

# 同時 Join 可能なドローネーオーバーレイネットワークの実装とその評価

大西 真晶<sup>1,a)</sup> 森岡 和行<sup>1</sup> 大和田 泰伯<sup>1,b)</sup> 原井 洋明<sup>1,c)</sup>

**概要:** 我々は超広域災害の被災地域で即時展開が可能な簡易無線メッシュ基地局システムを土台にした拡張性が高いデータ転送システムを開発している。特にリンクの切断・接続の動的な変化に対応し、アドホックにネットワークを構成する高度な能力を有し、さらには、超広域をカバー可能なスケーラビリティを備えた無線メッシュネットワーク構成制御技術は、このシステムの基礎となる重要な要素である。そこで我々は地理位置に基づいた局所分散構成法でありノードの同時 Join に対応したドローネーオーバーレイネットワーク構成法を実装し、無線メッシュ基地局に組込む作業を進めている。本研究では P2P ミドルウェアを用いた本実装の設計について述べ、大規模ネットワークエミュレーション環境上での動作確認試験の結果について報告する。

## 1. はじめに

内閣府 中央防災会議 防災対策推進検討会議 南海トラフ巨大地震対策検討ワーキンググループの最終報告 [1] において、南海トラフ巨大地震の最悪の場合の被害範囲は、「南海トラフ巨大地震では、震度 6 弱以上または浸水深 30cm 以上の浸水面積が 10ha 以上となる市区町村は、30 都府県の 734 市区町村に及び、その面積は全国の約 32%、人口は全国の約 53%を占める超広域にわたるものである。」とされており、また、その発生確率は「文部科学省地震調査研究推進本部における長期評価において 30 年以内の発生確率が南海地震について 60%程度、東南海地震について 70%~80%とされている」とされている。

このような災害が発生した際には、一旦、情報通信ネットワークは機能不全に陥る。先の東北地方太平洋沖地震（東日本大震災）においても、ネットワークサービスが回復するに一月以上を要したが、そこには、政府・自治体や通信事業者による相当な密度での復旧活動があった。それ以上の広範囲に被害が及べば、ネットワークサービスの初期回復フェーズでは、ヒューマンリソース、通信設備、通信路を十分に確保することは難しい。そこで、できるだけ自動的に、また、それぞれの機器や通信路が突発的に機能不全に陥っても、残っている機器のみで情報ネットワー

クを構築することが望まれるであろう。

これを踏まえて著者らは、被災するであろう各地域で常設、または即時展開でき、超広域網も構成可能な超広域災害時展開用情報ネットワークの構築を検討している。特に重要な要素として接続・切断が頻繁であるダイナミックなリンクに対応したスケーラブルなアドホックネットワーク構成技術を挙げる。以下の三つがその要件である。

- 通信機器の場当たり的な設置やバッテリー不足の問題からリンクの不定期な切断が頻繁に起こった場合への対応が可能なこと（ダイナミックリンク要件）。
- ネットワーク管理技能者の被災や交通機関の被災によってネットワーク構築を非管理技能者に委ねる場合でも構成可能なこと（アドホック要件）。
- 災害範囲が超広域である為、事前のクラスタリング設計が困難な状態での巨大ネットワーク構成維持が可能であること（スケーラビリティ要件）。

しかし、従来のアドホックネットワーク構成手法のほとんどは、これら全ての要件を同時には満たしていない。例えば、サーベイ論文である [2] において、アドホックモバイルネットワーク用のスケーラブルなルーティングプロトコルは大きく、Flat routing protocols, Hierarchical routing protocols, Geographic position assisted routing protocols に分類されており、具体的なプロトコルが列挙されているが、各々以下の理由でそれぞれ三つ全ての要件を同時には満たしていない。

- Flat routing protocols は、小ノード数を想定したプロトコルであり、リンク状態をネットワーク全体に通知

<sup>1</sup> 独立行政法人 情報通信研究機構  
Koganei, Tokyo, Japan

a) ohnishim@nict.go.jp

b) yowada@nict.go.jp

c) harai@nict.go.jp

する。この為、全体的にリンク状態がダイナミックであったりノード数が大きくなった場合に制御情報が大きくなり、ダイナミックリンク要件やスケラブル要件を満たさない。これを抑える為に更新頻度や更新範囲について工夫をし得るが、これは事前知識無しネットワーク構成という面から見ると不利な特徴でありアドホック要件を満たさない。この為、三つの要件を同時に満足できない。

- Hierarchical routing protocols は、クラスタリングの結果選ばれた一部の上層ノードのみが同じクラスタに所属するノードの代表となり制御情報をやりとりする、この為、制御情報がノード全体にやりとりされる他、リンク状態の変化によってクラスタが分断された場合に到達性が保障できなくなり、この判定と再構成を必要とするなど、ダイナミックリンク要件を満たさない。
- Geographic position assisted routing protocols では、地理位置情報を用いてのネットワークの構成及びルーティングを行うことによって、制御情報を局所化することができ、原理的には三つの要件を満たすことができる可能性がある。これは GPS や測量網を利用した外部知識の導入に相当するが、比較的誰もが扱いやすい知識であると言える。しかし、[2] 内のプロトコルは、相対的に地理位置が近いノード間のみで制御情報をやりとりし、任意のノードに対して必ずルーティングを行えるネットワークを自律分散的に構成するプロトコルは含まれていない。

現在では地理位置座標を指定して任意ノードへの到達が可能な Geographic position assisted routing protocol として迂回経路を用いることで全ノードへの到達性が保障されていないネットワーク上にドロネーグラフ [3] の構造を持つドロネーオーバーレイネットワーク [4] を局所分散構成する手法が提案されている [5]。また、ドロネーオーバーレイネットワークには地理位置座標を用いて分散的にリンク構造を階層化する手法 [6] も提案されている。ドロネーオーバーレイネットワークは、各ノードから接続されたノードのうちで最も目的地である地理座標に近いノードにメッセージを転送するようなグリーディルーティングが任意ノードを起点として成り立つ為に必要十分な構造を持っており、その制御情報送信範囲の局所性によって三つの要件も満たしている。また任意の連結グラフ状の接続構造を必ずドロネーグラフ構造に変化させる為、ネットワーク上の任意の複数のノードに同時に複数のノードが接続したとしても連結グラフの範囲に収まる為、分散的な同時 Join にも対応可能である。この特徴は被災した各地域で展開されたネットワークを相互接続する際に有用である。また、著者らは過去には、地域インフラとなる固定基地局による無線通信基盤 [7] を提案しており、この基地局上にドロネーオーバーレイネッ

トワークの構成手法を実装することによって、超広域災害時展開用情報通信システムの基盤としたい。

本研究では、[5] で提案されている手法を更に整理し、実装を Java 言語で作成された P2P 構造化オーバーレイネットワーク作成用ミドルウェアである PIAX[8] を使って作成した。迂回経路を用いたドロネーオーバーレイネットワーク構成手法に必要な各機構を単体のノード内で動作が閉じるように実装し、分散性を高めた。また、その実装を 2500 個の仮想ノードを用いた大規模のエミュレーション環境にて展開し、ドロネーオーバーレイネットワークの分散構成に成功することを確認した。更に迂回経路を前提としたグリーディルーティング機構も実装した上で独自に実装したオーバーレイネットワーク層で動作する Ping メッセージによる動作確認も行った。

本論文は以下によって構成される。2章では、参照研究及び実装基盤について述べる。3章では、本実装の構成について述べる。4章では、大規模エミュレーション環境上での動作確認を行う。5章では、まとめについて述べる。

## 2. 参照研究及び実装基盤

本実装にあたって参照した研究及び利用した実装基盤について述べる。

### 2.1 参照研究

無線マルチホップネットワーク上へのドロネーオーバーレイネットワーク構成手法については主に、[5] を参照した。この手法は [4] を拡張した手法である。[4] において新たなオーバーレイ接続を確立する際、ネットワークでの到達性が無かった場合、そのリンクを代替する迂回経路を構成して代替する。これによってローレイヤにおける全ノードへの到達性が保障されていない場合でもドロネーオーバーレイネットワークを構成する手法である。ノードの地理位置という外部情報を活用することで制御情報を局所に抑えることができる為、リンクトポロジーの形状を全ノードが知る必要がある他のネットワーク構成手法にはそぐわないような、ノード数の多いリンク状態が頻繁に変化するような状況でも正常な動作が期待できる。[5] の手法は、[4] において接続を確立する操作を迂回経路構成操作に置き換える手法について述べられている。この為、[4] によるドロネーオーバーレイネットワークの構成が可能であれば、[5] によってもドロネーオーバーレイネットワークが構成可能であるということである。

また、[9] においても迂回経路の利用について言及している。しかし、[9] は前提としている詳細な証明を伴うドロネーオーバーレイを構成手法論文 [10] において、同時 Join については、シリアル Join に変換して構成する方法についてしか言及されていない為、本実装では [5] の手法を採用した。

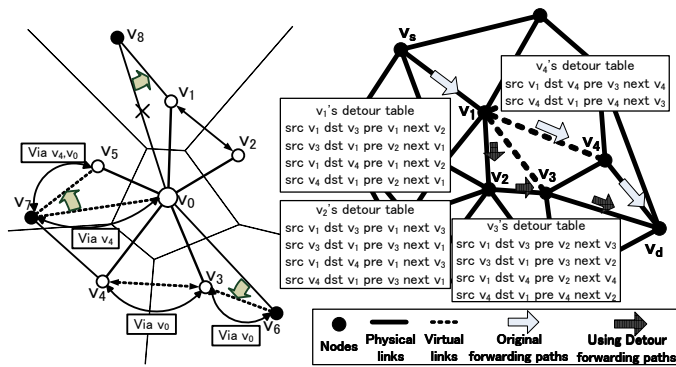


図 1 分散構成法における各ノードの振舞い

## 2.2 ドロネーオーバレイネットワーク分散構成法の概要

本実装で参照する [5] の手法の概要を図 1 左を使用して述べる. 各ノードは通信時の相手の特定に使用できる ID と平面上の位置座標からなる. あるノード  $v_0$  は自身の持つ他のノードの情報を全てを使って, 自身を中心とした局所ドロネー図 [4] を計算する. この局所ドロネー演算によって,  $v_0$  の知るノードは自身のドロネー隣人になる可能性があるドロネー隣人ノード (図 1 左内,  $v_1-v_5$ ) と, ドロネー隣人になる可能性が無い非ドロネー隣人ノードリスト (図 1 左内,  $v_6-v_8$ ) に分けられる.  $v_0$  は, ドロネー隣人ノードリストにあるノードとは接続が無ければ新たに接続を確立し, あれば保つ.

また, ドロネー非隣人ノードには物理リンクで直接  $v_0$  と接続するものも含まれているが, 以降, 隣人ノードが active であるかぎり, その物理リンクを用いて通信することはない. したがって, ドロネー非隣人ノードリストにあるノードとの接続は切断する. この時, 自身と切断されるノードを, そのノードに最も近いドロネー隣人ノードと接続させる (接続委譲). 図 1 左では,  $v_8$  と  $v_0$  の接続を切断する前に,  $v_8$  と  $v_1$  とを接続をさせている.

また, ドロネー隣人ノードの中で局所ドロネー図において隣接関係にあるノード双方にお互いの情報を通知する (三角化通知). 図 1 左では,  $v_1$  と  $v_2$  は,  $v_0$  から, それぞれ, 相手の情報を通知される. この受け取った情報は  $v_1, v_2$  それぞれにおける局所ドロネー演算において使用され, 必要であれば  $v_1$  と  $v_2$  はお互いを接続しようとする.

しかし, ローレイヤのネットワークが全ノードへのルーティングテーブルが整備される前の無線基地局間リンクの集合なのであった場合, 必要に応じたオーバレイリンクの確立は不可能である場合が考えられる. この場合, 接続が必要なノード同士を直接オーバレイリンクによって接続するのではなく,  $v_0$  を経由する迂回経路テーブルを各ノードに設定してやることによってオーバレイ接続の代わりにする.

図 1 左では,  $v_3$  と  $v_6$ ,  $v_3$  と  $v_4$  は, それぞれ接続委譲と三角化通知に基づいて接続されるべきである. しかし直接

接続できなかった場合には, お互いに  $v_0$  に迂回経路情報を保存し, ルーティングの際に利用できるようにして直接接続の代わりとしている.

また, 長大な迂回経路を構成する際は, 迂回経路に使われるオーバレイ接続の一部がより短い迂回経路によって代替されている場合がある. 例えば, 図 1 左の  $v_5-v_7$  接続は,  $v_7-v_0$  を接続する迂回経路を取り込んでおり,  $v_0$  に加えて  $v_4$  を経由する迂回経路を利用して接続されている. これらの迂回経路テーブルの例を図 1 右に示している. dst に向かう為には, 次に next に向かえばよいというテーブルであり, このテーブルに従って残距離に基づくグリーディルーティングを行うと,  $v_8$  から  $v_d$  へは,  $v_1, v_2, v_3$  を経由して到達することになる. 迂回経路で接続する 2 ノードがお互いを送信先として迂回経路構築用広告メッセージを投げ, そのメッセージが通った全てのノードの迂回経路テーブルに迂回経路情報を設定する. 迂回経路構築用広告メッセージを受け取った各ノードは, 後述するプロセス管理システムによって広告元ノードを記録しておき, 広告が送信先に到達したことを確認した Ack を受け取ったタイミングでテーブルに情報を設定する.

またドロネー隣人接続の変化や直接接続が切断されるなどした場合, 迂回経路を削除する. 具体的には, 迂回経路上の各ノードが迂回経路の必要性を各自監視し, 不必要と判断したノードが, 迂回経路に沿って迂回経路削除用のメッセージを流すことで削除する. 各迂回経路作成時に迂回経路構築用広告メッセージを投げたノードが発行した迂回経路 ID を登録しておき, 迂回経路削除メッセージではその ID を指定して, 該当する迂回経路情報を削除する.

## 2.3 実装基盤

本実装を作成するにあたり PIAX[8] を利用した. PIAX (P2P Interactive Agent eXtensions) は, P2P 構造化オーバレイネットワークとエージェント機構を構成する為のオープンソースのフレームワークである. ノード ID とネットワークロケータを関係づけて管理するクラスがあり, このクラスを使って任意の ID のノードに対するメッセージを PIAX のファンクションを使って送受信することができる. また, PIAX には仮想的なノードを一つの実マシン上で複数動かして管理する機構が入っており, 後述する StarBED<sup>3</sup>[11] での大規模な動作実験では, この機構を利用した.

## 3. 実装の基本構成

本実装の基本構成について述べる.

### 3.1 応用想定及び実装指針

本研究における実装は, 固定のマルチホップ無線基地局や, 緊急車両による常に移動する基地局によるマルチホッ

ブ無線基地局上での利用を想定している。

ドローネーオーバーレイネットワークの土台となるネットワークのノード同士は平面上の位置を持ち、一定距離以下のノード同士は接続される。ノード同士は1対1でローレイヤ接続されているものとし、ローレイヤの通信においては1ホップまでしか通信できないものとする。つまりローレイヤにおいてはノード全体のネットワークが形成されておらず、ドローネーオーバーレイネットワークがネットワーク層を担っている。

本実装において、ローレイヤにおける1対1の通信はIPのUDPを利用して行われる。ノード間の全ての制御用メッセージ、シミュレーション計測用メッセージは、Javaのメッセージ送信用APIを介して送信される。

### 3.2 メッセージ送受信機構

本実装はIPプロトコル上へのオーバーレイネットワーク構成機構の一種であり、IPプロトコル層とオーバーレイネットワーク層の中間部分についてはPIAXを利用して実装している。PIAXはJavaで書かれたミドルウェアである為、IPプロトコルを備え、JavaVMが提供された環境全てで動作が可能である。

各ノードにはIPアドレスとノードIDの対が割り当てられており、ドローネーオーバーレイネットワークにおけるノードIDは通信相手を指定できるロケータとして定義されている。通信相手との通信方法は幾通りか考えられる為、ドローネーオーバーレイネットワーク層ではノードIDのみを管理し、PIAXを使った通信先管理機構に、IPアドレスとノードIDの対となるテーブルを持ち必要に応じて検索を行う。この部分を拡張すればIPプロトコル以外の異種のローレイヤプロトコルを同時に利用することも可能である。

各ノードはドローネーオーバーレイプロトコルに従って、制御メッセージを生成し、送信先にIDを指定する形でドローネーオーバーレイプロトコル層からPIAX層に引き渡す。

他のノードから受信したメッセージはPIAX層からドローネーオーバーレイ層に渡され、受信メッセージリストにて管理される。この受信メッセージリストは、ドローネーオーバーレイ層の局所ドローネー演算が行われる際に取り出され利用される。空になった受信メッセージリストは、局所ドローネー演算が行われている間も新たな受信メッセージを受け付け続ける。この新たな受信メッセージは、次回の局所ドローネー演算が行われるまで保存される。今回の実装では、新たな受信メッセージがあった時点で再び局所ドローネー演算が行われる。

### 3.3 ローレイヤ接続の表現

ローレイヤの接続は、他のノードのIPアドレスをノードIDと対にしてリストを作成、格納することで管理される。これらの接続の確立、切断操作はPIAXの機

構を利用して実装した。

### 3.4 プロトコル進行状況管理機構

各通信メッセージは、接続委譲プロトコル、三角化通知プロトコル、迂回経路構成プロトコルの実行プロセスによって生成される。これらのプロセスは、各々が独立しており、他のプロセスから、その動作に影響を受けることがない。

局所ドローネー演算によって上述のプロセスが複数独立して生成されプロセス毎にIDを割り振られて管理される。本実装では、チケットクラスと呼ぶプロセス進行状況管理クラスを作成した。このクラスにおいて、各プロトコルプロセスは、チケットIDを割り振られて、各プロトコルプロセスの進行手順と進行状態を記録するチケットインスタンスによって管理される。送出したメッセージに対する返信メッセージがあった場合、チケットインスタンスに記録された状態を一歩進め、その新しい状態に応じた新たなメッセージを生成して送出する。

また同内容のチケットを二重生成しない様に、チケット生成時に、既に存在するチケットのリストを検索して同様の内容があった場合、生成を取りやめる。

チケットのタイムアウトについても監視している。通信エラーなどの様々な要因によって、チケットの状態遷移が永久に起こらない事態は起こり得る。この為、生成されたチケットのリストは一定時間ごとにチェックし一定時間チケットの状態遷移が見られなかった場合、そのチケットを管理機構から削除する。また、管理機構から削除されたチケットが発生した場合、直ちに局所ドローネー演算を行い必要なチケットの再生成を行う。以上のチケットの状態の遷移は複数のノードで関連するメッセージをやり取りして進行するが、その進捗状況はチケットを発行したノード主体で管理している。

### 3.5 各実行プロセスの実装時の工夫

本実装は各ノードが分散的に独立した各プロセスを行うことで、連結グラフポロジからのドローネーグラフポロジへの収束が起こる。この為、各ノードが個別に行う機構やメッセージ交換に失敗した場合やタイムアウトが起こった場合でも、連結グラフの状態さえ保っていれば再度、各々のノードが各プロトコルを走らせることで、いつかドローネーグラフに収束する。この特性のお陰で、失敗時の細かな状況を把握することなく、タイムアウト時に再度局所ドローネー化プロセスを走らせることで全体的な収束プロセス再開することができ、実装がシンプルかつロバスタなものとなっている。なお、[4]に書かれた各プロセスのうち接続委譲のプロセスのみが、記述どおりに実装すると委譲先のノードへ新規接続を依頼し、その確立を確認したのち、接続を切断するというプロセスになっていた。この場合、

**全メッセージ共通内容** //全メッセージに共通の内容  
 publisherNodeData : NodeData //メッセージ発行者のノードの情報,  
 //NodeDataは, ID, Position(x,y)から成る  
 messageID : long //メッセージ発行時にメッセージ発行者が付与したID  
 sourceNodeData : NodeData //メッセージを直前に送出したノードの情報  
 destinationNodeData : NodeData //メッセージの送信先ノードの情報

図 2 全メッセージ共通内容

**DELEGATION REQUEST メッセージ**//接続委譲先に対する接続受け取り手続き実行依頼  
 publisherNodeData : NodeData //メッセージ発行者のノードの情報, NodeDataは,  
 //ID, Position(x,y)から成る  
 messageID : long //メッセージ発行者が付与したメッセージID  
 sourceNodeData : NodeData //メッセージを直前に送出したノードの情報  
 destinationNodeData : NodeData //メッセージの送信先ノードの情報

**DELEGATION REQUEST ACK メッセージ**//接続受け取り手続き実行依頼の受諾通知  
 delegationNodeData : NodeData //委譲対象となる接続先のノードの情報  
 ticketID : long //接続委譲プロセス管理用に作成されたチケットのID

**DELEGATION REQUEST ACK メッセージ**//接続委譲要求に対するACK  
 ticketID : long //接続委譲プロセス管理用に作成されたチケットのID

図 3 接続委譲関連

委譲先と委譲元のノード間にまたがったプロセス管理が必要になってしまうことが分かった. そこで, この接続切断処理は, 委譲されるノードが主体となって行うように改良することで, 複数のノードに跨るプロセス管理を不要とした. 委譲されるノードは委譲元ノードから委譲先ノードの情報を受け取り, その後は独立して委譲先への接続を確立し, 委譲元への接続を切断する.

また三角化通知と迂回経路構成時に使用される経路の広告は, 受信した各ノードにおいて一時的に保存される. これは迂回経路構成が必要な場合の中継ノード決定や迂回経路を接続する先からの Ack の転送先決定の際に参照する.

### 3.6 メッセージ構造

図 2 から図 5 に全プロトコルで利用されるメッセージのヘッダを列挙し, 各ヘッダの要素を内容を下の記述形式で表す.

**メッセージ名** //メッセージの概略  
**要素名 A** : データ形式 A //説明コメント  
**要素名 B** : データ形式 B //説明コメント  
**要素名 C** : データ形式 C //説明コメント  
 ...

**TRIANGULATION NOTIFICATION メッセージ** //三角化通知メッセージ  
 notificationNodeData : NodeData //通知するノードのデータ

図 4 三角化通知関連

**DETOUR CONSTRUCTION ADVERTISEMENT メッセージ** //迂回経路構築用広告メッセージ  
 srcNode : NodeData //広告の始点となるノードデータ  
 dstNode : NodeData //広告の終点となるノードデータ  
 midRelayNode : NodeData //広告の中央中継点となるノードデータであり,  
 //まずこのノードを目指して転送, 到達後, 終点宛てに転送  
 ticketID : long //迂回経路広告プロセス管理用に作成されたチケットのID

**DETOUR CONSTRUCTION ADVERTISEMENT SUCCESS メッセージ**  
 //広告が終点まで到達した場合の終点からの返答  
 //各ノードでチケットを調べ広告元を探して転送  
 ticketID : long //各中継ノードが広告を転送した際に作成したチケットのチケットID

**DETOUR CONSTRUCTION ADVERTISEMENT FAILURE メッセージ**  
 //広告が失敗した場合の失敗を認識したノードからの返答  
 //各ノードでチケットを調べ広告元を探して転送  
 ticketID : long //各中継ノードが広告を転送した際に作成したチケットのチケットID

**REMOVE DETOUR メッセージ** //迂回経路削除用メッセージ  
 detourPublisherNodeData : NodeData //削除される迂回経路の広告元ノードの情報  
 detourMessageId : long //広告メッセージに割り当たっていたメッセージID

図 5 迂回経路関連

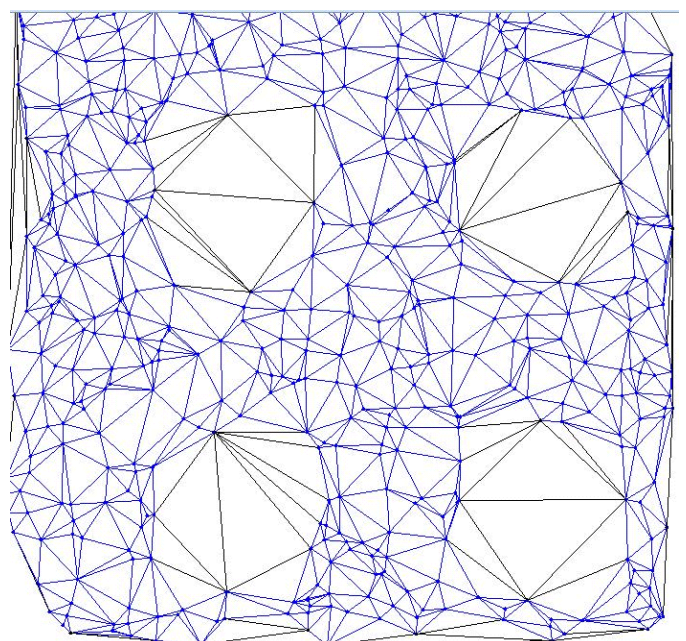


図 6 穴のある分布のビューア画像



## 4. 評価

### 4.1 評価ハードウェア環境

実装の動作確認環境として NICT が持つ StarBED<sup>3</sup>[11] を利用した。StarBED<sup>3</sup> は 1000 台超の実験専用 HaaS 環境による大規模なネットワークエミュレーション基盤である。1000 台の実 PC から構成されており、本来であれば任意のノード間で通信可能である。今回の実験では、通信相手先を制限することで、近傍ノードとしか通信できないマルチホップ無線環境を模倣した。PIAX は 1 台の PC 上で複数の仮想的なノードを動かして別々に動作させることができる。これを利用して StarBED<sup>3</sup> の各 PC の上で 20 個の仮想ノードを動かして実験を行った。各 PC サーバは、Cisco UCS C200 M2 を基本として構成されている。それぞれ管理用ネットワークセグメントと実験環境を想定したネットワークセグメントに接続されている。管理用ネットワークセグメント側から、各ノードに指令を与えて動作させることが可能となっている。

### 4.2 仮想ノードの配置環境想定

各仮想ノードは、1.0\*1.0 の平面空間に配置されているものとする。ノード配置の分布は一様分布と迂回が必要な建造物などを想定した四つの穴があるノード分布の二種類それぞれで実験を行う。配置の穴は、平面空間座標 (x,y) において、 $(0.2 \leq x \leq 0.4, 0.2 \leq y \leq 0.4)$ ,  $(0.2 \leq x \leq 0.4, 0.6 \leq y \leq 0.8)$ ,  $(0.6 \leq x \leq 0.8, 0.2 \leq y \leq 0.4)$ ,  $(0.6 \leq x \leq 0.8, 0.6 \leq y \leq 0.8)$  の範囲に仮想ノードを配置を避けることで作成される。図 6 は、500 ノードで 4 つの穴があるノード配置によるドローンオーバレイネットワーク構成時のビューア表示例である。また各仮想ノードは 0.1 の通信範囲を持ち、それより近い位置に存在する仮想ノードとローレイヤのリンクを保持する。

500,1000,1500,2000,2500 の五種類のノード数それぞれについて各種計測を行う。

更にメッセージを 1 ホップ転送した際の遅延を StarBED のハードウェア環境そのままにした場合と、+10ms した場合、+20ms した場合の三段階設定した。この遅延はネットワークの輻輳時の遅延を仮定している。現在のところ、無線基地局 [7] の基地局間のリンクは Wifi リンクによる構成を想定している。802.11g の理論的な最大速度である 54Mbps であり、後述の実験用 Ping メッセージは約 1500bit で構成されている為、秒間約 300 メッセージ転送することができる。しかし災害応用を想定した場合に、避難所など利用者が密集している場所にある基地局はユーザ端末が発する Wifi 電波からの干渉を受ける可能性がある。この為、干渉が予想される場合には、実際の速度は極度に低下する可能性がある。例えば通信速度が 7Mbps まで落ち込んだ場合、

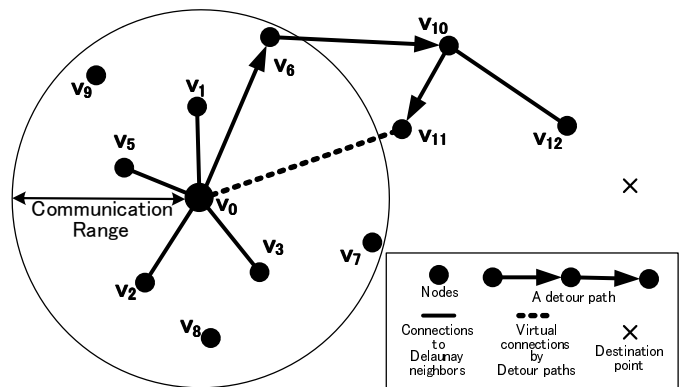


図 7  $v_0$  の利用できる経路例

転送 Ping メッセージ数は秒間約 50 メッセージまで落ち込む。これ以上の数のメッセージが基地局に殺到すると、遅延が起こることとなる。利用者数によってどの程度の遅延が起こるかを予測することは難しいが、瞬間的に遅延が起こりつつも通信が継続可能な状況や DTN 環境への応用も見据えて、以上の遅延を設定した。

### 4.3 評価項目

各ノードは配置後、分散的に独立して動作しドローンオーバレイネットワークを構成する。ドローンオーバレイネットワークの構成が確認できた後、ドローンオーバレイネットワークの疎通を確認する為に独自に設定したオーバレイ層用の Ping メッセージを各ノードから自分以外の全てのノードそれぞれの位置に向かって送信する。この Ping メッセージを受け取った各ノードは Ping メッセージ送信元に対して、Ack メッセージを送信する。この返答である Ack メッセージが到達するまでの時間を各ノードが計測する。

ノードの配置環境毎のレスポンスタイムを比較し、本実装を利用した際の一連の転送処理のボトルネックについて検討を行う。

尚、この比較は各環境における Ping response time のホップ数と平均を用いて行う。また各環境における Ping の最大ホップ数についても確認し、後述のルーティングルールによるノード転送がノード数の増減に関わらず一定のホップ数で収まっていることを確認する。

### 4.4 Ping メッセージの転送ルール

各ノードにおける Ping メッセージの転送ルールについて説明する。Ping は Destination となる平面上の点座標を含んでおり、各ノードはその Destination になるべく近づくようなノードに Ping メッセージを転送していく。この時、各ノードは直接の接続を持つノードと、迂回経路で到達可能なノードの両方の中から最も Destination に近いノードを選び、そのノードに近づくようメッセージを転送する。

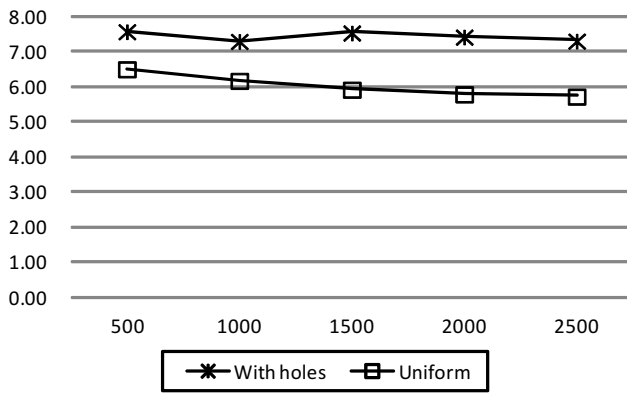


図 8 Ping の平均ホップ数

このルールによってメッセージ内に特に迂回経路上を移動中かどうかを記述せずとも、迂回経路を使用したメッセージ転送が可能である。また、この転送ルールの場合、ある迂回経路を通して転送されている最中でもより Destination に近い他の直接接続されたノードや、迂回経路の終端ノードに向けた転送に自然と切り替わる。これでもより Destination に近いノードに向かってメッセージは転送され続けている為、ネットワーク全体がドロネーグラフ状のトポロジである限り、いつか Destination に最も近いノードまで、Ping メッセージが転送される。

図 7 は  $v_0$  の周囲の転送先ノードを表現した図である。Ping メッセージの転送例を示す。  $v_0$  は、コミュニケーションレンジ内の全てのノードに対して転送を行える。このコミュニケーションレンジの中のノードに限れば Destination に最も近いノードは、  $v_7$  である。しかし、迂回経路テーブル内には、  $v_6$  を経由することによって  $v_{11}$  に到達可能である情報が存在する。この為、  $v_0$  は  $v_6$  にメッセージを転送する。  $v_6$  に転送されたメッセージは更に迂回テーブルに基づいて  $v_{10}$  に転送される。しかし、  $v_{10}$  はメッセージを  $v_{11}$  に転送することなく、より Destination に近い  $v_{12}$  にメッセージを転送する。

尚、迂回経路が途切れてしまったにも関わらず、迂回経路テーブルの情報が更新されなかったノードが存在した場合、メッセージが同じノードの間で永遠に転送され続ける状態になる可能性がある。迂回経路の切断は速やかに検知し、その迂回経路上の全てのノードから、その迂回経路を示すカラムを削除する。

#### 4.5 遅延に応じたレスポンス速度評価

図 8 は、Ping メッセージが Destination に到達する際にかかったホップ数の平均値である。ノード数が増加するにつれて減少する傾向がみられる。これは、ノードが増加するほど、コミュニケーションレンジ内の転送先候補が増加し、通信範囲である 0.1 だけ Destination への距離をより近づけることのできる転送先を見つけることが可能である

Number of Nodes	500	1000	1500	2000	2500
Maximum number of Ping hop counts (Uniform)	18	20	19	16	22
Maximum number of Ping hop counts (With Holes)	23	23	25	27	26

図 9 Ping の最大ホップ数

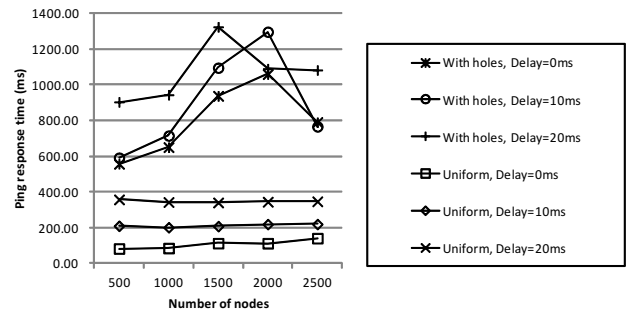


図 10 Ping の平均応答時間

為と考えられる。また穴が存在した場合には、それを迂回するようなルーティングが必要となる為、一様分布に比べてホップ数が上昇していることが分かる。

図 9 は、Ping メッセージが Destination に到達する際にかかったホップ数の環境ごとの最大値である。こちらは、ノード数の増減に対応するような増減傾向が無くほぼ一定している。最低ノード数の 500 ノードの場合でも中継ノードの候補が十分存在した為に、最も遠いノード間のルーティングであっても中継ノード数が一定となったことが伺える。

また穴が存在した場合には、それを迂回するようなルーティングが必要となる為に、一様分布に比べてホップ数が上昇していることが分かる。

図 10 は、Ping メッセージによって Destination からの Ack が返ってくるまでに要した時間の環境ごとの平均値である。概ねホップ数平均の推移に従った増減になっている。しかし一様分布と穴あり分布間のホップ数平均の差が 1 から 2 程度だったのに対してレスポンスタイムは、数倍の上昇している。逆に意図的に加えた転送時のディレイは、レスポンスタイムに対してほとんど影響していない。これは、迂回経路テーブルが増加した際の迂回経路を探し出す処理がレスポンスタイムの増加の大部分を占めていることが言え、よりよい実装の為に経路テーブル検索処理を改良する必要があることが分かる。

## 5. まとめ

本研究では、広域災害時に展開可能なスケラブルなマルチホップ無線ネットワークの実現に向けての、ドロネーオーバーレイネットワークの Java 言語による実装を [5] に基づいて行った。また、StarBED<sup>3</sup> 環境下での動作テストの結果、一通りの動作確認を行うことができ、実装の改良すべき点も明らかとなった。今後は常設の無線基地局への導入、更には移動体基地局への導入に向けて、更なる実装の

洗練を行っていく。

## 参考文献

- [1] 中央防災会議 防災対策推進検討会議 南海トラフ巨大地震対策検討ワーキンググループ, 南海トラフ巨大地震対策について (最終報告) [http : //www.bousai.go.jp/jishin/nankai/taisaku.wg/](http://www.bousai.go.jp/jishin/nankai/taisaku.wg/), 内閣府 (2013).
- [2] X. Hong, K. Xu, and M. Gerla, "Scalable Routing Protocols for Mobile Ad Hoc Networks", IEEE Networks, vol. 16, no. 4, pp. 11-21, (2002).
- [3] B. Delaunay, Sur la sphere vide, Izvestia Akademii Nauk SSSR, Otdelenie Matematicheskikh i Estestvennykh Nauk, 7:793-800 (1934).
- [4] M. Ohnishi, R. Nishide, S. Ueshima, "Incremental Construction of Delaunay Overlaid Network for virtual Collaborative Space," IEEE Creating, Connecting and Collaborating through Computing (C5 2005), pp.75-82, (2005).
- [5] Masaaki Ohnishi, Masugi Inoue, Hiroaki Harai, "Incremental Distributed Construction Method of Delaunay Overlay Network on Detour Overlay Paths," Journal of Information Processing, Vol. 21, No. 2, pp. 216-224, (2013).
- [6] S. Tsuboi, T. Oku, M. Ohnishi, and S. Ueshima, "Generating Skip Delaunay Network for P2P Geocasting," in Proc. of the Sixth International Conference on Creating, Connecting and Collaborating through Computing (c5 2008), pp. 179-186, Jan. 2008.
- [7] Masugi Inoue, Masaaki Ohnishi, Toru Sanefuji, Hiroaki Morino, "Fast Recovery from Link Failures and Black-out of Managed Wireless Mesh for NerveNet," IEEE Global Telecommunications Conference (GLOBECOM 2010), pp.1-6, (2010).
- [8] Y. Teranishi, "PIAX: Toward a Framework for Sensor Overlay Network," Consumer Communications and Networking Conference (CCNC 2009). 6th IEEE Date of Conference, pp.10-13, (2009).
- [9] Simon S. Lam and Chen Qian, "Geographic Routing in  $d$ -Dimensional Spaces With Guaranteed Delivery and Low Stretch," Networking, IEEE/ACM Transactions on Volume. 21 , Issue. 2, pp.663 - 677, (April 2013).
- [10] Dong-Young Lee and Simon S. Lam, "Efficient and Accurate Protocols for Distributed Delaunay Triangulation under Churn," Proceedings IEEE ICNP 2008, pp.124 - 136, (2008).
- [11] StarBED<sup>3</sup>Cubic [English web site](http://StarBED.nict.go.jp/en/index.html)  
<http://StarBED.nict.go.jp/en/index.html>