, V. Gondi, J. O. Hallstrom, K. Wang, G. Eidson, “OpenFlow-based Load Balancing for Wireless Mesh Infrastructure,” The 11th Annual IEEE CCNC - Smart Spaces and Wireless Networks, pp.444-449, 2014.

8) K.P. Vijayakumar, P. Ganeshkumar, M. Anandaraj, “Review on Routing Algorithm in Wireless Mesh Networks”, International Journal of Computer Science and Telecommunications, Volume 3, Issue 5, May 2012.

9) W. S. Connner: “IEEE 802.11 TGs Usage Models”, IEEE802.11-04/662r16, January 2005.

10) “Software-Defined Networking:The New Norm for Networks”, OPEN NETWORKING FOUNDATION, April 2012.

11) 朝比奈 啓, 山本 尚生, “無線メッシュネットワークにおける干渉ノード数を考慮した経路構築法”, 電子情報通信学会(IEICE), pp 99-104, 2014

12) A.Robertsingh, D.Devaraj, R.Narmathabanu, “Development and Analysis of Wireless Mesh Networks with Load-balancing for AMI in Smart Grid,” 2015 Intl. Conference on Computing and Network Communications (CoCoNet'15), pp.106-111, 2015.

13) 高橋 裕, 山口 実靖, “OpenFlow による負荷変動を考慮した動的負荷分散の一考察”, 情報処理学会第 76 回全国大会, 2Y-3,pp467-468, 2014.

14) S. J. Lee and M. Gerla, "Dynamic load-aware routing in ad hoc networks," IEEE International Conference on Communications. Conference Record (ICC 2001.), pp. 3206-3210, 2001.