

# アドホックネットワークにおける凸包端末の検索手法

駒井友香<sup>1,a)</sup> 原 隆浩<sup>1,b)</sup> 西尾章治郎<sup>1,c)</sup>

**概要:** アドホックネットワークにおける位置依存サービスの一つとして、ネットワークの凸包の頂点となる端末の情報を取得する凸包検索がある。アドホックネットワークにおいて凸包検索を実現する方法として、全ての端末の位置情報を取得することが考えられるが、これは、大きなトラヒックが発生する原因となる。また、アドホックネットワークでは端末が移動するため、凸包の形状が時刻により変化する。そのため、リアルタイムにネットワークの凸包を検出する手法が必要となる。本稿では、アドホックネットワークにおいて、トラヒックの削減、検索時間の削減、および検索結果の取得精度の維持を目的とする凸包検索手法を提案する。そのための方法として、各端末が返信時にローカルな凸包を作成することにより、凸包の頂点となる可能性のある端末のみの情報を返信する LCH (Local Convex Hull) 法、および、ネットワークの外周端末へクエリを順に伝搬する LW (Local Wrapping) 法の二手法を提案する。シミュレーション実験により、提案手法では、全ての端末の位置情報を取得する単純な手法と比べ、小さなトラヒック、かつ短い検索時間により、高い取得精度を維持した凸包の検出を行うことができることを確認した。

## 1. はじめに

近年、無線通信技術の発展と、計算機の小型化や高性能化に伴い、ルータ機能をもつ端末のみで一時的な無線ネットワークを形成するアドホックネットワークへの関心が高まっている。アドホックネットワークでは、端末が自身の無線通信範囲内に存在する端末と通信する場合、基地局を介さずに直接通信できる。一方、自身の無線通信範囲内に存在しない端末と通信する場合、これらの端末の間に存在する端末がパケットを中継して、マルチホップ通信を行う。アドホックネットワークは既存の通信基盤を必要とせず、端末のみで自律分散的にネットワークを構築できるため、災害時の救助活動やイベント会場での情報共有、およびセンサネットワークでの情報収集などへの応用が期待されている。

広い領域において多数の端末がアドホックネットワークを構成している状況では、ネットワークの地理的な形状を把握する等、位置に依存した情報を検索するような位置依存サービスが求められている。そのための有効な方法として、ネットワークの凸包の頂点となる端末の情報を取得する凸包検索がある。凸包検索により、ネットワークの凸包を検出することができるため、端末の存在する領域の把握、

最遠点对となる端末の発見など様々なアプリケーションへの応用が考えられる。

ここで、与えられた点集合の凸包を検出するためのアルゴリズムは、これまでに多く提案されている。これらのアルゴリズムは、集合内の全ての点（ネットワーク内の端末の位置情報）を入力として、凸包の頂点を高速に算出することができる。そのため、これらのアルゴリズムを用いてネットワークの凸包を検出するためには、まずネットワーク内の端末の位置情報を取得する必要がある。これを実現する最も単純な方法として、端末が検索クエリをネットワーク全体にフラッディングし、これを受信した端末が、自身の位置情報を返信することが考えられる。しかし、クエリを受信した全端末が位置情報を返信すると、凸包の頂点ではない端末の無駄な返信が多くなり、不要なトラヒックが発生してしまう。この場合、アドホックネットワークでは無線通信帯域が限られているため、パケット衝突による転送遅延の増大や検索結果の精度の低下が予想され、効率的に凸包を検出することができないと考えられる。特に、アルゴリズムによる計算のコストと比べ、位置情報の収集にかかる通信コストが大きいため、これを削減することが重要となる。そのため、クエリによる検索時に、必要な端末の情報のみを取得するための仕組みが必要である。

また、アドホックネットワークでは、端末が移動するため、凸包の形状や凸包の頂点となる端末の集合が検索を行う時刻により変化する。そのため、リアルタイムに凸包を検出することが必要である。

<sup>1</sup> 大阪大学大学院情報科学研究科マルチメディア工学専攻  
Yamadaoka 1-5, Suita, Osaka 565-0871, Japan

a) komai.yuka@ist.osaka-u.ac.jp

b) hara@ist.osaka-u.ac.jp

c) nishio@ist.osaka-u.ac.jp

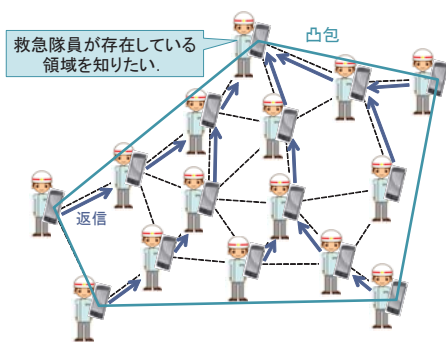


図 1 凸包検索例

図 1 は、災害地域において、救急隊員がアドホックネットワークを構築し、ある救急隊員が凸包検索を行う例を示している。このとき、全ての端末の位置情報を取得すると、無駄に大きなトラフィックが発生する。また、トラフィック増大によってパケット衝突の機会が増加し、転送遅延が増大するため、リアルタイムな凸包検出を実現することができず、検索結果の精度の低下が予想される。

そこで本稿では、アドホックネットワークにおいて、トラフィックの削減、検索時間の低減、および検索結果の取得精度の維持を目的とする凸包検索手法を提案する。提案手法では、検索クエリの受信により取得した情報より、各端末が自律的に返信する情報を判断する。そのための方法として、LCH (Local Convex Hull) 法と LW (Local Wrapping) 法の二つの手法を提案する。LCH 法では、ネットワーク全体にクエリを送信し、返信時に、中継端末において受信した端末の情報を基に、ローカルな凸包検出を行い、ローカルな凸包の頂点となる端末の情報のみを返信する。一方、LW 法では、領域全体を  $xy$  平面とみなし、まず  $y$  座標が小さい端末へクエリを転送し、その後、ネットワークを囲うように、ネットワークの外周に存在する端末へ順にクエリを伝搬することで、凸包の頂点となる端末の情報を取得する。提案手法では、ネットワーク内の全ての端末の位置情報を取得することなく、凸包の頂点となる端末の情報を取得することができる。

以下では、2. で関連研究について述べる。その後、3. で提案手法について述べ、4. でシミュレーション実験の結果を示す。5. で議論を行い、最後に、6. で本稿のまとめと今後の課題について述べる。

## 2. 関連研究

本章では、数学的な凸包検出アルゴリズムに関する従来研究、および様々なネットワークにおける境界検出に関する従来研究について紹介する。さらに、本研究と従来研究を比較し、その相違点について述べる。

### 2.1 凸包検出アルゴリズム

これまでに多くの凸包検出アルゴリズムが提案されてい

る [1]。文献 [5] では、パッケージ包装の技術を基にした方法 (Javis の行進法) が提案されている。この方法では、点の集合のうち  $y$  座標が最小となる点から検索を開始し、この点と各点の偏角をそれぞれ算出する。偏角が最も小さい点を凸包の頂点として発見し、この点を原点として再度各点との偏角を算出することを繰り返し、凸包を検出する方法である。LW 法は、この方法に発想を得ている。文献 [4] では、Graham 走査法が提案されている。この方法は、点の集合のうち  $y$  座標が最小となる点と各点の偏角を算出し、偏角が小さい順にそれぞれの点が凸包の頂点となるか判断するものである。これらのアルゴリズムは、検索対象となる点の集合を入力として凸包を少ない計算量で検出することを目的としており、アドホックネットワークにおいて凸包検出のためのトラフィックの削減に着目した本手法とは目的が異なる。

### 2.2 様々なネットワークにおける境界検出

筆者らの知る限り、アドホックネットワークにおける凸包の検出に関する研究は、これまでに行われていない。そこで、ネットワークの凸包検出に類似する、ネットワークの境界検出に関する従来研究について述べる。文献 [3] では、アドホックネットワークにおいて端末の隣接関係を基にした境界の検出手法を提案している。この手法では、隣接関係を基に作成したグラフ中に存在する三角形を塗りつぶした領域を囲む線を境界として検出することができる。そのために、ネットワークを骨格となるいくつかの開領域に分割し、この骨格を基に境界の検出を行う。文献 [9] では、3D センサネットワークにおいて端末の隣接関係を基にした境界の検出手法を提案している。この手法では、ポロノイ領域を作成することでネットワークの境界付近の端末の候補を選出し、よりネットワークの外側に存在する端末を最終的な境界端末として算出する。文献 [7] では、密なセンサネットワークにおいて観測値の地理的な境界線を検出する方法を提案している。この手法では、各端末が隣接端末のセンサの観測値を受信し、自身が境界付近に存在すると判断した場合、センサの観測値を送信する。これにより、境界線を作成するために必要な境界付近の端末のみからセンサの観測値を収集することができる。これらの手法は、各文献において定義している境界の検出を行うものであり、それぞれの定義に特化したアプローチを採用している。一方、本研究では、凸包の検出を目的としており、これらの研究とは定義とアプローチが共に異なる。

### 3. 凸包端末の検索手法

本章では、本稿で提案する凸包検索手法を説明する。以下では、まず想定環境について述べ、その後、凸包端末へのクエリの転送、返信方法の詳細として、LCH 法と LW 法における検索手順について説明する。

### 3.1 想定環境

本稿では、アドホックネットワークを構成する各端末が、ネットワークの地理的な凸包を検索する環境を想定する。凸包検索を行う端末は、検索クエリを発行し、ネットワーク内の端末の中から、凸包の頂点となる端末の情報を取得する。ネットワーク内には、同等の性能（通信伝搬距離  $rm$  程度）を持つ  $n$  個の端末（識別子： $M_1, M_2, \dots, M_n$ ）が存在し、各々が自由に移動する。ネットワークに参加する端末は既知であり、ネットワークを構築する前に各端末に一意的識別子が付与されているものとする。各端末は、GPS などにより自身の位置情報  $(x, y)$  を把握しているものとする。なお、経度を  $x$  座標、緯度を  $y$  座標と対応するものとする。

### 3.2 Local Convex Hull(LCH) 法

ネットワーク全体にクエリを送信し、各端末が位置情報を返信する際、中継端末では返信された端末の情報を把握することができる。そのため、中継端末において、既に受信した端末の集合を入力とするローカルな凸包検出を行うと、凸包の頂点となる可能性のない端末を各中継端末において判断することができる。LCH 法では、中継端末が凸包の頂点とならない端末の情報を返信しないことにより、無駄なトラフィックを削減する。

以下では、検索クエリを発行した端末  $M_s$  と検索クエリメッセージを受信した端末の動作について説明する。

- (1) クエリ発行端末  $M_s$  は隣接端末へ検索クエリを送信する。このメッセージには、メッセージ発行端末  $M_s$  の識別子および位置情報が含まれる。
- (2) 検索クエリを受信した端末  $M_p$  は、送信元端末を親端末として記録する。検索クエリを初めて受信した場合、応答返信待ち時間 (Wait Time : WT) を次式のように設ける。

$$WT = \alpha \cdot \left(1 - \frac{a}{d}\right). \quad (1)$$

ここで、 $\alpha$  はシステム管理者が設定するパラメータ、 $a$  は  $M_s$  と  $M_p$  の距離、 $d$  は事前に設定される正の定数 ( $a < d$  を満たす。) である。この式を用いることで  $WT$  の値は、端末  $M_s$  から遠い端末ほど小さくなる。上記の処理に加えて、全ての隣接端末にクエリメッセージを送信する。一方、受信したクエリメッセージが、既に受信したものである場合、このメッセージを破棄する。

- (3)  $WT$  の値が小さい ( $M_s$  からの距離が遠い) 端末から、クエリ返信を開始する。具体的には、 $WT$  が経過した  $M_q$  は、端末の識別子と位置情報を返信データとして、クエリ応答メッセージに添付し、自身の親端末に送信する。
- (4) クエリ応答メッセージを受信した端末は、受信した

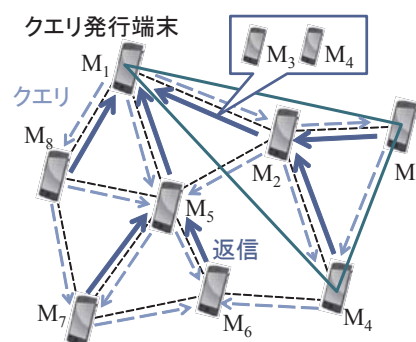


図 2 LCH 法による凸包検索

メッセージ中の返信データと、それまでに他の端末から受信したクエリ返信メッセージ中の返信データ、および、自身の情報を統合して、自身の返信データを作成する。ここで、作成した返信データに情報が含まれる端末により凸包検出アルゴリズムを実行し、凸包の頂点となる端末の情報のみを自身の返信データとする。自身がクエリ発行端末でない場合、 $WT$  の経過後、作成した返信データを添付したクエリ返信メッセージを、親端末へ返信し、手順 (3) へ戻る。

図 2 を用いて、端末  $M_1$  がクエリを発行した場合の LCH 法による凸包検索の例を説明する。まず、クエリ発行端末  $M_1$  はクエリを隣接端末へ送信し、これを受信した  $M_2$ ,  $M_5$ , および  $M_8$  は  $M_1$  を親端末として記録した後、クエリを隣接端末へ送信する。 $M_1$  から最も遠い  $M_4$  から  $WT$  の経過後、自身の位置情報を親端末  $M_2$  へ送信する。 $M_3$  および  $M_4$  から返信を受信した端末  $M_2$  は  $WT$  の経過後、 $M_1 \sim M_4$  の位置情報を基にローカルな凸包を生成し、クエリ発行端末以外でその凸包の頂点となる  $M_3$  と  $M_4$  の情報を親端末へ送信する。これをネットワーク全体で行うことにより、 $M_1$  はネットワークの凸包の頂点となる端末の情報を取得することができる。

LCH 法では、返信を中継する端末において、すでに把握している端末の情報より、凸包の頂点となる可能性のない端末の情報は返信しないことで、無駄なトラフィックの削減を行うことができる。

### 3.3 Local Wrapping(LW) 法

LCH 法のように、クエリをネットワーク全体へ送信する方法では、全ての端末がクエリの転送を行う必要がある。しかし、この方法では凸包の頂点とならない端末へのクエリ転送が多くなり、トラフィックが増加してしまう。ここで、性質上、凸包の頂点となる端末は、ネットワークの淵 (外周) に存在する。そのため、行進法 [5] を基にし、ネットワークの外周端末を順に巡回してクエリを伝搬することにより、小さなトラフィックにより検索を行う。

提案する LW 法では、まず、 $y$  座標が小さい端末へクエリを送信し、この端末からネットワークの外周端末へ順に

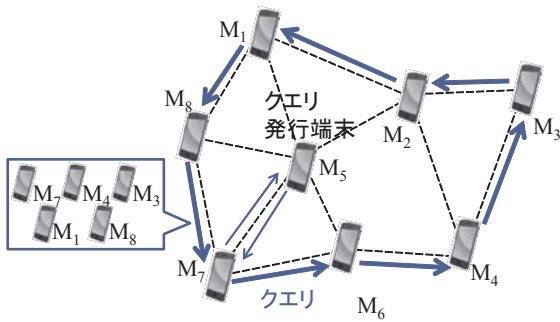


図 3 LW 法による凸包検索

クエリを伝搬する. 図 3 は, 端末  $M_5$  がクエリを発行した場合の LW 法による凸包検索を行う例を示している. 図に示すように,  $y$  座標が小さい端末  $M_7$  へクエリを転送した後, 反時計まわりに外周端末へクエリを伝搬する. 外周端末へ効率的にクエリを伝搬するために, LW 法では, 転送元の端末とのなす角が小さいほど返信の待ち時間を小さく設定し, なす角が小さく, より外側に存在する端末を次にクエリを転送する端末として選択する.

以下では,  $y$  座標が小さい端末へのクエリ送信に, 用いるジオルーティング [6] について説明したのち, 凸包の頂点端末へのクエリ転送および返信の詳細について述べる.

### 3.3.1 ジョルーティング

本手法では, 3 ウェイハンドシェイクを用いたジオルーティングにより, 指定した位置により近い端末へのメッセージの転送を行う.

具体的には, まず, 送信元端末が隣接端末検索メッセージをブロードキャストし, これを受信した端末が送信元端末より指定位置に近い場合, 返信待ち時間を設定する. この返信待ち時間は, 指定位置に近ければ近いほど短く設定されるため, 指定位置に最も近い端末が最も早く送信元端末へ返信を行う. 返信を受け取った送信元端末は, 返信元端末へメッセージを送信する. メッセージを受信した端末は隣接端末検索メッセージ送信元端末として同様の手順を繰り返し, 指定位置に最も近い端末までメッセージを伝搬する.

このジオルーティング手法により, 1 本の経路で転送が行えるため, 少ないトラフィックで指定した位置に最も近い端末までメッセージを伝搬することが可能である.

### 3.3.2 凸包の頂点端末へのクエリ転送, 返信

以下では, 検索クエリを発行した端末  $M_s$  と検索クエリメッセージを受信した端末の動作について説明する.

- (1) クエリ発行端末  $M_s$  は 3.3.1 項で示したジオルーティングを用いて  $y$  座標が小さい端末までクエリを転送する. この時の指定位置は, メッセージを送信する端末の位置  $(x, y)$  とした場合,  $(x, y - r)$  とする. この手順で, 返信メッセージを一つも受信しない場合, 自身が  $y$  座標が小さい端末であると判断し, 凸包端末の情

### Algorithm 1 返信データの選択

```

1: Data:返信データ
2: i = # of data in Data
3: j = i-1
4: while  $\angle \text{Data}[i] \text{Data}[j] \text{Data}[j-1] < \pi \theta$  do
5:   remove Data[j] from Data
6:   j = j-1
7: end while
    
```

報取得を開始する.

- (2) クエリ (転送) メッセージを受信した端末  $M_t$  (始点端末) は, 自身の識別子と位置情報を返信データとして追加する. この返信データに含まれる端末数が 3 以上の場合, アルゴリズム 1 を用いて返信データを選択する. このアルゴリズムでは, クエリ転送を行った端末のうち凸包の頂点ではない端末を取り除くことができる.
- (3) 端末  $M_t$  は, 全ての隣接端末に次端末検索メッセージを送信する. このメッセージには, 返信データの最後尾から 2 つの端末 ( $M_t$  と  $M_q$ ) の識別子と位置情報が含まれる. なお,  $M_t$  が検索開始端末である場合は,  $M_t$  の識別子と位置情報のみを送信する.
- (4) 次端末検索メッセージを受信した端末  $M_p$  は, 次端末返信メッセージ送信のための待ち時間  $WT$  を次式のように設ける.

$$WT = \text{Max\_delay} \cdot \left( \frac{\theta}{2\pi} \right). \quad (2)$$

ここで,  $\text{Max\_delay}$  はシステム管理者が設定するパラメータ,  $\theta$  は  $\angle M_q M_t M_p$  のなす角 (半直線  $M_t M_q$  を始線として反時計回り (図 4) である. なお,  $M_q$  の位置情報が含まれていない場合には, 原点を  $M_t$  とした  $M_p$  までの偏角を算出する. また,  $M_q = M_p$  の場合,  $\theta = 2\pi$  とする. この式を用いることで  $WT$  の値は, 端末  $M_p$  とのなす角が小さい端末ほど小さくなる.

- (5)  $WT$  が経過した端末  $M_p$  は,  $M_t$  へ次端末返信メッセージを送信する. このメッセージには,  $M_p$  の識別子と位置情報が含まれる.
- (6) 次端末返信メッセージを受信した  $M_t$  は, 既にクエリ転送メッセージを送信したことがあり, 前回のクエリ転送先が  $M_p$  である場合, 手順 (8) へ進み, 検索クエリ発行端末へのクエリの応答を開始する. この条件に当てはまらない場合,  $M_t$  は, 送信元端末  $M_p$  へクエリ転送メッセージを送信する. このとき, クエリ転送先として  $M_p$  の識別子を保持する. クエリ転送メッセージには, クエリ発行端末  $M_s$  の識別子と位置情報, および返信データが含まれている. 一方,  $M_t$  宛ての次端末返信メッセージを傍受した端末  $M_u$  は,  $WT$  を停止し, 次端末返信メッセージの返信を行わない.
- (7) クエリ転送メッセージを受信した端末  $M_p$  は, 手順 (2) へ戻り, 始点端末としてクエリの伝搬を続ける.

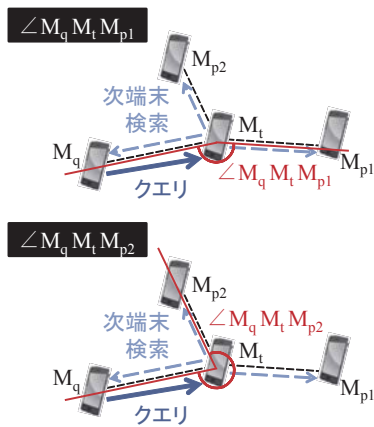


図 4 LW 法におけるなす角

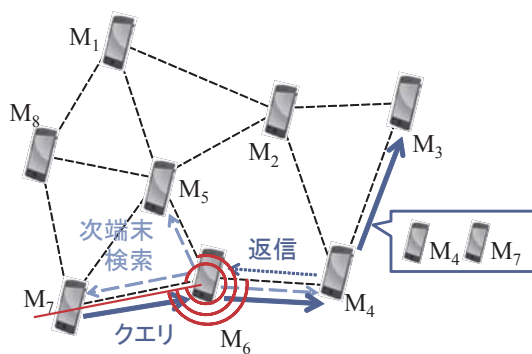


図 5 LW 法によるクエリ転送例

(8) 端末  $M_t$  は、3.3.1 項で示したジオルーティングを用いて検索クエリ発行端末までクエリ応答メッセージを送信する。この時の指定位置は、クエリ発行端末の位置である。クエリ応答メッセージには、クエリ発行端末  $M_s$  の識別子と位置情報、および返信データが含まれる。

図 5 を用いて、LW 法で端末  $M_7$  からクエリを受信した端末  $M_6$  の動作について説明する。端末  $M_6$  は、次端末検索メッセージを隣接端末へ送信する。これを受信した端末  $M_4$ ,  $M_5$ , および  $M_7$  は、 $\angle M_7 M_6 M_{4(5,7)}$  の大きさを基に WT を算出する。最もなす角の小さい端末  $M_4$  が最も早くクエリ要求メッセージを送信し、端末  $M_6$  はクエリ転送メッセージを  $M_4$  へ送信する。クエリ要求メッセージを傍受した  $M_5$  および  $M_7$  はメッセージの送信を停止する。 $M_6$  からクエリ転送メッセージを受信した端末  $M_4$  は、同様の手順により  $M_3$  へクエリ転送メッセージを送信する。この際、 $\angle M_4 M_6 M_7$  の大きさが  $\pi$  未満となるため、 $M_6$  のデータを破棄し、 $M_4$  と  $M_7$  のデータを返信データとして添付する。

LW 法では、ネットワーク全体へクエリの転送を行う必要がないため、小さなトラヒックにより凸包の頂点の情報を取得することができる。また、転送元の端末とのなす角を基に返信の待ち時間を設定し、次にクエリを転送する端

表 1 メッセージの種類とサイズ

メッセージの種類	サイズ [B]
クエリ	64
次端末検索 (LW 法)	48
次端末返信 (LW 法)	16
クエリ転送 (LW 法)	$64+32p$
クエリ応答	$64+32p$
応答受信 Ack	16

末を選択することで、全ての隣接端末の位置情報を取得することなくクエリの転送を行うことができ、検索時間を短くすることができる。

## 4. 性能評価

本章では、提案手法の性能評価のために行ったシミュレーション実験の結果を示す。本実験では、ネットワークシミュレータ Qualnet5.2[8] を用いた。

### 4.1 シミュレーション環境

1,000[m] × 1,000[m] の 2 次元平面上の領域に  $n$  台の端末 ( $M_1, \dots, M_n$ ) が存在する。各端末はランダムウォークモデル [2] に従い、0.1~ $v$ [m/s] の速度で移動するものとした。各端末は、IEEE802.11b を使用し、伝送速度 11[Mbps]、通信伝搬距離が 100[m] 程度となる送信電力でデータを送信する。また、各端末は自身の位置を常に把握しているものとした。ここで、式 (1) 中の  $\alpha$ 、および式 (2) 中の  $Max\_delay$  は、事前実験を基にそれぞれ  $(1.0+n/250)$ 、0.07 と設定した。

比較手法として、クエリ発行端末がネットワーク全体へクエリを送信し、全ての端末の位置情報を取得する単純手法を用いた。基本的な動作は LCH 法と同様であるが、式 (1) 中のパラメータ  $\alpha$  を  $(1.0+n/250)$  および  $2 \times (1.0+n/250)$  とした 2 通りについて評価した。図中では、それぞれ「単純」、「単純 (時間 2 倍)」と表記する。

以上のシミュレーション環境において、ネットワーク内の端末数および端末の最大速度を変化させ、各手法の性能を評価した。各端末の初期位置をランダムに決定し、10 秒ごとに 50 回ランダムに選択した端末が凸包検索クエリを発行し、以下の評価値を調べた。

- トラヒック  
凸包検索クエリを処理するために送信されたメッセージの平均バイト数 (1 クエリ当たりの平均)。表 1 は提案手法、および比較手法における全てのメッセージのサイズを示す。この表において、 $p$  はメッセージに情報が含まれる端末の数である。
- 応答時間  
クエリ発行端末が検索クエリを発行してからクエリ応答を受信するまでの平均時間 (1 クエリ当たりの平均)。
- 取得精度

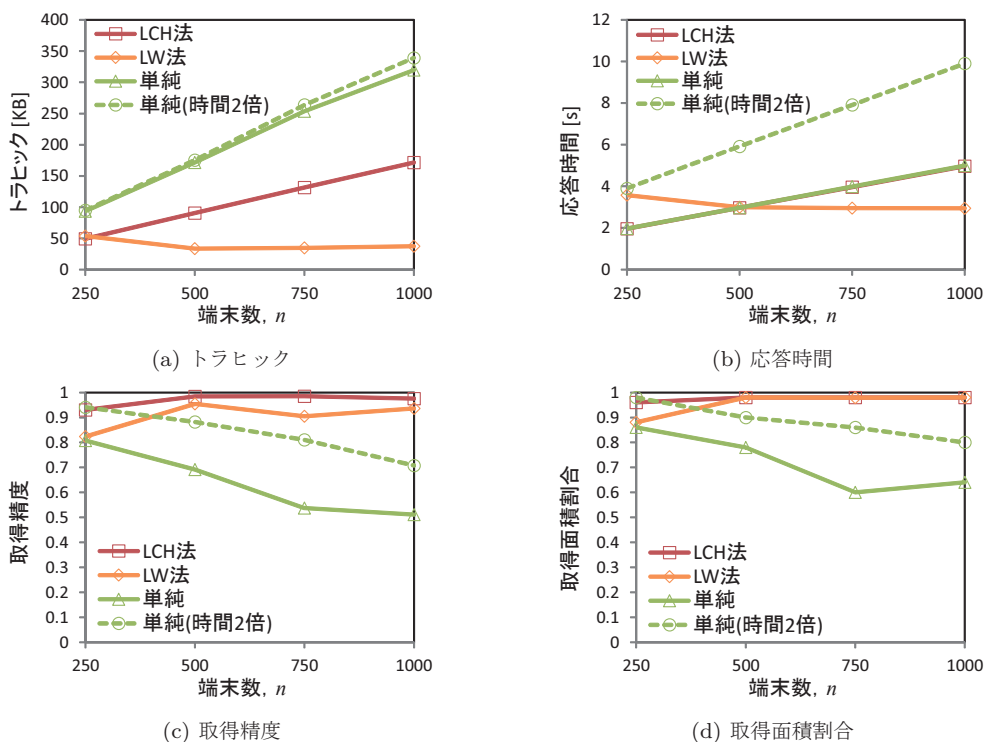


図 6 端末数  $n$  の影響

クエリ発行時点のネットワークの凸包の頂点となる端末のうち、クエリが情報を取得できた端末数の割合 (1クエリ当たりの平均)。

● 取得面積割合

クエリにより情報を取得できた場合の、クエリ発行時点のネットワークの凸包の面積に対する検出した凸包の面積の割合 (1クエリ当たりの平均)。

4.2 端末数  $n$  の影響

端末数  $n$  を変化させた場合の結果を図 6 に示す。これらの図中のグラフにおいて、横軸はネットワーク内の端末数  $n$  を表し、縦軸は、図 6(a) はトラヒック、図 6(b) は応答時間、図 6(c) は取得精度、図 6(d) は取得面積割合を表す。

図 6(a) の結果より、端末数  $n$  が増加すると、LCH 法および単純手法ではトラヒックが大きくなることわかる。これは、ネットワーク全体にクエリを転送する際、全ての端末がクエリを送信するためである。さらに、単純手法では、全ての端末が位置情報を返信するため、端末数が増加するにつれ、非常に大きなトラヒックが発生する。一方、LW 法では、 $n$  が増加すると、若干トラヒックが減少している。LW 法では、ネットワーク全体へクエリを転送する必要はなく、トラヒックは基本的にクエリ転送のホップ数に依存するため、端末数の影響は小さい。ただし、 $n$  が小さくなる (端末の密度が低下する) と、クエリ転送時にリンク切断が起きやすくなるため、メッセージの再送によりトラヒックが増加する。

図 6(b) の結果より、LCH 法および単純手法では、 $n$  が大きくなるにつれて応答時間が長くなっていることがわかる。これは、返信の際にメッセージの衝突を避けるために、返信待ち時間を端末数に応じて設定するためである。一方、LW 法では、 $n$  が大きくなると、応答時間が短くなっている。これは、トラヒックについての考察で述べたとおり、端末の密度の低下によりクエリ転送時にリンク切断が起きやすく、メッセージの再送により応答時間が長くなるためである。

図 6(c) の結果より、LCH 法では、高い取得精度を維持できていることがわかる。一方、LW 法は LCH 法と比べると取得精度が低くなっている。これは、LW 法では、クエリを一本経路で転送しており、リンク切断による転送失敗の影響を受けやすいためである。特に、 $n$  が 250 の場合、端末の密度が疎となり、リンク切断の機会が増加するため、取得精度が低下している。単純手法は、提案手法と比べると取得精度が低くなっている。これは、返信するデータが多く、メッセージの衝突が頻繁に発生するためである。特に、 $n$  が大きくなるにつれて返信するデータ数が増加し、メッセージの衝突がより頻繁に発生するため、取得精度は大きく低下してしまう。ここで、単純手法において、待ち時間を 2 倍にした場合、取得精度はある程度高くなるが、図 6(b) で示されるとおり、応答を取得するまでの時間が長くなってしまふ。

図 6(d) の結果より、提案手法では、凸包の面積をほぼ達成できていることがわかる。しかし、 $n$  が 250 のとき、LW

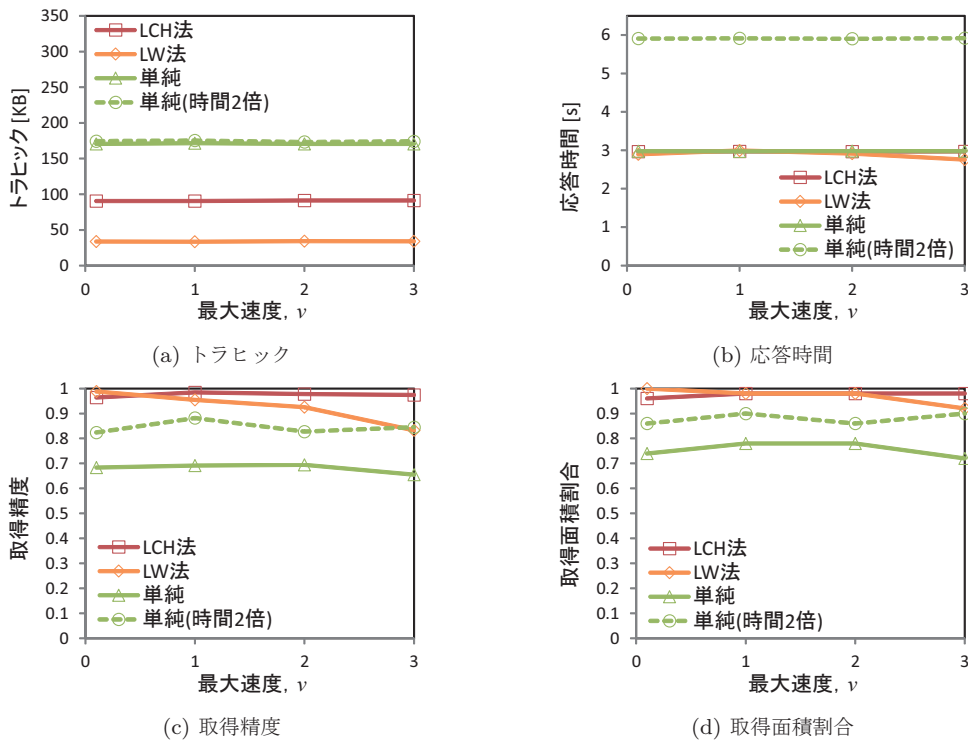


図 7 最大速度  $v$  の影響

法では、割合が低くなっている。これは、一時的に孤立する端末が発生し、この端末へクエリを送信できない場合が生じて、本来の凸包と比べて小さな領域を凸包として検出したためである。単純手法では、図 6(c) が示すとおり、パケットロスにより凸包の頂点となる端末の情報を取得できないため、凸包を適切に検出することができないことがわかる。

### 4.3 最大速度 $v$ の影響

最大速度  $v$  を変化させた場合の結果を図 7 に示す。これらの図中のグラフにおいて、横軸は最大速度  $v$  を表し、縦軸は、図 7(a) はトラヒック、図 7(b) は応答時間、図 7(c) は取得精度、図 7(d) は取得面積割合を表す。

図 7(a) の結果より、全ての手法で端末の速度に関わらず、ほぼ一定のトラヒックとなることがわかる。図 7(b) の結果より、全ての手法でほぼ一定の応答時間となっていることがわかる。これは、応答時間は LCH 法および単純手法では  $\alpha$  の値、LW 法では  $Max.delay$  の値に依存するためである。

図 7(c) の結果より、LCH 法および単純手法では、 $v$  の値に関わらずほぼ一定となっている。特に、LCH 法は、 $v$  が大きくなっていても非常に高い取得精度を維持できている。一方、LW 法では、 $v$  が大きくなるにつれて取得精度が低下している。これは、端末の速度が大きくなると、クエリ転送の間に凸包の頂点となる端末の集合が変化する可能性が高く、クエリの伝搬に時間がかかる LW 法では、取得精

度が低下する原因となるためである。

図 7(d) の結果より、提案手法では、凸包の面積をほぼ達成できていることがわかる。特に LW 法では、9 割以上を達成しており、図 7(c) で示した LW 法の取得精度と比べて高くなっている。これは、凸包の頂点となる端末の集合が変化するため取得精度は低下するが、凸包のサイズは十分に把握できることを表している。

## 5. 議論

提案手法では、基本的にネットワークの形状に依存することなく検索結果を取得することができる。例えば、図 8 のような凹みのある形状のネットワークの場合でも、問題なく動作する。ただし、LW 法は、取得時間がクエリを伝搬する際のホップ数、つまりネットワークの外周の長さに依存するため、図 9 のような場合に、取得時間が大きくなってしまふ。クエリの伝搬に大きく時間がかかると、移動により凸包が変化してしまい、取得精度が低下する可能性が高くなると考えられる。

## 6. 結論

本稿では、アドホックネットワークにおいて、トラヒックの削減、検索時間の低減、および検索結果の取得精度の維持が可能な凸包検索手法を提案した。提案手法では、検索クエリの受信により取得した情報より、各端末が自律的に返信する情報を判断する。そのための方法として、LCH 法と LW 法の二つの手法を提案した。LCH 法では、ネット

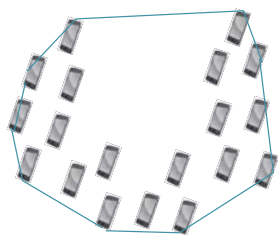


図 8 凹みのある形状のネットワーク例

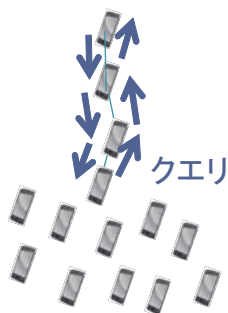


図 9 外周が長くなるネットワークの例

ワーク全体にクエリを送信し、返信時に、中継端末において受信した返信データを基に、ローカルな凸包検出を行い、ローカルな凸包の頂点となる端末の情報のみを返信する。一方、LW法では、まず  $y$  座標が小さい端末へクエリを転送し、その後、ネットワークを囲うように、ネットワークの外周に存在する端末へ順にクエリを伝搬することで、凸包の頂点となる端末の情報を取得する。提案手法により、ネットワーク内の全ての端末の位置情報を取得することなく、凸包の頂点となる端末の情報を取得することができる。

シミュレーション実験の結果から、提案手法では、全ての端末の位置情報を取得する単純手法と比べ、トラフィックを大きく削減し、検索時間を短縮しつつ、高い取得精度を維持できていることがわかった。特に、LCH法では、端末数や速度に依存することなく、高い取得精度を維持できていることがわかった。一方、LW法は、全ての端末へクエリを送信する必要がなく、端末数が大きい場合にも、小さなトラフィックおよび短い検索時間を達成できる。

本稿において、LCH法は、高い取得精度を維持できる一方、端末数が大きい場合には、トラフィックが増大していた。これは、ネットワーク内の端末へクエリを転送する際、全ての端末がクエリの送信を行うためである。そのため、今後は、ネットワーク内の必要な端末へのクエリの転送を保証しつつ、クエリ転送にかかるトラフィックを削減するように手法を拡張する予定である。また、本稿では凸包の検索は単発のものを想定していたが、実際には連続的にネットワークの凸包を把握したいといった要求もあると考えられる。そのため、アドホックネットワークにおける連続的な凸包検索手法についても検討を行う予定である。

## 謝辞

本研究の一部は、文部科学省科学研究費補助金・基盤研究 B(24300037) の研究助成によるものである。ここに記して謝意を表す。

## 参考文献

- [1] C.B. Barber, D.P. Dobkin, and H. Huhdanpaa, "The quickhull algorithm for convex hulls," *ACM Trans. on Mathematical Software*, Vol.22, No.4, pp.469-483, 1996.
- [2] T. Camp, J. Boleng, and V. Davies, "A Survey of Mobility Models for Ad Hoc Network Research," *Wireless Communications and Mobile Computing (WCMC)*, Vol.2, No.5, pp.483-502, 2002.
- [3] D. Dong, Y. Liu, and X. Liao, "Fine-grained boundary recognition in wireless ad hoc and sensor networks by topological methods," *Proc. MobiHoc*, pp.135-144, 2009.
- [4] R.L. Graham, "An Efficient Algorithm for Determining the Convex Hull of a Finite Planar Set," *Information Processing Letters*, Vol.1, pp.132-133, 1972.
- [5] R.A. Jarvis, "On the identification of the convex hull of a finite set of points in the plane," *Information Processing Letters*, Vol.2, No.1, pp.18-21, 1973.
- [6] Y. Komai, Y. Sasaki, T. Hara, and S. Nishio, "KNN Query Processing Methods in Mobile Ad Hoc Networks" *IEEE Trans. on Mobile Computing*, Vol.13, No.5, pp.1090-1103, 2014.
- [7] K. Matsuo, A. Kanzaki, T. Hara, and S. Nishio, "Overhearing-based Efficient Boundary Detection in Dense Mobile Wireless Sensor Networks" *Proc. MDM*, pp., 2014.
- [8] Scalable Network Technologies: "Qualnet," <http://www.scalable-networks.com/>.
- [9] H. Zhou, H. Wu, and M. Jin, "A robust boundary detection algorithm based on connectivity only for 3D wireless sensor networks," *Proc. INFOCOM*, pp.1602-1610, 2012.