

有線/無線相互補完通信を用いた マルチゲートウェイメッシュネットワークの実装

河合 孔明¹ 遊佐 直樹² 峰野 博史³

概要: 大規模災害の被災地や通信インフラが安定稼働しない発展途上国において、通信インフラを誰でもどこでも簡単に構築できる仕組みが求められている。ネットワークをその場で構築する方法としてワイヤレスメッシュネットワークがあげられる。しかしながら、ノード間通信を無線 LAN のみで行うため、障害物や距離、電波干渉による電波障害が発生する。そこで本研究では、有線 LAN、および無線 LAN をノード間通信に利用可能なマルチゲートウェイメッシュネットワークである Adaptive Multi-Gateway Mesh Network (AMGMN) を提案する。有線 LAN と無線 LAN の両方を利用することで、敷設の容易性を保ちつつ電波障害に耐えられるメッシュネットワークを実現した。また、インターネットへの通信経路の有無に応じて、ゲートウェイノード数を増減させ、特定のノードへの負荷集中を防ぐ仕組みを実装した。AMGMN をブータブル USB に実装することでネットワークの構築を自動化し、構築したネットワークの性能評価を行った結果、Linux 実装の IEEE 802.11s の約 1.4 倍のスループットが得られた。

1. はじめに

大規模災害の被災地や通信インフラが安定稼働しない発展途上国において、通信インフラを誰でもどこでも簡単に構築できる仕組みが求められている。例えば、東日本大震災では約 190 万の固定通信回線が被災した。それらの復旧には約 1 か月半かかったと言われ、通信インフラの迅速な復旧が課題となった [1]。一方、発展途上国では、医療従事者の少ない地域で遠隔医療の試みが始まっているが、頻繁に発生する停電によって通信が安定しないという課題がある [2]。そのため、必要な通信インフラを誰でもどこでも簡単に構築し、安定稼働させる仕組みが必要である。

本研究では、有線 LAN、および無線 LAN をノード間通信に利用可能なマルチゲートウェイメッシュネットワークである AMGMN を提案する。AMGMN では、ノードがインターネットへの接続状況に応じてゲートウェイノード、中継ノード、アクセスポイントノードに変化し、外部ネットワークへのパケット転送設定を自律的に行うことができる。また、AMGMN の実装をブータブル USB 上で行い、ネットワークの構築を自動化することで、誰でもどこで

も簡単に構築できるネットワークを実現する。評価では、ルーティングオーバーヘッドの評価、および敷設が容易なメッシュネットワークの既存手法との比較を行う。

以下、本論文の構成を述べる。2 章で関連研究を紹介する。3 章で AMGMN の提案をし、4 章で実装について述べる。5 章で AMGMN の評価を行い、6 章でまとめとする。

2. 関連研究

ネットワークをその場で構築する方法の一例として、ワイヤレスメッシュネットワークの標準規格である IEEE 802.11s [3] があげられる。IEEE 802.11s によって、ノード間通信に必要な物理的な配線をなくすることができ、場所を選ばないネットワーク構築が可能になる。しかしながら、ノード間通信を無線 LAN のみで行うため、電波障害の影響を受けるという問題がある。

無線通信の電波障害を緩和する手法として、複数のチャネル [4-8] や指向性アンテナ [9] を用いるものがある。特に [4] では、他のチャネルとの重なりを避けるだけではなく、重なりのあるチャネルの中から電波干渉が少ないチャネルを動的に選択することが可能である。しかしながら、これらの手法は距離や障害物による電波障害を防ぐことができない。

より多くの無線通信の電波障害要因に対応する方法として、相互補完通信 [10] や標準規格の IEEE P1905.1 [11] がある。これらは有線と無線が混在するネットワークを構築

¹ 静岡大学情報学部
Faculty of Informatics, Shizuoka University

² 静岡大学創造科学技術大学院
Graduate School of Science and Technology, Shizuoka University

³ 静岡大学大学院
Graduate School of Informatics, Shizuoka University

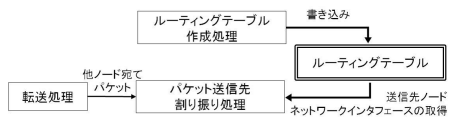


図1 ノードの処理の構成

する手法である。通信障害要因の異なる通信を組み合わせることで、ネットワーク通信の信頼性、安定性を高めることができる。しかしながら、これらの手法では複数のゲートウェイノードを利用することについて言及されていないため、トラフィックが1箇所に集中してしまう可能性がある。

1つのゲートウェイノードにトラフィックが集中することを防ぐ方法としてマルチゲートウェイがある。マルチゲートウェイとは複数のゲートウェイノードを用いて負荷分散を行う技術である。ワイヤレスメッシュネットワークにはマルチゲートウェイを採用したものがある [12–15]。例えば、[12]では、あるゲートウェイノードで輻輳を検知した際、送信量が最も多い子ノードを別のゲートウェイノードに接続させることによって、スループットの低下を防止している。本研究では負荷を分散する方法としてマルチゲートウェイを利用する。

3. AMGMN の提案

3.1 AMGMN の構成

本研究では、有線LAN、および無線LANをノード間通信に利用可能なマルチゲートウェイメッシュネットワークであるAMGMNを提案する。AMGMNにはゲートウェイノード、中継ノード、アクセスポイントノードがあり、それぞれインターネットゲートウェイへのパケット転送、他ノードへのパケット転送、クライアントへのインターネットの提供を行う。インターネットゲートウェイとは、外部ネットワークへの直接的経路を持つ機器を指す。各ノードは図1に示す、ルーティングテーブル作成処理、転送処理、およびパケット送信先割り振り処理を行う。ルーティングテーブル作成処理では経路情報を生成する。各ノードは他のノードとルーティングテーブル作成用のパケットをやりとりし、経路情報をルーティングテーブルに反映させる。転送処理では、他ノード宛てパケットをパケット送信先割り振り処理に渡す。パケット送信先割り振り処理では、ルーティングテーブルをもとに、送信先ノードと送信に使用するネットワークインタフェースを決定し、送信する。ルーティングテーブルはルーティングテーブル作成処理とパケット送信先割り振り処理で共有する。

3.2 AMGMN ルーティングプロトコルの概要

AMGMNではルーティングプロトコルとして、相互補完通信で利用されている Mutual Complementary Communi-

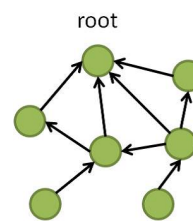


図2 DODAG

cation Protocol (MCCP) をマルチゲートウェイに対応させた、AMGMN ルーティングプロトコルを用いる。本節ではAMGMN ルーティングプロトコルの概要を述べる。

AMGMN ルーティングプロトコルは異なる種類の通信媒体を相互補完的に利用するためのネットワークプロトコルである。AMGMN ルーティングプロトコルのルーティング手法はMCCPやRPL [16]で採用されている Destination Oriented Directed Acyclic Graph (DODAG) ルーティングをベースとしている。DODAG ルーティングとは図2に示すトポロジを形成するルーティングのことである。AMGMN ルーティングプロトコルでは、複数のDODAGを構築し、各ノードがすべてのDODAGに参加できるようにすることでマルチゲートウェイを実現する。以下、DODAG ルーティングの詳細、およびDODAGを用いたマルチゲートウェイの実現方法について述べる。

DODAG ルーティングでは各ノードにランクという、rootノードに対する相対的な距離を表す指標を用いる。ランクの値はrootノードの値を基準として、rootノードから距離が遠いほど増加していく。各ランクは、式1に基づいて計算する。

$$Rank(n) = \min(Rank(p) + RankIncrease(n, p)) \quad (1)$$

式1では、ある近隣ノード p のランクに、ノード n , p 間のメトリックから求めたランクの増分 $RankIncrease(n, p)$ を足した値を求める。すべての近隣ノードを式1の p に当てはめ、 $Rank(n)$ が最小となるときノード p をノード n の親ノード、 $Rank(n)$ をノード n のランク値とする。ルーティングメトリクスについてはMCCPと同様に種々の指標が利用可能であり、想定するネットワーク環境に合わせて定義することが可能である。AMGMN ルーティングプロトコルで採用したメトリックの測定方法、および $RankIncrease$ の求め方は次節で述べる。

DODAGの構築は、3つの制御メッセージ DODAG Information Object (DIO), DODAG Information Solicitation (DIS), Destination Advertisement Object (DAO) をやり取りすることで行われる。DIOメッセージには自身のランク値といったDODAG構築に必要な情報が含まれ、このメッセージをノード間でやり取りすることによりDODAG上の親ノードを決定する。DISメッセージは、近隣ノードにDIOメッセージの送信を要求する際に用いる。DAO

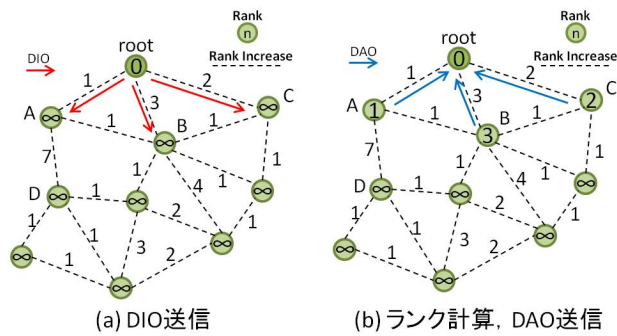


図 3 root から DIO を送信した場合

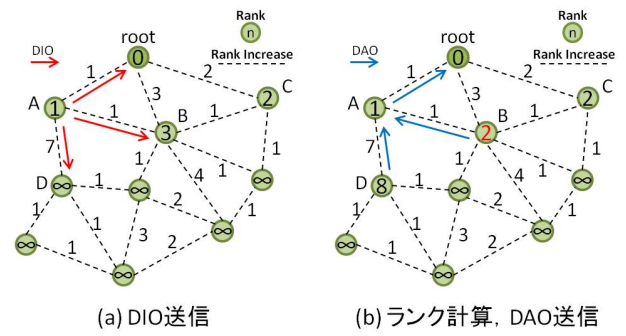


図 4 ノード A から DIO を送信した場合

メッセージは、各ノードの子ノードの情報を root ノードまで伝搬させる。

図 3, 図 4 を用いて DODAG 構築例を示す。図中の丸はノードを表し、丸の中の数字、および各リンクに添えてある数字はそれぞれノードのランク、およびそのパスを選んだ際の RankIncrease を表す。root ノードは最小のランク値を持っており、例では 0 としている。root ノード以外は自身のランクを無限大で初期化している。

まず、図 3(a) のように root ノードが近隣の 3 つのノードに、自身のランク値 0 の情報を含んだ DIO メッセージをブロードキャストする。すると図 3(b) のように、ノード A のランクは $0+1$ で 1 に、ノード B とノード C はそれぞれ 3 と 2 に計算される。このとき、ノード A, B, C は他のノードからの DIO メッセージを受信していない状態であるため、root ノードを自身の親ノードとし、DAO メッセージを送信する。

次にランクが 1 のノード A が DIO メッセージを送信する場合について述べる。図 4(a) のように、ノード A は周囲の近隣ノードに対して、自身のランク値 1 の情報を含んだ DIO メッセージをブロードキャストする。このとき、ノード B は、ノード A からの DIO メッセージによって、自身のランクが $1+1$ で 2 に計算される。この値は、最初に root ノードの DIO メッセージから計算されたランク値 3 よりも小さいため、自身のランク値を 2 に設定し、親ノードとしてノード A を選択する。そして図 4(b) のように、ノード B とノード D はノード A に DAO メッセージを送り、更にノード A は root ノードへ DAO メッセージを送る。以上のように、それぞれのノードがメッセージをやり取りすることによって DODAG が構築される。

DIO の送信周期は、Trickle タイマによって動的に設定する。Trickle タイマとは、ネットワークの通信が安定している間はタイマ設定時間が長くなり、不安定な間は設定時間が短くなるタイマである。各端末の生存確認は一定周期で Hello パケットをブロードキャストすることで行う。

AMGMN ルーティングプロトコルでは、各ノードが複数の DODAG に参加することを許可し、マルチゲートウェイを実現している。DODAG ルーティングでは、root ノー

ドが DODAG ID という識別子を持っており、DODAG ID を DIO メッセージによって下位のノードに伝搬することで DODAG 全体を区別している。各ノードは自分が参加した DODAG とは別の DODAG から受信した DIO メッセージを無視するため、他の DODAG には参加できない。AMGMN ルーティングプロトコルでは DODAG ID による DIO パケットの区別をなくすことで、各ノードが複数の DODAG に参加できるようにし、マルチゲートウェイを実現する。

3.3 メトリック測定と RankIncrease 決定方法

本節ではランク計算に必要となるメトリックの測定方法、およびランクの増分である RankIncrease の決定方法について述べる。AMGMN では、Hello パケット受信時に各リンクの受信強度である Received Signal Strength Indicator(RSSI) を測定し、メトリックに反映する。無線 LAN は比較的通信が不安定になりやすいため、RSSI をメトリックに反映し、有線 LAN は比較的通信が安定するため、RSSI の代わりに最大のメトリック値 100 をメトリックに反映させる。これにより、通信特性の異なる無線 LAN と有線 LAN の安定性を共通の指標で評価することができる。

ノードは RankIncrease の初期値として固定値を持っており、RSSI が閾値を下回った場合に RankIncrease を増加させることで、不安定な経路が通信に利用されにくくする。RSSI の閾値、および RankIncrease の初期値と増分は、想定するネットワークに応じて変更することができる。本研究では、実装の一例として、RSSI の閾値を -50dBm、RankIncrease の初期値を 100、増分を初期値の 2 倍とする。

3.4 環境適応型ノード

外部ネットワークへのパケット転送設定を自律的に行うことでインターネットへの経路を維持させる方法を考える。ゲートウェイノードがインターネットへの通信経路を失った場合でも、ゲートウェイノードを中継ノードやアクセスポイントノードに切り替えることによって、下位ノードのインターネット経路を維持できる。IEEE 802.11s では上記と同様の機能は提案されていないため、AMGMN で新

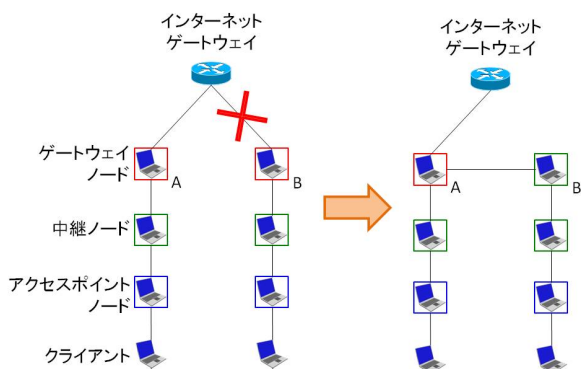


図 5 環境適応型ノードの具体例

たに環境適応型ノードとして提案する。

インターネットゲートウェイへの直接的な経路を持つノードはインターネットに接続されたノードであると判断し、ゲートウェイノードとして機能させる。インターネットゲートウェイへの直接的な経路がない場合は中継ノードやアクセスポイントノードとして機能させる。どちらの機能を持たせるかはネットワークインタフェースの状況によって選択する。ネットワークインタフェースの状況の判定方法は4.4節で述べる。各ノードは定期的にインターネットゲートウェイへの経路の有無、およびネットワークインタフェースの状況を監視し、変化に応じてノードの役割を変更する。

図5に具体例を示す。図5の左側のトポロジでは、インターネットゲートウェイに2つのゲートウェイノードが接続されている。それぞれのゲートウェイノードの下には中継ノード、アクセスポイントノード、クライアントが続いている。ここで、インターネットゲートウェイとゲートウェイノードであるノードBを結ぶ経路が切れたとする。インターネットゲートウェイとノードBを結ぶ経路が切れると、ノードBはゲートウェイノードから中継ノードに自動的に変化し、別のノードと接続を開始する。その様子が図5の右側である。例ではノードBはノードAに接続している。インターネットゲートウェイへの経路が復旧すると、ノードBは再度ゲートウェイノードに変化する。以上の仕組みにより、どのような状況でノードBが親ノードに選ばれても、下位ノードのインターネット経路を維持することができる。

4. AMGMNの実装

4.1 実装環境

AMGMNの実装環境について述べる。本研究ではAMGMNのプロトタイプを表1のUSBメモリに実装することで、さまざまなノートPCを用いたネットワークの構築を想定する。ネットワーク構築に必要な設定をシェルスクリプトによって自動化し、USBメモリにインストールしたUbuntuを起動すると同時にネットワークが自動構

表 1 使用する USB メモリ

項目	詳細
USBメモリ本体	SP Silicon Power MARVEL M01 32GB SP032GBUF3M01V1B
USBメモリ容量	32GB
OS	Ubuntu 12.10 32bit

表 2 ノードとして使用するノート PC

項目	詳細
PC本体	ASUS U24A U24A-PX3210H
CPU	CORE i5
メモリ	4GB
HDD容量	500GB
無線LANモジュール	Atheros AR5BHB92
有線LANモジュール	RTL8111/8168B
本体OS	Ubuntu 12.10 32bit

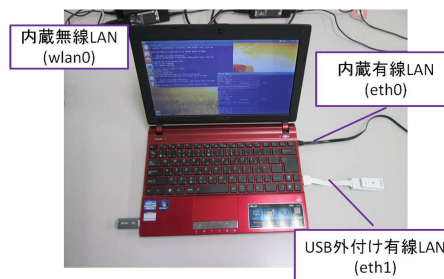


図 6 使用ノート PC とネットワークインタフェースの構成

築されるようにする。

ネットワーク構築には表2のノートPCを使用する。ネットワークインタフェースの構成は図6のものを前提とする。ノートPCに内蔵された有線LAN、無線LANはそれぞれネットワークインタフェース名 eth0, wlan0 として認識させ、後述する AMGMN driver を用いて bond0 というネットワークインタフェースに統合する。bond0 はノード間通信のみに利用する。多くのノートPCは有線LAN NIC、および無線LAN NICを1つずつ搭載していると考えられるため、全ノードの必須機能であるノード間通信には内蔵されたNICを使用する。

ノードのゲートウェイ機能は、有線LAN NICをノートPCに追加して実現する。アクセスポイント機能も実装を容易にするために有線LAN NICで実現する。ノートPCにUSB外付け型の有線LAN NIC(PLANEX UE-100TX-G3)を追加し、eth1として認識させる。

4.2 モジュール構成

図7にプロトコルスタックの構成図を示す。図1で示したルーティングテーブル作成処理は AMGMN routing programとしてアプリケーション層で実装し、パケット送信先割り振り処理は AMGMN driver として実装した。ルーティングテーブルは仮想ファイルを用いて AMGMN driver と AMGMN routing program で共有させた。AMGMN

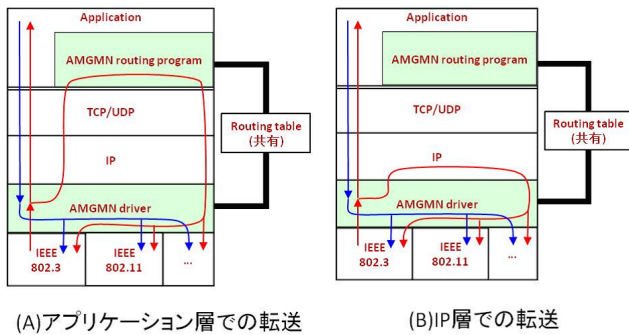


図 7 プロトコルスタックの構成図

driver 内ではパケットの宛先 MAC アドレスの書き換えを行い、MAC 層でのルーティングを行う。AMGMN driver は Linux でチーミングを実現する Bonding Driver を改良したものである。チーミングとは複数のネットワークインターフェースを 1 つに統合することで、ネットワークの負荷分散効果やスループット、信頼性を向上させる技術である。AMGMN driver では有線 LAN インタフェース、および無線 LAN インタフェースを 1 つに統合することで、有線 LAN と無線 LAN が混在するメッシュネットワークを実現する。無線 LAN はアドホックモードに設定しておき、他のノードへの無線経路を構築する。

図 1 の転送処理は図 7(A), (B) の通り、アプリケーション層、または IP 層で実装可能である。本研究ではその両方を実装した。アプリケーション層で転送を行う場合は、AMGMN routing program 内で送受信パケットを監視し、受信パケットの宛先 IP アドレスが自分宛てでない場合は、AMGMN routing program で受信パケットを転送する。転送したパケットは AMGMN driver に渡される。

IP 層で転送を行う場合は、IP 層のフォワーディング機能を用いる。宛先 IP アドレスが自分の IP アドレスでないパケットは IP 層で折り返して AMGMN driver に渡す。いずれの場合も、AMGMN driver は受け取ったパケットの宛先 MAC アドレスを書き換えて、適切なネットワークインタフェースから送信する。

4.3 親ノード選択の実装

AMGMN での親ノードの選択方法について述べる。AMGMN では自身のランク値を最小にするノードを親ノードとして選択する。自身のランク値を最小にするノードが複数存在する場合には、子ノード数が最も少ないノードを親ノードとすることによって、一つのノードにトラフィックが集中することを防ぐ。

子ノード数によって親ノード選択を行う場合には、親ノードが頻繁に変化すること、および複数の子ノードが同時に同じノードを親ノードとして選択することを防ぐ必要がある。親ノードの頻繁な変化は、親ノードの候補となるノード間で子ノード数の大小関係が変化する場合に発生す

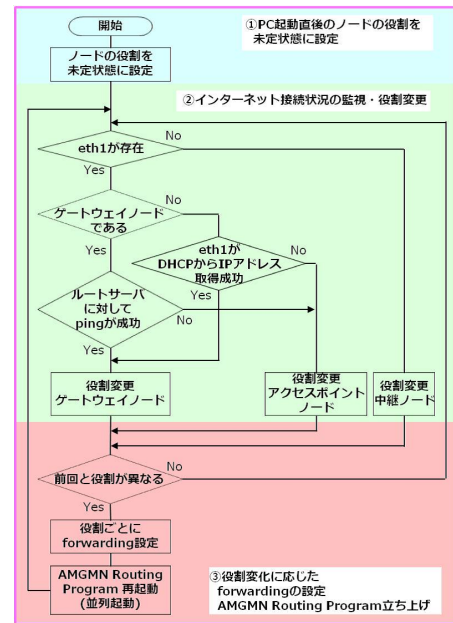


図 8 インターネット接続判定

る。自身の親ノードを新しい親ノード変更した場合に、元の親ノードの子ノード数が新しい親ノードの子ノード数よりも下回ると、元の親ノードを再び親ノードとして選択し直すことになる。これを防ぐために、親ノードの変更を行っても親ノードの候補間で子ノード数の大小関係が変化しない場合のみ、親ノードの変更を適応する。

複数の子ノードが同時に同じノードを親ノードとして選択する問題は、親ノードの変更の許可に確率を用いることによって解決する。ランク値が自分以上の近隣ノード数 n を求め、 $1/n$ を確率とする。各ノードはデフォルトゲートウェイにしているノードの IP アドレス、および自身が持つ子ノード数を Hello パケットに含めて通知することで、互いの子ノード数を定期的に把握する。

4.4 環境適応型ノードの実装

AMGMN ではインターネットへの接続状況とネットワークインタフェース eth1 の有無によってノードの役割を変更する。ノードの役割を変更する手順を図 8 に示す。eth1 が存在しない場合はノードを中継ノードとして動作させる。eth1 が存在する場合には、eth1 からインターネットゲートウェイ内の DHCP サーバにリクエストを送る。IP アドレスの取得に成功した場合はインターネットに接続されていると判断し、そのノードをゲートウェイノードとして動作させる。ゲートウェイノードはインターネットを通じて DNS ルートサーバに ping を送り、インターネットへの経路が維持されていることを判断する。DHCP サーバへのリクエストや DNS ルートサーバへの ping に失敗した場合にはノードをアクセスポイントノードして機能させる。ノードのゲートウェイノードとしての動作、およびアクセスポイントノードとしての動作は、フォワーディングの設

定によって eth1 をルータにすることで実現する．ノードの役割を別のものに变化させる場合は，AMGMN routing program を再起動すると同時に，eth1 に対して再度フォワーディングの設定を行う．

4.5 IP アドレス割り当て

本節では，ノード間通信に利用する bond0 に IP アドレスを割り当てる方法を提案する．IP アドレスを自動的に割り当てる方法として，DHCP や Auto IP があるが，いずれの場合も IP アドレスの重複が無いことを確かめるために全ノードにブロードキャスト送信を行う必要があるため，マルチホップ環境には適していない．本研究では重複の生じにくい IP アドレスを割り当てる簡単な方法として，以下の方法を利用する．

ネットワークを自動構築するとき，MAC アドレスとネットワーク構築スクリプトの実行開始時刻をもとに IP アドレスを自動生成する．具体的には次の通りに IP アドレスを生成する．IP アドレスの第一，第二オクテットは 172.16 とし，ネットマスクを 255.255.0.0 とする．IP アドレスの第三オクテットには使用しているノート PC の MAC アドレスの第六オクテットを割り当てる．IP アドレスの第四オクテットには式 2 で求めた値を割り当てる．式 2 の sec はネットワーク構築用スクリプトの実行開始時刻の秒の値， $msec_{100}$ は 100 ミリ秒台の値， $msec_{10}$ は 10 ミリ秒台の値を示す．式 2 により，IP アドレスの第四オクテットには 240 通りの値をランダムに割り当てることができ，同じ値が生成されるタイミングは 1 時間ごとの 0.25 秒間のみである．第三オクテットは端末の識別に利用される MAC アドレスの一部を用いているため，第三オクテットと第四オクテットを組み合わせることで，重複が起こりにくい IP アドレスを割り当てることができる．

$$4thOctet = sec + 60 * \left(\frac{msec_{100} * 10 + msec_{10}}{25} \right) + 2 \quad (2)$$

5. 評価

5.1 評価環境

実装した AMGMN を用いてネットワークを構築し，スループットを測定する．ノードには 4.1 節で述べたノート PC 5 台を，ルータには表 3 に示すノート PC を用いる．ネットワークの構築に利用する有線 LAN は CAT 5e，無線 LAN は IEEE 802.11g である．スループットを測定するためのソフトウェアである Iperf [17] を用いて，本評価環境の有線 LAN，無線 LAN 単体のスループットを測定したところ，それぞれ約 95Mbps，11Mbps であった．なお，Iperf のウィンドウサイズ，およびバッファ長はそれぞれデフォルトの 8KB を使用しており，以降も利用する．

評価トポロジは図 9 の通りである．ゲートウェイノードから末端ノードまでの各リンクには有線 LAN，無線 LAN

表 3 ルータとして使用するノート PC

項目	詳細
PC 本体	ONKYO DR3A-A25
CPU	CORE i5
メモリ	8GB
HDD 容量	500GB
無線 LAN モジュール	Atheros AR5BHB92
有線 LAN モジュール	RTL8111/8168B
本体 OS	Ubuntu 12.10 32bit

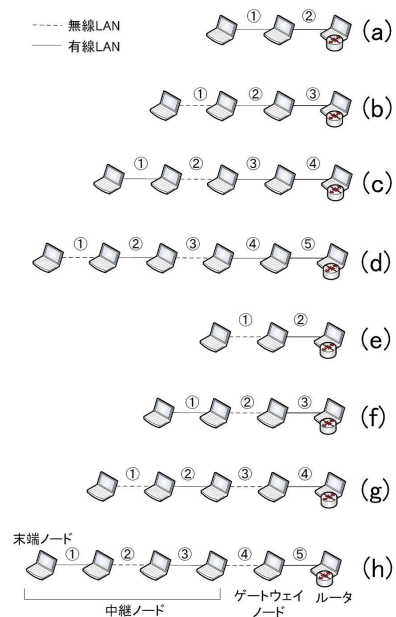


図 9 評価トポロジ

を交互に使用している．評価トポロジには 2 つのパターンがある．1 つは図 9(a) から (d) のようにゲートウェイノードと隣の中継ノードを有線 LAN で接続した場合である．もう 1 つは図 9(e) から (h) のようにゲートウェイノードと隣の中継ノードを無線 LAN で接続した場合である．無線 LAN のリンクを複数利用する場合には，それぞれ別の SSID を割り当て，一直線のトポロジを実現する．

末端ノードからルータまでのスループットを測定する．ルータで Iperf をサーバとして起動し，末端ノードでは Iperf をクライアントとして起動する．末端ノードからルータへ TCP でパケットを送信し，スループットを測定する．各トポロジで 10 回測定し，その平均値を結果とする．

5.2 アプリケーション層，IP 層でのスループットの比較

他ノード宛てパケットをアプリケーション層で転送する場合と IP 層で転送する場合のスループットを比較する．図 9 の全てのトポロジについて評価を行う．

アプリケーション層での転送と IP 層での転送のスループットを比較した結果を図 10 に示す．アプリケーション層で転送を行うよりも IP 層で転送を行った方がスループットが高かった．特にアプリケーション層で転送を行った

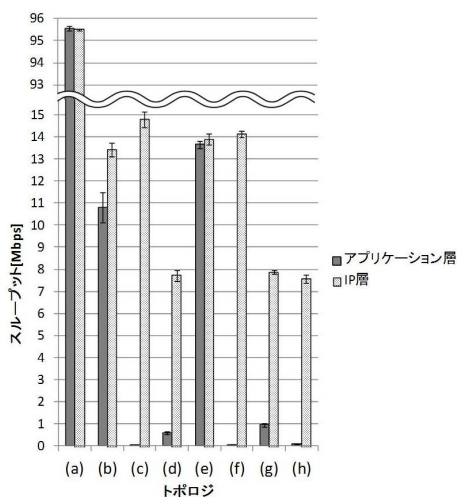


図 10 アプリケーション層，IP 層でのスループットの比較

場合，図 10(g) は 0.95Mbps であるのに対し，図 10(f) は 0.06Mbps となり，ホップ数が減っているにもかかわらずスループットが 1/16 に低下している。図 9(g) と (f) では，末端ノードからの経路が有線 LAN，無線 LAN の順で始まるのか，無線 LAN，有線 LAN の順で始まるかという違いがある。末端ノードからの経路が有線 LAN，無線 LAN の順で始まる場合にスループットが大きく低下すると考えられる。以下，スループット低下の理由について述べる。

図 9(f)①は有線 LAN であるため，95Mbps 前後のスループットが出るが，図 9(f)②は無線 LAN であるため，11Mbps 前後のスループットしか出ない。したがって，有線 LAN を通ったパケットが無線 LAN を通るときに，多くのパケットが送信待ちの状態となる。送信待ちパケットの送信が始まる前に末端ノードで TCP のタイムアウトが発生すると再送が行われる。末端ノードで再送が行われるとより多くのパケットが送信されるため，スループットが低下する。図 9(g) のように末端ノードからの経路が無線 LAN，有線 LAN の順で始まる場合は，図 9(g)①にて 11Mbps 前後でパケット送信が行われる。しかしながら，11Mbps は先に続く有線 LAN，無線 LAN の許容範囲であり，送信待ちが発生しないため，スループットの大幅な減少は発生しない。

図 10 を見ると，スループットの大幅な低下はアプリケーション層で転送した場合のみに発生している。それは，アプリケーション層で転送を行う場合，受信パケットをメモリのカーネル空間からユーザ空間へコピーし，ユーザ空間からカーネル空間へのコピーを再度行うことで転送を実現しているためである。パケットのコピーには時間がかかるため，送信待ちパケットの送信が始まる前に TCP のタイムアウトが発生しやすい。一方，IP 層で転送を行うとメモリコピーが発生しない。そのため，有線 LAN，無線 LAN の順で始まる経路であっても，送信待ちパケットは TCP のタイムアウトが発生する前に送信され，末端ノードからの再送の影響によるスループット低下は起こりにくい。実

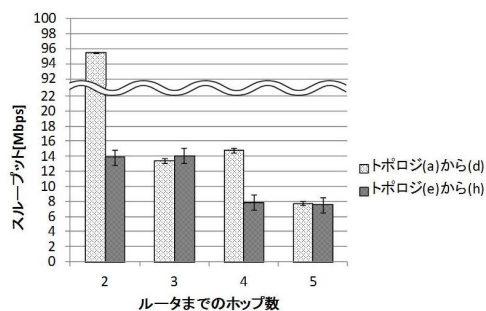


図 11 各ホップ数のスループット

際にパケットモニタリングソフトの Wireshark [18] を用いて末端ノードの送信パケットを分析したところ，アプリケーション層で転送を行った場合には，TCP の再送が何度も発生していることが確認できた。

以上より，アプリケーション層で実装を行った場合には，有線 LAN，無線 LAN の順で始まる経路において大幅にスループットが低下することがわかる。これは有線 LAN と無線 LAN が混在するネットワークで高スループットを得ることにふさわしくない。そのため，本研究では IP 層で転送を行う方式を採用し，以降の評価で使用する。

5.3 ルーティングオーバーヘッドの分析

各ホップ数でのスループットを比較し，マルチホップにおけるルーティングオーバーヘッドについて考察する。ゲートウェイノードと隣接したノードを図 9(a) から (d) のように有線 LAN で接続した場合と，図 9(e) から (h) のように無線 LAN で接続した場合の各ホップ数でのスループットを測定した。

各ホップ数でのスループットを図 11 に示す。トポロジ (a) から (d) の場合とトポロジ (e) から (h) の場合では，ホップ数 2 のときに約 7 倍の差がある。これは有線 LAN と無線 LAN の最大スループットの差によるものである。

トポロジ (a) から (d) に関する結果では，ホップ数が 4 から 5 に変化する際にスループットが 1/2 に低下している。トポロジ (e) から (h) に関する結果では，ホップ数が 3 から 4 に変化する際にスループットが 1/2 に低下している。トポロジ (a) から (d) のホップ数 3 の場合も，トポロジ (e) から (h) のホップ数 4 の場合もスループットは無線 LAN の許容範囲であるが，スループットは低下している。スループットが 1/2 に低下する理由は次の通りである。

トポロジ (a) から (d) では，ホップ数が 4 から 5 に変化する際に無線 LAN のリンクが 2 つになる。トポロジ (e) から (h) では，ホップ数が 3 から 4 に変化する際に無線 LAN のリンクが 2 つになる。トポロジ内に無線 LAN のリンクが 2 つある場合，無線 LAN の RTS/CTS により，片方の無線 LAN は送信待ちとなる。そのため，末端ノードから 1 つ目の無線 LAN リンクから 2 つ目の無線 LAN リンクの

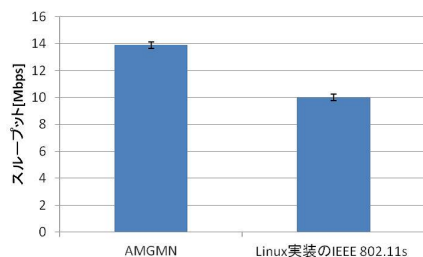


図 12 Linux 実装の IEEE 802.11s との比較

直前まで (図 9(d) の場合は①から②) パケットが伝達された後、パケットの送信待ちが発生する。2つ目の無線 LAN リンクが送信可能状態になると、送信待ちパケットの送信が開始される。有線 LAN についてはパケットの衝突が発生する環境にないため、送信待ちは発生しない。よって、全体のスループットは、2つ目の無線 LAN リンクの直前までのスループットと、2つ目の無線 LAN リンクからルータまでのスループットの平均を 1/2 したものになると考えられる。図 9(d) の場合、①から②と、③からルータまでの平均である 11Mbps を 1/2 した 5.5Mbps と概算できる。

以上のことから、スループット低下の原因は無線 LAN のリンク数、およびネットワークインタフェースの性能限界であるといえる。よって、転送処理時のオーバーヘッドは非常に小さいと考えられる。

5.4 Linux 実装の IEEE 802.11s との比較

Linux 実装の IEEE 802.11s [19] と AMGMN のスループットを比較する。評価には図 9(e) のトポロジを使用する。

Linux 実装の IEEE 802.11s と AMGMN のスループットの比較結果を図 12 に示す。AMGMN のスループットは 13.65Mbps、Linux 実装の IEEE 802.11s のスループットは 9.999Mbps となり、AMGMN によって約 1.4 倍高いスループットが得られることがわかった。Linux 実装の IEEE 802.11s の詳細についてはソースコードを解明中であるが、AMGMN によって Linux 実装の IEEE 802.11s 以上の通信速度を持つネットワークが構築可能であるとわかった。

6. 終わりに

本研究では、有線 LAN と無線 LAN が混在するマルチゲートウェイメッシュネットワークである AMGMN を提案し、USB メモリ上に実装することで、メッシュネットワークの自動構築を実現した。実装した AMGMN を評価した結果、Linux 実装の IEEE 802.11s と比較して約 1.4 倍のスループット値を達成した。今後、複数のゲートウェイノードが存在する場合について、トラフィック量を用いた親ノード選択方式の実装を行い、シミュレーションによって負荷分散効果を評価する。また、最適なリンク選択方法を検討し、RIP や OSPF といった既存マルチパス通信手法との比較を行う。

参考文献

- [1] 総務省.: 平成 23 年版 情報通信白書, pp. 1–11, 2011.
- [2] 黒田 正博.: ボディエリアネットワークにおける広域健診・医療の実現, 月刊 自動認識, 7 月号, Vol. 26, No. 8, pp. 42–50, 2013.
- [3] Hiertz, G. R., et al.: IEEE 802.11 s: the WLAN mesh standard, *Wireless Communications*, IEEE, Vol. 17, Issue 1, pp. 104–111, 2010.
- [4] Liu, Y., et al.: Channel assignment exploiting partially overlapping channels for wireless mesh networks, In *Global Telecommunications Conference, GLOBECOM 2009*. IEEE pp. 1–5, 2009.
- [5] Riggio, R., et al.: Interference and traffic aware channel assignment in WiFi-based wireless mesh networks, *Ad Hoc Networks*, Vol. 9, Issue 5, pp. 864–875, 2011.
- [6] Marina, M. K., et al.: A topology control approach for utilizing multiple channels in multi-radio wireless mesh networks, *Computer Networks*, Vol. 54, Issue 2, pp. 241–256, 2010.
- [7] Dhananjay, A., et al.: Practical, distributed channel assignment and routing in dual-radio mesh networks, In *ACM SIGCOMM Computer Communication Review*, Vol. 39, No. 4, pp. 99–110, 2009.
- [8] Bahl, P., et al.: SSCH: slotted seeded channel hopping for capacity improvement in IEEE 802.11 ad-hoc wireless networks, In *Proceedings of the 10th annual international conference on Mobile computing and networking*, pp. 216–230, 2004.
- [9] Das, S. M., et al.: DMesh: incorporating practical directional antennas in multichannel wireless mesh networks, *Selected Areas in Communications, IEEE Journal on*, Vol. 24, Issue 11, pp. 2028–2039, 2006.
- [10] 遊佐 直樹, 他.: PLC/RF 相互補完通信における環境変化に追従可能な DODAG ルーティングメトリクスの研究, *情報処理学会論文誌 コンシューマ・デバイス & システム*, Vol. 3, No. 1, pp. 77–86, 2013.
- [11] P1905.1 Working Group (入手先 (<http://grouper.ieee.org/groups/1905/1/>)) (2013.09.30).
- [12] Nandiraju, D., et al.: Achieving load balancing in wireless mesh networks through multiple gateways, In *Mobile Adhoc and Sensor Systems (MASS), 2006 IEEE International Conference on*, pp. 807–812, 2006.
- [13] Ito, M., et al.: Proposal and evaluation of multiple gateways distribution method for wireless mesh network, In *Proceedings of the 3rd International Conference on Ubiquitous Information Management and Communication*, pp. 18–25, 2009.
- [14] Amir, Y., et al.: The smesh wireless mesh network. *ACM Transactions on Computer Systems (TOCS)*, Vol. 28, Issue 3, No. 6, 2010.
- [15] Gopalakrishnan, V.: Performance improvement of multi gateway WMN over single gateway, In *Green Technologies (ICGT), 2012 International Conference on*, pp. 52–57, 2012.
- [16] RFC 6550 - RPL: IPv6 Routing Protocol for Low-Power and Lossy Networks (入手先 (<http://tools.ietf.org/html/rfc6550>)) (2013.09.30).
- [17] Iperf (入手先 (<http://iperf.sourceforge.net/>)) (2013.09.30).
- [18] Wireshark (入手先 (<http://www.wireshark.org/download.html>)) (2013.09.30).
- [19] 802.11s - Linux Wireless (入手先 (<http://wireless.kernel.org/en/developers/Documentation/ieee80211/802.11s>)) (2013.09.30).