

## メッシュ網上へのドロネーオーバーレイ網構築 に向けた迂回経路分散構成手法の検討

大西真晶<sup>†</sup> 井上真杉<sup>†</sup>

広域大規模災害発生の瞬間と直後に通信網が平常時に近い形で健全であることは、被害状況の把握、二次被害の低減に有用であることは明白である。しかし、大規模災害のダメージは網の広域各所に予測不可能に分散して現れる為、これに対応可能な迂回路網を多数持つネットワーク構造を取り込んだ設計を行わなければ健全を保つことは困難である。筆者らは小さな回路を多数連結した物理リンク構造を持ち構成機器が同種の基地局のみで構成された広域にまたがるメッシュアクセス網を大規模災害に耐えるアクセス網と成り得ると考えている。そこで本研究では、非常に広域な無線メッシュ網に使用可能な分散局所的な経路情報の同期のみによる分散ネットワーク構成法として無線メッシュ網上へのドロネーオーバーレイネットワークの分散構成手法について検討する。ドロネーオーバーレイネットワークではドロネーグラフのトポロジを模倣することでジオメトリックルーティングの完全な経路到達性を目指しており、また位置情報の補助を用いた分散的な構成手法が可能である。

### A study of the distributed detour path construction methods for Delaunay overlay networks on mesh networks

Masaaki Ohnishi<sup>†</sup> and Masugi Inoue<sup>†</sup>

In catastrophic disaster situation, it is important keeping normally situation like network's usability for checking on the damage. But, catastrophic disasters give the dispersed damage to the whole network. So, keeping connectivity of a whole network is difficult. But, if network has many detours, keeping connectivity of a whole network is easier. So Therefore, we think that the wide area mesh network can resist the damage from catastrophic disasters. So, In this paper, we study of the distributed detour path construction methods for Delaunay overlay networks on mesh networks. Delaunay overlay network can be constructed by a distributed method with geographical positions. And, the network can geometric long-distance greedy routing.

### 1. はじめに

2011年3月に東北地方太平洋沖地震が起こった。日本で三例目の震度7を記録する大地震の後、沿岸部広域に数十メートルの津波が押し寄せるといふ二次災害が起こり、死者・行方不明者が2万5000人を超えるという大規模な人的被害をもたらした。このような広域大規模災害発生の瞬間と直後に通信網が平常時に近い形で健全であることは、被害状況の把握、二次被害の低減に有用であることは明白である。しかし、広域大規模災害においてダメージはインフラシステム全体の各所に分散して無数に現われ、通信のみならず交通網、送電網をも瞬時に寸断する。交通網と送電網の寸断は、分散的な構造となっておらず完全自動化されていない通信網の復旧を非常に困難にする。例えば1995年の兵庫県南部地震においては、被害の大きかった地域では電柱設備の2割以上が倒壊・破損し架空線を寸断しアクセス網が無効化した。また、停電が2時間で復旧したにも関わらず被害の軽微な地域の通信網であっても1日以上復旧しない場合があった。これは網の管理が人の手を前提としたものであり、交通機関が寸断されるような大規模災害時には人の手によるシステムの復旧に労力を要した為である。また用意されていた非常用電源が動作しない例も随所に見られた[1]。このような広域大規模災害に望んでも平常時と同様の健全性を保つネットワークを設計する場合、以下の要件を満たすことが重要である。

- ネットワーク全体が同質の機器で構成され、ハード的な単一障害点が無い。
- 網構造に無数の回路が存在し、無数の物理リンク切断に際しても物理的なネットワークトポロジの連結性を喪失しない。
- ネットワーキングプロトコル、ルーティングプロトコルが災害ダメージによる通信リソース、リンク品質の変動に対しても対応可能であるような高い分散性、スケーラビリティを持つ。
- システム維持コストの低減の為、平常時に利用するシステムの一部として動作し、その利用に耐える通信性能を発揮する。
- 機器の設置後、ネットワークの構成、管理などが自律分散的に完全自動で行われる。

図1では、我々がNerveNet [2][3]と名付けて研究開発を進める無線メッシュ網を中心とした地域サービスプラットフォームの広域展開シナリオについて平常時の構成と非常時の構成について簡略に示したものである。上部は平常時の構成を示しており、下部は大規模災害時の構成を示している。各黒点は、インフラストラクチャ型無線メ

<sup>†</sup> 独立行政法人 情報通信研究機構  
National Institute of Information and Communications Technology

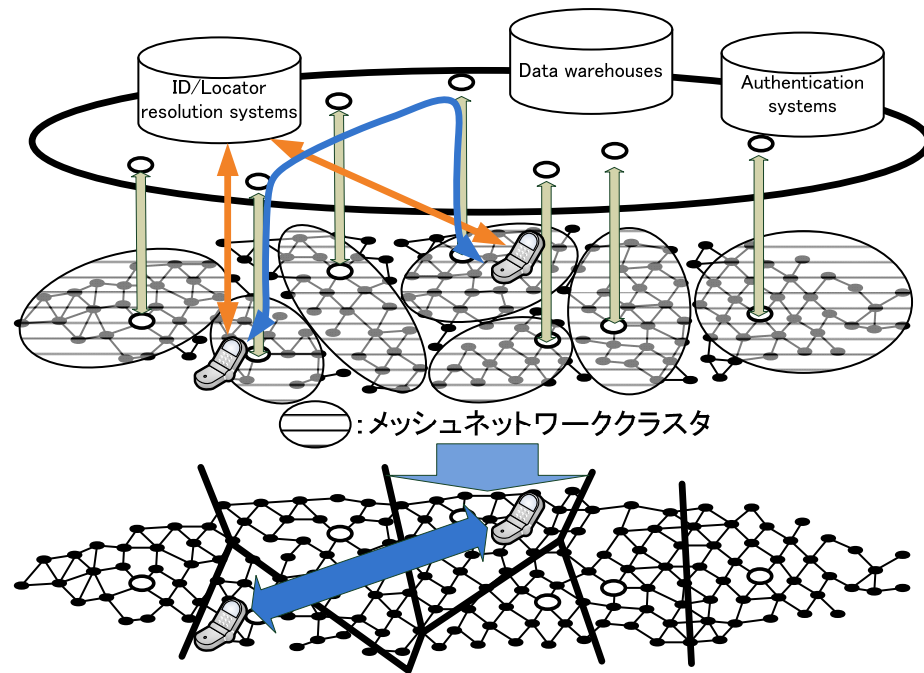


図 1 基幹網から独立した複数メッシュ網の結合による広域メッシュ網

メッシュにおける単種類の基地局，その間のエッジは，基地局間の物理リンクを示している．上部の白い楕円はバックボーンネットワークを示しており，いくつかの点から各メッシュクラスタに対して接続されており，その接続関係を双方子矢印で表現している．その上のデータベースは，ID/locator 解決やデータ管理，認証情報などの為の各種データベースを示している．平常時におけるメッシュ網は十分な性能を発揮できる程度のサイズの大きさのクラスタに分けられている．各クラスタは幾つかの基地局から有線を主体とするバックボーンネットワークに接続されており，バックボーンネットワークを介して他のネットワークへの通信を行う．また，広域における端末位置などの情報やネットワークの利用に関する様々な基本的な機能についてバックボーンネットワーク上のデータベースに問い合わせ平常時の通信情報サービスを提供する．平常時，各クラスタ内においては地域見守りや地域広告の配信などの地域に密着したサービスを行うことを想定しており，また設置後の機器のコンフィギュレーションを完全自動で行うことを目指している．対して，もし大災害によってバックボーンネッ

トワークが不全となるような非常事態になった場合，隣接のメッシュ網クラスタ同士が結合して大きなクラスタとなり，非常に広域を一つのクラスタとしてカバーする．各基地局は L2 スイッチを用いてネットワークを構成するが，その時，多経路を選択可能とするようにツリー状の VLAN を多数設定する．これにより多経路を用いたデータ転送を可能とする．各基地局はデータベースを持ちクラスタ内の端末の所在情報や認証情報などのネットワークを機能させるための制御情報をクラスタ内の全基地局間で交換し管理する．平常時は各クラスタ内の基地局数は少なく，制御信号のフラグディングのトラフィックが問題にならない程度の基地局数でクラスタを構成し，経路切替などの制御を行う．しかし，非常時の場合，非常に数千台規模の基地局がひとつのクラスタとして接続されることを想定している．通常大規模なネットワークのスケラビリティは，クラスタリングにより幾つかのノードをまとめ，それらをサブネットワーク化し，サブネットワーク内の代表ノード間のみで制御信号を交換することによって帯域的な制御信号を削減する．これを応用し広大な無線メッシュにおいても自動的なクラスタリング手法を導入し，代表ノードによる制御信号送信の為の広域ネットワークを構成することが考えられる．しかし，無線メッシュ網においては，各リンクは瞬間的な切断が頻繁に起こる．また，大災害直後であれば，基地局の配置や通信リソース，リンク状態などが綿密に計算して設置された場合とは変化していることが考えられる．このような場合にクラスタリングを自動で行ったとしても適切なクラスタリングというもの定義の困難が予想される他，この手法によって制御信号を削減した場合にも，各ノードからの制御信号はネットワーク全体に到達することには変わらない．各リンクの瞬断時に頻繁に経路切り替えの為の制御信号が送信された場合にネットワーク全体の経路情報を一貫して保つことができるかは未知である．そこで，この広域メッシュネットワークにおいては，非常に局所の制御信号の流通により，ネットワークング，ルーティングが可能となるようなジオメトリックルーティング用ネットワークを構成し利用する．具体的にはドローンオーバレイネットワーク[4]の構成手法を拡張し，無線メッシュ網上への分散的な構成手法を考案し用いる．本論文では，この無線メッシュ網上へのドローンオーバレイネットワークの迂回経路トンネルを用いた構築方法について検討する．2章では，無線メッシュ網上へのドローンオーバレイネットワークの概要，3章では，迂回経路の分散生成法，4章では，シミュレーションによる評価，5章にて関連研究を示し，6章でまとめについて述べる．

## 2. 無線メッシュ網上へのドローンオーバレイネットワークの概要

### 2.1 ドローンオーバレイネットワークの利点と構成手法のアイデア

ドローンオーバレイネットワークは，ノード位置とユークリッド距離に基づくグリ

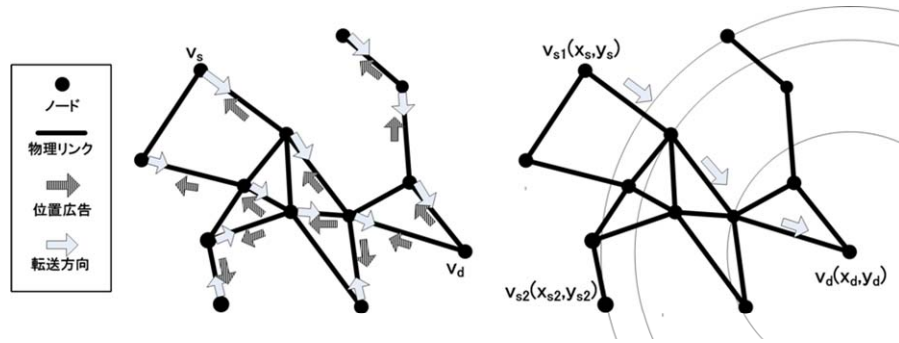


図2 一般的なルーティングとジオメトリックルーティング

ーディなジオメトリックルーティングの到達性を保証する為のオーバーレイネットワークである。図2に、一般的なルーティングとジオメトリックルーティングについて示した。まず、左側は一般的なルーティング手法について示している。一般的なルーティング手法では多くの場合ネットワークは離散的なグラフのトポロジで表現される。各ノードは自身の位置をネットワーク全体に対してマルチホップ転送によって広告する。図では、転送先  $v_d$  が自身の位置をネットワーク全体に広告しており、その広告の流れを黒矢印で示した。この広告を受け取った各ノードは広告を転送してきた  $v_d$  に近い上流ノードを記憶しておき、 $v_d$  宛てのデータが自身のところに回ってきた場合、これら上流ノードに転送を行うことで  $v_d$  にデータが到達する。対して、右側のジオメトリックルーティングにおいては、各ノードに平面上の位置を与えておく。この時平面上の位置を与える方法については様々な方法が考えられるが、無線マルチホップネットワークにおいては、緯度経度などの地理的な位置情報が一般的である。各ノードは、自身の 1HOP 隣接のノードにのみ自身の位置を送ることでネットワークを行う。ルーティング手法は位置に基づいた様々な手法が考えられるが、ここでは例として、ユークリッド距離を短縮していくグリーディルーティングについて示す。まず転送先を平面上の位置座標で与える。各ノードは、自身の隣接ノードのうち、その座標へのユークリッド距離が最も近いノードへとデータ転送する。例えば、図では  $v_{s1}$  から白矢印に沿っての転送が行われ転送先  $v_d$  まで転送される。この様にグリーディルーティングは、1HOP 先までの各ノードの位置情報の通知という制御信号によってネットワークを完了し、遠距離までのルーティングを可能とする。但し、特に工夫のないネットワークとグリーディルーティングでは、必ずしも送信先までデータが転送されるとは限らない。図の場合、 $v_{s2}$  からスタートした転送は、 $v_{s2}$  と接続された唯一のノードが  $v_{s2}$  よりも転送先  $v_d$  から遠い為に転送が起こらない。1HOP という局所の

制御信号のやりとりでネットワークできるが到達性の保証が無いので使用が難しいといえる。しかし、このような平面上の位置に基づくグリーディルーティングによる任意の二ノード間における到達性を保証するグラフとしてドロネーグラフが知られており、[4]において、ドロネー図状のオーバーレイネットワークを P2P ネットワークとして増加分散的に構成する手法が提案されている。また、この構成手法は連結グラフを対象としたネットワークング手法であり、汎用性が高い。

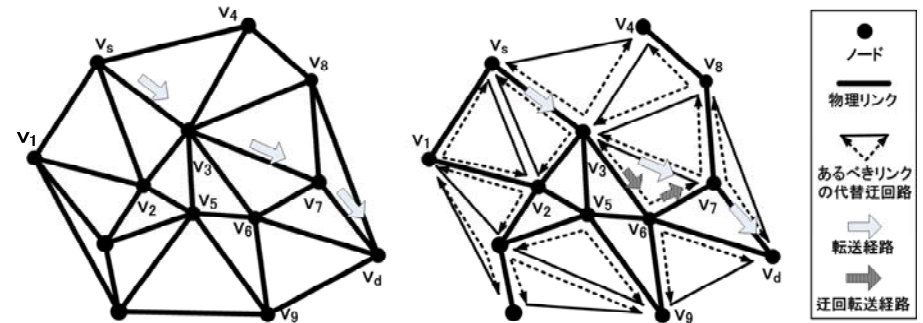


図3 ドロネーグラフと迂回経路によるドロネーグラフ

図3は、ドロネーグラフと迂回経路によるドロネーグラフを示した。ドロネーグラフは三角形分割図となっており各三角形の外接円に他のノードが一つも入らない三角形分割となっている。このような三角形分割図は、ひと通りしか描くことができず、ドロネーオーバーレイネットワークの接続方法はノード位置に対して一つに定まる。このドロネーオーバーレイネットワークのリンク集合はグリーディルーティンによって任意の送信先に到達する為に必要十分な集合となっており、例えば、このような物理リンクトポロジを用意することでネットワークのルーティングの理屈の上では完全に成功させることができる。[4]における構成手法では、このようなネットワークトポロジを周囲6ノード程度と情報交換することで構成することができ、ルーティング手法と同様に構成手法として局所性を持つ。しかしメッシュ網上においては例えば右のグラフのようにドロネーグラフとしての物理リンクが無い場合が考えられる。そこで本研究では、迂回するような形の仮想リンクとなるトンネル接続を接続すべき二つのノード間に構築し、物理リンクと迂回トンネルによるドロネーオーバーレイネットワークを構成することを考える。例えば図3の右側では、 $v_3$  と  $v_7$  の間に  $v_6$  を経由するようなトンネルを張って、ドロネーネットワークとなるような接続を確立する。[4]における構成法では、 $v_3$  と  $v_7$  の間にオーバーレイ接続を確立する必要があった場合、その二者の位置を知る  $v_6$  が  $v_3$  と  $v_7$  に通知し、直接的なオーバーレイトンネルの確立を行った。し

かし、本研究の前提では、まだネットワーク全体はローレイヤにおいてもローションを使用している通信可能となっている保証は無い。そこでローレイヤは物理リンクが接続されたノード同士しか通信できないと仮定し、それらのリンクをまたがるトンネルを構成することでドロネーオーバーレイネットワークを構成する。

### 3. 迂回経路の分散生成法

#### 3.1 準備 (UDG)

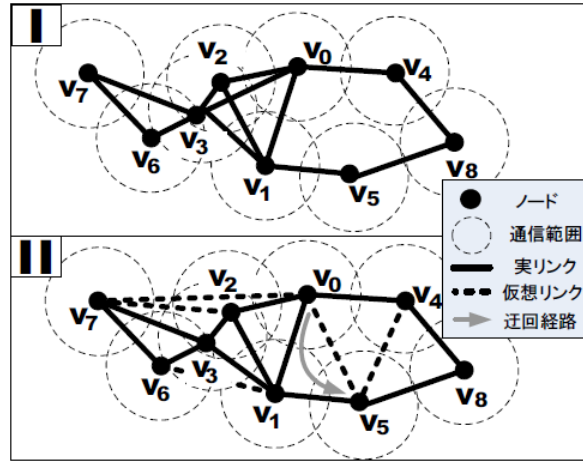


図 4 UDG 上で生成するドロネーネットワーク

ここでは、各基地局ノード（以下ノード）の前提についてまとめる。ノードは以下の前提に基づくものとする。

- (1) 各ノードは、すべて同じ処理能力をもち、マルチホップ無線基地局内の計算主体である。
- (2) 各ノードは、識別可能な固有の ID 及び、GPS などで自ノード位置(X/Y 座標) 通信半径を所持するものとする。
- (3) 各ノードは、通信範囲外のノードに対しての通信は作成中の経路表により構成されるオーバーレイネットワークによるマルチホップ転送を行い、直接もしくはマルチホップ転送によって、他ノードと(2) の情報について交換可能である。

ここでは、このような前提を表すモデル例として Unit Disk Graph(以下  $UDG(V)$ ) を用いる。 $UDG(V)$  とは、全ノードの通信範囲の半径を  $r$  として、任意の 2 ノード  $v_i, v_j$

$\in V$  において、 $|v_i v_j| < r$  を満たす 2 ノードが接続する無向グラフである(ここで、 $V$  はユークリッド平面上のノード集合、 $|v_i v_j|$  は  $v_i, v_j$  間の辺  $v_i v_j$  の距離を表す)。 $UDG(V)$  はこの定義に基づくため、平面グラフではなく、辺が交差する場合が存在する。また本稿で用いるネットワークは、与えられた任意の 2 ノード間の経路の存在を仮定するため、 $UDG(V)$  が連結グラフであるとする。

図 4 の(I) に 9 個のノードから成る  $UDG(V)$  を示す。図上では、通信範囲の直径を前述の  $r$  とするため、ノードを中心とした円が交差する場合にノード間が繋がる。そのため、図上では  $v_0$  と  $v_1$  間は繋がるため、通信経路が存在する。

**Note:** また、ここでは説明の簡単化のため、各ノードは同期的に計算処理を実行するが、非同期の場合にも対応できる。

#### 3.2 準備 (仮想リンク)

図 4 の(II) に  $UDG(V)$  上から、ノード間をドロネー辺で繋がる、提案ネットワークの例を図示する。図上では  $v_0$  は、通信範囲内のノード  $\{v_1, v_2, v_4\}$  とは実線で繋がり、通信範囲外のノード  $\{v_5, v_6, v_7\}$  は破線によって繋がり、これらをノードが所持する隣人表とする。このように、ノードが通信範囲内のノードと繋がる辺を実リンク、直接通信可能でないノード繋がる辺を仮想リンクと定義する。また、これらの辺の両方をまとめてリンクと呼ぶ。

以上の関係をまとめると、任意のノード  $v_i \in V$  が所持するノード集合を  $V_i$  とし、 $v_j \in V_i$  が、 $r < |v_i v_j|$  を満たす場合に  $v_i$  は  $v_j$  と仮想リンクで繋がる。

図 4 の(II) において  $v_0$  の所持する仮想リンクで繋がるノード  $v_5$  と通信を行う場合には、実リンクを辿り、マルチホップ通信を行うことで  $v_5$  と通信を行う。そのため、 $v_0$  はノード  $v_5$  と通信を行うために実リンクの経路が必要である。この実リンクの経路を迂回経路と定義する。

ノード  $v_i \in V$  から仮想リンクで繋がるノード  $v_j$  までの迂回経路を  $Path(v_i, v_j)$  によって表す。また、 $Path(v_i, v_j)$  上の具体的な経路を示す場合には、 $v_i$  から近いほうから順番に  $(v_a, v_b, \dots, v_n)$  と表す。迂回経路は順番も考慮するため、 $2 \leq |Path(v_i, v_j)|$  の場合には、 $Path(v_i, v_j) \neq Path(v_j, v_i)$  となる。実装上は迂回経路は、各ノード上のテーブルで管理される。

**Note:** ただし、仮想リンクに対する迂回経路は複数存在する。そのため、仮想リンク間の経路を常に最小のホップ数で到達できるとは限らない。図 4 の(II) では、 $v_0$  の  $v_5$  までの迂回経路  $Path(v_0, v_5)$  は、 $(v_1)$  以外にも  $(v_4, v_8)$  などが考えられる。

#### 3.3 全体的な生成手順

全体的なドロネー図の自律分散生成法について述べる。 $UDG(V)$  上のノード  $v_i \in$



$V$  が実行するドローン図生成アルゴリズムは所持する隣人数 $|V_i|$  によって異なり、 $(|V_i| = 1)$  の場合には、 $v_i$  は隣人表に 2 つ以上のノードを所持するまで他者からの通知を待つ。

$(|V_i| = 2)$  の場合には、 $v_i$  は三角化通知のみを行う。

$(|V_i| \geq 3)$  の場合には、 $v_i$  は局所ドローネ化、委譲、三角化通知の 3 つの処理を順番に行う。

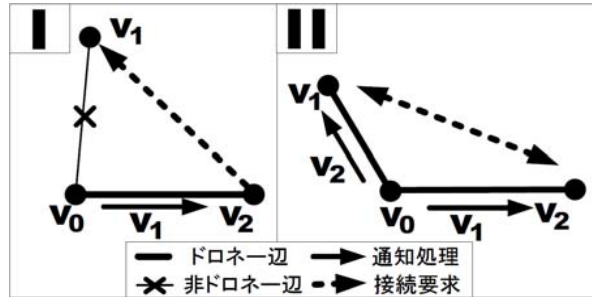


図 5 ドローン図生成時の通知処理と接続要求

なお、初期状態において  $v_i$  は、 $|v_i v_j| < r$  となる全ての  $v_j$  を  $V_i$  のノードとし、他ノードとの接続要求や通知を行うことによって  $V_i$  を更新する。これを全てのノードが繰り返し実行することで、全体としてドローン図を構築する。ここで、上記で示した主要 3 つのアルゴリズムの内容について確認する。

局所ドローネ化とは、 $v_i$  を中心として時計順に並べ全ての隣人ノードを、外接円判定によって  $v_i$  とドローネリンクで繋がるノード（ドローネノード）か、繋がらないノード（非ドローネノード）かの 2 種類に分類する処理である。

委譲とは、 $v_i$  の局所ドローネ化によって非ドローネノードをドローネノード内でその非ドローネノードに最も近いノードへ委譲する通知処理である。また、三角化通知とは、ドローネノード間が互いノード間の関係を認識させるために行う通知処理である。図 5(I) に委譲処理を示す。 $v_0$  はドローネノードである  $v_2$  に非ドローネノードである  $v_1$  を通知する。ここで、 $v_0$  から通知を受け取った  $v_2$  は  $v_1$  に対して通信経路の接続要求を行うが  $v_2$  の通信範囲内にないため、直接接続できない。同様に、(II) の三角化通知においても、 $v_0$  は  $v_1, v_2$  の 2 つのドローネノード間が相互に接続するため、両方に対して通知を行うが、通知を受けたノードが自身の通信範囲内に存在しない場合には接続できない。

そこで、ノード  $v_i \in V$  がこれらの通知処理を行う場合には、迂回経路を付加して通知することでこれに対応する。これらの通知処理は、 $v_i$  が所持するノードから対象となる 2 つを取り出し、そのノード間の関係に基づき通知処理を行う。そのため、次

節では、2 つのノード間に対する迂回経路の生成法について説明する。また、この迂回経路を実装した場合、非常に長い迂回経路を如何に管理するかは問題である。そこで、迂回経路をテーブルによって分散的に管理する手法の基本的な考え方についても述べる。

**Note:** なお、ここではノードの隣人による迂回経路の生成は先に来た通知を優先する。つまり、 $v_i$  が  $v_j$  からの通知処理によって  $Path(v_i, v_k)$  の通知を受けた場合に、 $v_j$  以外の他のノード  $v_l$  から  $v_k$  及び  $Path(v_l, v_k)$  の通知を受けると、 $v_l$  からの通知は破棄する。

### 3.4 迂回経路生成

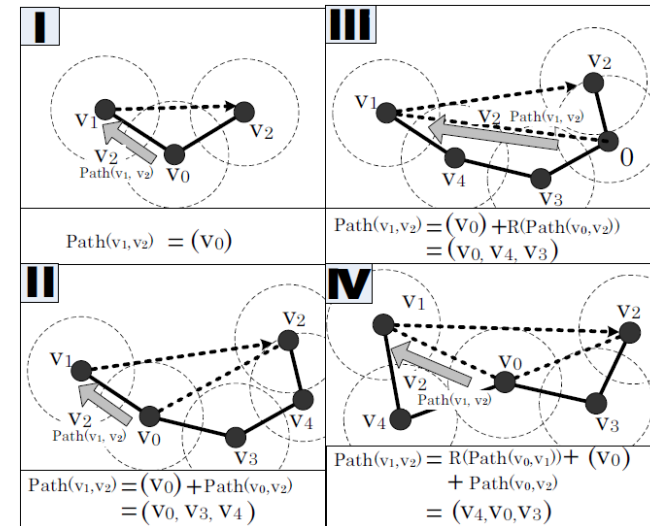


図 6 迂回経路の生成例

本節では、各ノードによる迂回経路の生成アルゴリズムについて Algorithm 1 及び図 6 を用いて説明する。ここでは、ノード  $v_i \in V$  が隣人表から 2 つのノード  $v_j, v_k \in V_i$  を取り出した場合において、 $v_i$  が  $v_j$  に対して  $v_k$  の情報を通知する時における迂回経路  $Path(v_j, v_k)$  の生成について示す。また、 $v_j, v_k$  間の迂回経路を  $Path(v_j, v_k) = null$  としてアルゴリズムを始める。これらは Algorithm 1 の 2 行目に初期状態として与えている。

**Step1.**  $v_i$  は  $r < |v_j v_k|$  であるかをチェックする. これは Algorithm 1 の 2 行目に当たる.  
**Step2-1.**  $v_i$  は  $v_k$  及び  $v_j$  の両方と実リンクで接続される場合,  $v_i$  自身のみが  $v_j$  と  $v_k$  間の迂回経路となるため,  $Path(v_j, v_k)$  に  $v_i$  自身のみを追加する. これは Algorithm 1 の 3,4 行目に当たる. また, 図 3(I) のように  $v_0$  の所持するノード  $v_1, v_2$  が実リンクで繋がっているとこの処理を実行する.

Algorithm 1 ノード $v_i$ による迂回経路生成
1: Given $v_j, v_k \in V_i, Path(v_i, v_j) \leftarrow \text{null}$
2: <b>if</b> $r < d(v_j, v_k)$ <b>then</b>
3: <b>if</b> $v_i$ has no $Path$ <b>then</b>
4: $Path(v_j, v_k) \leftarrow v_i$
5: <b>else if</b> $v_i$ has $Path(v_i, v_k)$ <b>then</b>
6: $Path(v_j, v_k) \leftarrow (v_i + Path(v_i, v_k))$
7: <b>else if</b> $v_i$ has $Path(v_i, v_j)$ <b>then</b>
8: $Path(v_j, v_k) \leftarrow (Reverse(Path(v_i, v_j)) + v_i)$
9: <b>else</b>
10: $Path(v_j, v_k) \leftarrow (Reverse(Path(v_i, v_k)) + v_i + Path(v_i, v_j))$
11: <b>end if</b>
12: <b>end if</b>

**Step2-2.**  $v_i$  の所持する  $v_k$  が他ノードから通知を受けたため, 仮想リンクで繋がる場合には, 迂回経路  $Path(v_i, v_k)$  を所持している. そのため,  $v_i$  はまず  $Path(v_j, v_k)$  に  $v_i$  を追加した後,  $Path(v_i, v_k)$  を順番に追加する. これは Algorithm1 において 5,6 行目に当たる. なお, アルゴリズム中 7 行目の”+”は  $Path$  の連結を示す. つまり,  $Path(v_i, v_k) = (v_a, v_b)$  の場合,  $Path(v_j, v_k)$  は  $(v_i, v_a, v_b)$  となる. 図 6(II) において,  $v_0$  は,  $Path(v_1, v_2)$  に自身を格納した後,  $Path(v_0, v_2)$  を格納する.

**Step2-3.**  $v_i$  の所持する  $v_j$  が仮想リンクで繋がる場合には迂回経路  $Path(v_i, v_j)$  を所持しているため,  $v_i$  は  $Path(v_j, v_k)$  に反転させた  $Path(v_i, v_j)$  を追加した後,  $v_i$  を追加する. Algorithm1 中 7,8 行目に当たり,  $Reverse(Path(v_i, v_j))$  とは, 経路の反転を表す. つまり,  $Path(v_i, v_j) = (v_a, v_b)$  の場合には  $Reverse(Path(v_i, v_j)) = (v_b, v_a)$  となり, 結果,  $Path(v_j, v_k) = (v_a, v_b, v_i)$  となる.  $v_j$  から  $v_i$  への経路は,  $v_i$  から  $v_j$  への経路の反転であるため, このような処理を行うことで,  $v_j$  から  $v_k$  までの辿ることが可能な経路を構成できる. これは, 図 6 中(III) に当たる.

**Step2-4.**  $v_i$  の所持する  $v_j$  及び  $v_k$  の両方が仮想リンクで繋がる場合には迂回経路  $Path(v_i, v_j)$  及び  $Path(v_i, v_k)$  を所持するため,  $v_i$  は  $Path(v_j, v_k)$  に反転させた  $Path(v_i, v_j)$  を追加した後,  $v_i$  を追加し, さらに  $Path(v_i, v_k)$  を追加する. これは, Algorithm1 中 9,10

行目, 図 6 における(IV) にあたる.

### 3.5 テーブルによる迂回経路の管理とルーティング

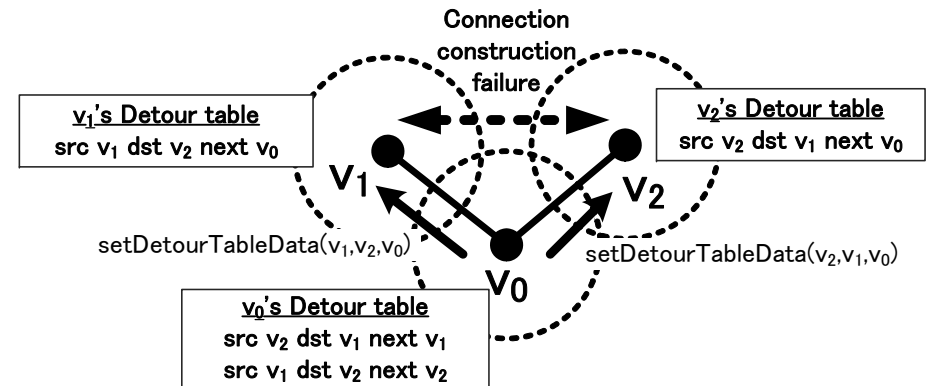


図 7 迂回経路の経路テーブルによる管理例

これまでに示した迂回経路データの保持方法では, 転送データに迂回経路において経由する全てのノードの長さに比例した迂回経路データが格納されることになる. そこで, ここでは迂回経路について各ノードのテーブルに分散して格納する手法について述べる. 図 7 に, 迂回経路の経路テーブルによる管理例を示した. 先程までの方法では, 各ノードは三角化通知を行う際に通知内に迂回経路生成の為の情報として自身のノード ID を付加して送信を行った. 図 7 のようなノードの関係でいえば,  $v_0$  は  $v_1, v_2$  双方に三角化通知を行う際,  $v_0$  の情報を入れて送信する. そして,  $v_1$  では,  $Path(v_1, v_2) = (v_0)$  という迂回経路を,  $v_2$  では,  $Path(v_2, v_1) = (v_0)$  という迂回経路を生成する. この時, 例えば,  $v_1$  から  $v_2$  への転送パケットを生成する場合, 送信元として  $v_1$ , 送信先として  $v_2$ , 経由ノードとして  $v_0$  を格納しなくてはならない. 迂回経路が中間 1 HOP 程度であればさして問題にはならないが, もし迂回ルートがある程度の長さを持つ場合には, その経由ノードの列が全てパケットの中に挿入されることとなる. そこで例えば  $v_1$  から  $v_2$  にノードを送信する場合にパケットの中には  $v_2$  のみしか入れないようにしたい. そこで迂回経路管理用テーブルを各ノードに持たせる. このテーブルの各行は三つの項目からなる. src は, 送信ノードを指す, dst は, 受信ノードを指す. パケットを受け取った場合この二つノードによってテーブルの一致検索を行い, 一致する行があった場合, next に書かれたノードに対して転送する. このテーブルの生成方法は例えば図 7 における  $v_0, v_1, v_2$  では以下のようなになる.

1.  $v_0$  は  $v_1$  と  $v_2$  に対して三角化通知に相当する `setDetourTableData` を送信する前に、自身の `detour table` に、 $v_1$  と  $v_2$  に通知するノードを `dst` と `next` とし、三角化通知相手の各ノードを `src` とする行を追加する。
2.  $v_0$  は、三角化通知の代わりに、`setDetourTableData` を送信する。`setDetourData` は、(送信ノード,受信ノード,経由ノード) の三つで構成されており、送信ノードと受信ノードの部分には、送り先のノードを `src` に入れ、通知内容となるノードを `dst` に入れ、`next` の部分に自身のノード ID を入れる。そして、このメッセージを受け取った各ノードは内容に沿った行をテーブルに付け加える。
3. 以上の手順で作成したテーブルは図 7 の各ノードのテーブル内容となっている。このノードを利用した場合、各ノードからは送信ノードと受信ノードを送信パケットに含めて送信することで受信ノードに届く。

例えば、 $v_1$  から  $v_2$  へ送信を行う場合、 $v_1$  と  $v_2$  の情報だけをパケットに入れる。 $v_1$  は物理的には  $v_2$  へのリンクを持っていないが、`Detour table` を参照することで迂回ルートを発見し、 $v_0$  を経由する転送を行う。この時、`arc` と `src` さえあれば転送を行うことができ、経由ノードを含めるパケットに入れる必要がなくなる。迂回経路が入り子になり、経由ノード数が増加した際にも同様に、`src` と `dst` だけで転送が可能になる。

#### 4. シミュレーションによる評価

提案手法による経路表生成手法の動作確認の為、Java によってシミュレータを試作した。有理数クラスを自作し、計算誤差が発生しないシミュレータ実装を行った。[1.0\*1.0]の正方形の空間に 256,512,1024,2048 個のノードをばらまき、それぞれについて、 $r=0.10$  と  $r=0.15$  を指定してのユニットディスクグラフを物理リンクのトポロジとして生成し、提案アルゴリズムによる迂回経路生成のシミュレーションを行った。迂回経路生成完了後、評価項目として、迂回経路長の平均、迂回経路長の分散、迂回経路を含むノードのリンク数の合計、迂回経路を含むノードのリンク数の分散を取得した。図 8 に迂回経路長の平均を示す。ノード数が 256 の時のみ迂回経路の長さが  $r=0.15$  の時 2.6 程度、 $r=0.10$  の時に 5 程度と大きな差になっているが、ノード数 512 の時、1024 の時、2048 の時は、 $r$  の値に関わらず 3 程度となっており差が小さく、ノード数毎の差もほとんど無い。これは、ノード数が大きくなったことによってノード数の密度が上がり物理リンクが十分に生成された為に長い迂回経路が生成されなかった為だと考えられる。逆にノード数 256 で  $r=0.10$  であった時は、ノード間にある程度の距離があり、迂回ルートが長くなったものと思われる。この迂回ルートの長さは、ドロネ

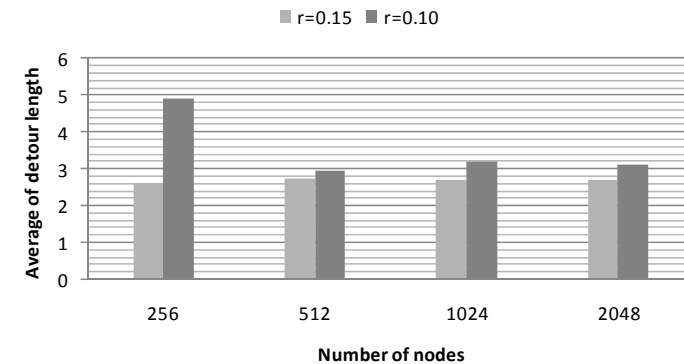


図 8 迂回経路の経路長平均

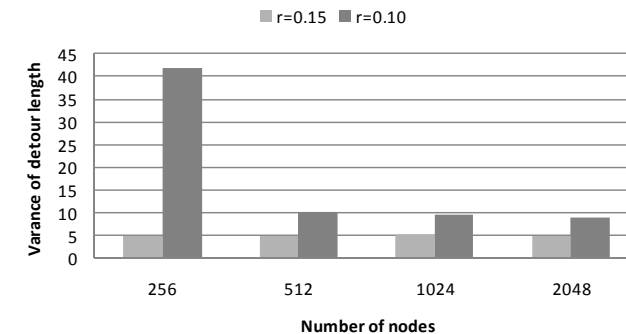


図 9 迂回経路の経路長分散

ーオーバーレイネットワークを作成する為に必要な制御信号の送信 HOP 数と等しくメッシュ網路上での制御信号の送信範囲の局所性を測る上で重要な値である。次に、図 9 に迂回経路長の分散を示す。ノード数が 256 の時の  $r=0.10$  の時に分散が非常に大きくなっている。また、その他の場合には軒並み迂回経路長の分散は小さい。これはノード数 256 の時のネットワークの物理リンクが少なく迂回経路長に様々なものが現れる要因になっているものと考えられる。逆にノード数が一定の密度を超えた場合は、総じて迂回経路長の分散は似たようなものになることが分かり、平均の場合の同様の傾

向と合わせて基地局ノードの配置がある一定密度を超えれば、制御信号の伝搬範囲は十分に局所性を保つということが言え、広大な無線メッシュ上でも用いやすいと捉えることができる。

## 5. 関連研究

無線環境を考慮したネットワーク上での経路選択は、スケーラビリティを保持するため、フラッディングの利用を避ける必要がある。そこで、ID やノード位置に基づくネットワークの構造化や対象範囲の制限をした通信によって、ネットワーク全体の通信量を低減しつつ、任意のノード間に対して到達可能な経路選択法が研究されている。

F. Araujo ら[5]は、構成法には深く触れていないものの平面上に配置された通信ノードをドロネー図状に結んでネットワーク化しルーティングする手法についてのアイデアについて示している。Karp ら[6] は、ノードの位置関係によってガブリエルグラフを構築し、欲張り法及び **Right-Hand-Rule** を利用することで、フラッディングを利用せずに、平面グラフ上で任意のノード間での経路選択を保障している。一方、提案手法では、構成したドロネーリンクを辿ることによって任意の2 ノード間で経路選択を行うことができる。Li 及び Wang ら[7],[8] は、通信範囲内のノードに焦点を当て、UDG 上から平面局所化ドロネー図と呼ばれる平面グラフを提案し、同構造を利用した欲張り法による経路選択について議論している。これに対して提案手法では、通信範囲外のノードには迂回経路によって到達可能なリンクを構成できるため、グラフが連結であれば、通信範囲に依存せずネットワークを構築することができる。Caesar ら[9] は、Chord 環を利用しており、各ノードがノード ID の近いノードを隣人表として所持し、ID 空間上で近いノード ID を順番に辿ることで ID のみによる経路選択法を実現している。また、ID 空間上の隣人がノードの通信範囲外にある場合には、マルチホップ通信による隣人までの経路を所持することで対応している。しかしながら、ノード数の増加に伴い ID 空間も大きくなるため、ノード数が大きい場合には隣人までのホップ数が増加する。一方、提案手法では、ノード数の増加に対して所持する隣人数は一定であり、拡張性が高い。

## 6. まとめ

本研究では、最初に大規模災害時に対応可能なアクセスネットワークの要件について阪神淡路大震災時の土木学会の調査報告を元に論じ、そこから我々が考える大規模災害時に耐えるアクセス網の全体像について示した。その中で特に大規模なアクセス

メッシュ網が、その中核を成す可能性について提示し、それを実現する為のネットワークングプロトコルとルーティングプロトコルについて論じ、ドロネーオーバーレイネットワークを無線メッシュ網上に構築し用いることを提案した。更にドロネーオーバーレイを無線メッシュ網上に構築するにあたって迂回経路を用いることが重要であると考へ、その生成手法について論じた。また、実際にオーバーレイネットワーク構成のためのシミュレーションを製作して、迂回経路長の平均と分散についてのデータを取得した。その結果、ある一定の密度を超えた基地局配置において、迂回経路長は短く、一定しており、制御信号の送信範囲が局所的で広大な無線メッシュ網のような分散性とスケーラビリティが必要なネットワークに適したネットワークングプロトコルとして用いることが可能であると思われる。

## 参考文献

- 1) 阪神・淡路大震災調査報告編集委員会『阪神・淡路大震災調査報告 ライフライン施設の被害と復旧』(社)土木学会(1997/9)
- 2) M. Inoue, et al., "A Novel Managed Wireless Mesh Architecture for Community Service Platform," IEEE WCNC, Apr. 2009.
- 3) 大西ほか, "分散処理・制御アクセスネットワーク NerveNet によるインタラクティブ広告配信," 信学技報, MoMuC2010-1, 2010年1月
- 4) 大西真晶, 源元佑太, 江口隆之, 加藤宏章, 西出亮, 上島紳一: "ノード位置を用いた P2P モデルのためのドロネー図の自律分散生成アルゴリズム", 情報処理学会論文誌: データベース, 47, SIG4(TOD29), pp. 51 - 64 (2006).
- 5) F. Araujo and L. Rodrigues: "Geopeer: A location-aware peer-to-peer system", Proceedings of the 3rd IEEE International Symposium on Network Computing and Applications (IEEE NCA04), Cambridge, MA, USA, pp. 39-46 (2004).
- 6) B. Karp and H. T. Kung: "GPSR: greedy perimeter stateless routing for wireless networks", MobiCom '00: Proceedings of the 6th annual international conference on Mobile computing and networking, pp. 243-254 (2000).
- 7) X.-Y. Li, G. Calinescu, P.-J. Wan and Y. Wang: "Localized delaunay triangulation with application in ad hoc wireless networks", IEEE Transactions on Parallel and Distributed Systems, 14, 10, pp. 1035-1047 (2003).
- 8) Y. Wang and X.-Y. Li: "Efficient delaunay-based localized routing for wireless sensor networks: Research articles", International Journal of Communication Systems, 20, 7, pp.767-789 (2007).
- 9) M. Caesar, M. Castro, E. B. Nightingale, G. O' Shea and A. I. T. Rowstron: "Virtual ring routing: network routing inspired by dhts", SIGCOMM, pp. 351-362 (2006).