

# P2P ネットワークにおける位置に関連した情報の 管理・検索手法の提案と評価

松浦知史<sup>†</sup> 新井イスマイル<sup>†</sup> 中村豊<sup>††</sup> 藤川和利<sup>††</sup> 砂原秀樹<sup>††</sup>

E-mail: {sato-mat, ismail-a}@is.naist.jp, {yutaka-n, fujikawa}@itc.naist.jp, suna@wide.ad.jp

<sup>†</sup> 奈良先端科学技術大学院大学情報科学研究科

<sup>††</sup> 奈良先端科学技術大学院大学情報科学センター

## 概要

移動可能な通信端末の発展により位置情報サービスへの要求が高まっている。また多くの通信端末がいたるところに存在しており、端末の持つ情報を位置に基づいて収集・加工することおよび端末が位置に基づいて情報を自ら流通させることが要求されている。そこで本論文では、P2P ネットワーク上において、位置に関連した情報の管理・検索手法を提案する。2次元座標を1次元の円周上に対応させることによって情報管理や検索の手法が単純化され、計算量  $O(\log N)$  で情報の検索が可能である。また一意な検索が可能のため情報の有無を確認でき、加えてフラッシングを用いる他の P2P システムと比べてネットワークに対する負荷が少ない。シミュレータを作成し提案システムを評価した結果、既存の DHT を利用したアルゴリズムと同程度の検索速度を示しながら、位置に基づく情報の管理・検索が可能である事を検証した。

## 1 はじめに

無線技術の向上や携帯電話、PHS の普及を受け、人や自動車などの移動ノード（以下、通信可能な計算機端末をノードと呼ぶ）は時間や場所などの制約を受けずにネットワークに接続可能である。移動端末の発展を受け、例えば周辺の天気や目的地の渋滞情報のように位置に依存するサービスへの要求が高まっている。

また PDA などの小型端末や自動車に搭載される車載器などの廉価化が進み、移動ノード数に関しても大幅な増加が予想される。増加するノードに対して全てのサービスをサーバ・クライアントモデルで実現すると、サーバに対してネットワーク帯域や記憶媒体などの負荷がさらに集中することが予想される。このような膨大な数のノードが生成するトラフィックやデータを扱うためには、情報を分散管理する機構が必要となる。

情報を分散管理する手法の一つとして Peer to Peer(以下、P2P) ネットワークに注目が集まってい

る。P2P ネットワークでは各ノードが情報やデータを管理し、対等な立場でノード間が直接通信することによって負荷分散を実現している。またネットワーク帯域の負荷や計算処理の負荷が局所的にかかる集中型データベースや集中型インデックスを用いることなしに、情報の分散管理が可能である。しかし既存の P2P ネットワークは比較的静的なネットワーク構成を前提としており、地理的に移動するノードが多い環境は基本的に想定していない。

本研究では位置に依存したサービスを提供するために、移動ノードを含めた多くのノード間でインターネットを利用し、位置に関連した情報を管理・検索が可能な P2P ネットワークの形成手法を提案する。そのための必要要件としては以下の3点が挙げられる。

- 位置に基づいて必要な情報を検索できること
- 位置に基づいた情報を蓄積し、提供できること
- 情報提供側に負荷が一極集中しないこと

情報は位置や時間、状況によってその価値を変えていく。刻々と位置を変えていく移動ノードにとって、これから通る道路の渋滞情報のように目的とする位

置と対応した情報はより有意な価値を持つ。例えばここで自動車の環境を考える。ある山道を走行中に凍結や濃霧、事故や落石といったその場所特有の情報を取得したとする。このようなその土地に関連した情報はその土地に関連付けて保存しておくのが適切であると考えられる。そうすることによって後続車やのちに山道を通過する自動車が情報を検索可能となる。また他の車両が取得した気象情報や事故情報を統合することでより精度が高く有用な情報が生成されることが考えられる。そのため位置に関連付けを行いながら情報を蓄積する仕組みと、蓄積された情報を位置に基づいて取得する仕組みが必要である。

情報を位置と関連させ、インターネット上で流通させる場合に従来のサーバ・クライアントモデルではサーバの負荷が必要以上に高くなる可能性がある。なぜなら、ノード数が今後も増加し続けることが予想され、現在よりもさらにサーバに対してネットワーク帯域や記憶媒体の負荷が一極集中すると考えられるからである。

サーバの負荷分散の手法としてロードバランサ(負荷分散装置)などの専用機器を利用する方法が考えられる。ロードバランサはOSI7階層モデルにおけるレイヤー4(L4)やレイヤー7(L7)でのスイッチングができ、複数台のサーバを用意することで負荷分散が可能である。しかし専用機器の購入や新たなサーバの準備、運用管理の必要がある。そのため、初期投資や管理にかかるコストが生じる。

P2Pネットワークを用いた場合、各ノードが情報を管理し合いノード同士が直接通信することにより負荷分散が実現できる。P2Pネットワークでは情報を蓄積する負荷や計算処理にかかる負荷およびネットワーク帯域の負荷が局所的に発生しないように設計されている。またノードの持つ計算機資源を利用しているため、ノード数が増加した場合に新規に機器を購入して対応するなどのコストが発生しない。加えてP2Pネットワークに参加しているノードが負荷を分散し合うので新たな管理コストも発生しないという利点がある。

そこで本研究では、P2Pネットワーク上で位置に基づく情報の管理・検索が可能なる手法を提案する。2次元座標を1次元の円周上に対応付けることにより、情報の管理・検索手法が単純化され、 $O(\log N)$ の計算量で検索可能である。他のフラッディングを用いる

P2Pネットワーク構築手法と比べて、一意な検索が可能であり検索クエリーの数を抑えることができ、情報の有無も確認できる。提案手法は既存の分散ハッシュテーブル(DHT)を用いた手法と同程度の検索速度を実現し、また特定の位置に基づいた情報の検索が可能となっている。

## 2 関連研究

### 2.1 既存の位置情報サービス

ここでは既存の位置情報サービスの提供方法についてまとめる。

移動ノードが位置情報サービスを受ける例として、外出時に周辺の天気、及びこれから移動するであろう場所、自宅周辺などの天気の状態を取得する場合を考える。このような情報は例えば気象庁が観測地点のデータを収集し、まとめて天気予報として発表している。

従来の統計情報のサービスというものは

1. 高価な観測機を多数設置
2. 中央のセンターにデータを収集
3. 集約情報を配信

という手法をとっていた。このシステムではサービスに対して独自のインフラを用意する必要がある。そのためにサービスごとに新しい機器を用意する必要があり、サービスの導入コストが大きい。

近年では移動ノードとインターネットが繋がり、インターネットに接続可能なセンサー群が街中を動き回っている。このような状況で統計情報のサービスのあり方も見直されている。日本自動車研究所(JARI: Japan Automobile Research Institute) [1]の試みでは、自動車に取り付けたワイパーセンサのデータ等をインターネットを利用してサーバに収集することにより詳細な降雨情報などの生成が可能となっている。JARIでは下記のような手法を取り入れインターネットを利用した情報の収集、サービスの提供を実現している。

1. 人や自動車が付随するセンサを通してデータを取得
2. 中央サーバにインターネットを介して情報を提供

表1 既存研究の比較

	位置に基づく情報検索	位置に基づく情報の蓄積	管理コスト	負荷分散
従来の位置情報サービス	○	○	×	×
JARIの位置情報サービス	○	○	×	×
既存のP2P	×	×	○(△)	○
LL-net	△	×	△	○

### 3. 中央サーバが情報を収集し、集約情報を配信

インターネットを介した柔軟なデータの収集方法はインフラとサービスを分離することに成功した。取り出すデータフォーマットに統一性があれば一つのデータを異なるサービスに利用できる。しかしサーバ・クライアント型のシステムであるため、今後さらなるノードの増加にサーバに対してネットワーク帯域などの負荷がより高まる可能性がある。

## 2.2 既存のP2Pネットワーク

P2Pネットワークを構築する手法としては多くのものが提案されている。Napster[2]に代表されるハイブリッド型のP2Pネットワークではコンテンツの情報を中央サーバが保持する。参加ノードは中央サーバでコンテンツの場所を検索し、次にコンテンツを保持しているノードと直接通信をすることで目的とする情報を得る。Gnutella[3]やFreenet[4]のピア型と呼ばれるP2PネットワークではNapsterのような特別なインデックスサーバを必要としない。しかし検索時にはフラッキングを用いるために通信回線に対して非常に多くの負担がかかるという欠点を持つ。特別なインデックスサーバなどを用いることなく検索の負荷も軽減しようと近年ではChord[5]、CAN[6]、Tapestry[7]、Pastry[8]など分散ハッシュテーブル(DHT)を利用したものが注目されている。さらにTapestry、Pastryに関してはアルゴリズムを改良することによりネットワークポロジを考慮したオーバーレイネットワークの構築が可能となるよう拡張されているものもある。これら既存の代表的なP2Pネットワークにおいては、全くランダムにまたはノードの持つファイルの種類やユーザーの嗜好を基にしてオーバーレイネットワークが形成される。そのため上記で挙げた手法はどれも直接位置情報を扱うものではない。

位置情報を考慮したP2PネットワークとしてはLL-net[9]がある。LL-netでは物理的な空間をあらかじめ小さなエリアに分割し、そのエリアに振ったIDをもとに位置情報に基づいたオーバーレイネットワークを構築する。しかしLL-netの仕組みとして

- エリア全体を管理する特別なノード(スーパーピア)が必要
- 広域検索時には全てのノードにクエリーを投げる
- エリアの大きさが固定
- 領域指定検索ではエリアを辿っていくため遠方領域の検索負荷が高い

などがあり、制約が多い。またLL-netはノードの検索を主眼にしているため、位置に基づいた情報の蓄積はできない。

表1において既存研究の比較を示した。既存のP2Pシステムではハイブリッド型の場合インデックスサーバを管理する必要があるために管理コストを△とし、ピア型では特定のマシンを管理する必要が無いので○とした。LL-netにおいては検索領域を指定して、その領域内に存在するノードを発見できる。しかしノードの持っている情報は常にノードの存在場所と一致するわけではない。例えば、ノードが存在する場所は東京の職場であり、ノードが保持する情報は横浜にある自宅の情報である場合などが考えられる。そのために位置に基づく情報検索は△とした。またスーパーピアを用いてエリアを管理する必要があるために管理コストは△とした。表1における4つの条件を全て満たすシステムは存在しない。

表2 検索ホップ数の比較

	検索ホップ数
提案システム	$O(\log N)$
Chord	$O(\log N)$
Pastry	$O(\log N)$
CAN	$O(d * N^{1/d})$
LL-net	$O(\sqrt{N})$

N: ノード数 (LL-net ではランデブーピアの数)

d: 次元数 (CAN ではハッシュ空間として N 次元トーラスを利用している。その時の次元数を d とする)

### 3 P2P ネットワークにおける位置に関連した情報の管理・検索

ここでは P2P ネットワーク上において位置に基づいた情報の管理および検索手法を提案する。まず提案システムにおける前提条件を示し、本研究で提案する情報の管理・検索手法の動作プロトコル、および情報の管理・検索アルゴリズムについて詳しく述べる。

#### 3.1 前提条件

提案する P2P ネットワークに参加する全てのノードは以下の前提条件を満たすものとする。

- 自分の位置情報を取得する能力を持つ (GPS 機器を備えるなど)
- インターネットを介して通信が可能である
- データを送受信すること、およびデータの検索や収集・加工が可能である

現在、携帯電話や PDA の多くはインターネットに接続する機能を既に持っており、車載器においても一部のものはインターネットの接続性を有する。また GPS の小型化、廉価化が進み多くの携帯電話や車載器に搭載されており、GPS が搭載されていない端末に関しても専用の GPS 機器を接続可能となっている。今後これらの傾向はさらに強まり、上記で挙げた前提条件を有するノードは増加していくと予想される。

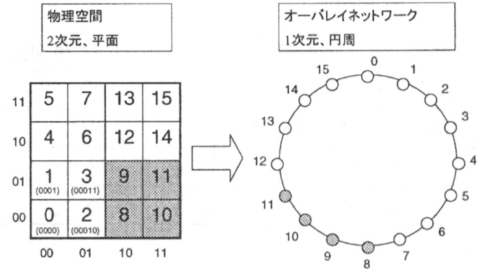


図1 2次元から1次元へのマッピング

#### 3.2 動作プロトコル

##### 3.2.1 2次元から1次元へのマッピング

提案手法では 2 次元座標から 1 次元の円周上への対応付けを行っている。1 次元の ID を利用することで  $O(\log N)$  の高速な検索が可能となり、2 次元上で検索するよりも検索効率が高い。検索手法に関しては 3.3.3 検索アルゴリズムの改良で述べる。表 2 に代表的な DHT と LL-net における検索ホップ数を示す。既存の CAN や LL-net などの 2 次元以上の ID を利用する P2P ネットワークでは、情報を取得する場合に隣接したノードを辿る検索手法を用いている。そのため検索に必要な計算量が 2 次元の ID を用いた場合は  $O(\log N)$  よりも計算量が多い  $O(\sqrt{N})$  である (LL-net では  $N$  = ランデブーピアの数。ランデブーピアとはエリアを代表するノードのことである)。

ノードは自分の位置情報 (2 次元座標) から 1 次元の ID を生成する。図 1 に 2 次元座標から 1 次元の円周への対応付けを図示する。網掛けの部分 (ID:8,9,10,11) を設け、対応箇所を示している。1 次元の ID を元にノードはオーバーレイネットワークに参加を行ったり、情報の管理・検索に利用する。1 次元の ID は 2 次元座標の各桁の値を交互に入れ替えて生成する。例えば取得した位置情報が (123, 789) だとすると生成される 1 次元の ID は 172839 となる。各桁を交互に入れ替えて生成した ID には下記のような特徴がある。

- ID を 1 ずつ増加させると対応する 2 次元座標は正方形を形作る
- ID を 1 ずつ増加し続けると、小さい正方形か

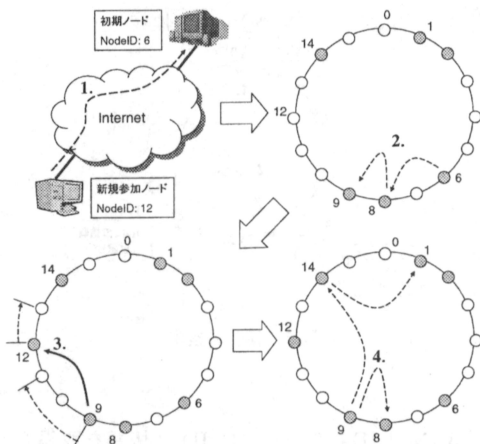


図2 オーバレイネットワークへのノードの参加

ら始まり大きな正方形を作る

このような特徴は領域検索時に領域の分割を抑え、検索速度を向上させるために有効に働く。詳しくは3.3.2 領域の検索で述べる。

### 3.2.2 オーバレイネットワークへの参加離脱プロトコル

ここではノードがどのようにしてオーバレイネットワークに参加するか説明する。参加希望ノードはオーバレイネットワークにすでに参加しているノードのIPアドレスを少なくとも一つ事前に知っている必要がある。この既知のノードを”初期ノード”と呼ぶ。参加希望ノードは自分の位置(2次元)からID(1次元)を生成し、そのIDとIPアドレスを元にオーバレイネットワークへ参加を試みる。参加時に生成したIDを”NodeID”と呼ぶ。参加希望ノードは初期ノードに対してNodeIDとIPアドレスを伝える。参加ノードからのクエリーはまず初期ノードへ届きそこから右回りにクエリーが順に渡され、参加希望ノードが加わるべき場所(円周上での該当するID)の隣接ノードまで伝播される。隣接ノードは参加希望ノードと通信し、そのノードをオーバレイネットワークへ参加させる。また新規参加ノードは同時に担当領域を割り当てられる。担当領域は新規参加ノードのNodeIDから右隣のノードのNodeIDの一つ手前までである。

新規参加ノードが生成したNodeIDが既に使用さ

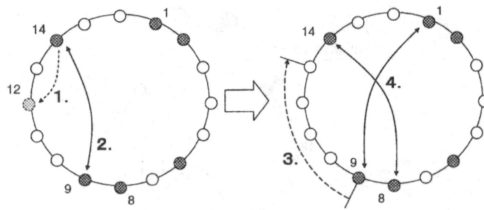


図3 オーバレイネットワークへからの離脱

れている場合は、オーバレイネットワーク上でそのNodeIDを持つノードが参加希望ノードを参加させる役割を担う。既にそのNodeIDを持つノードは担当領域の中心に該当するIDを新規参加ノードのNodeIDとして割り当て、参加希望ノードをオーバレイネットワークに参加させる。しかしID空間としては $2^{128}$ などの広い空間を通常使用するので衝突が起こる確率は極めて低い。

オーバレイネットワークへ参加したノードは定期的に二つ隣りまでのノードとのリンク状態を監視する。これはノードの離脱が起こった場合、担当領域や近隣ノードのリンクを再構築するためである。

ここでは例としてNodeID12のノードがオーバレイネットワークへ参加する様子を示す。このノードは初期ノードとしてNodeID6のノードのIPアドレスを知っているものとする。図2にオーバレイネットワークへの参加の様子を図示する。ID空間を白丸(○)で表し、ノードが既に参加している箇所のIDは黒丸(●)で表す。

1. NodeID6のノード(初期ノード)にNodeID(12)とIPアドレスを伝える
2. NodeID12を担当しているノード(NodeID9)を検索
3. NodeID9のノードが参加を許可し、同時に担当領域を割り当てる。新規参加ノードの担当領域はID12,13である。NodeID9のノードの担当領域はID9,10,11に切り替わる
4. 新規参加ノードの二つ隣りまでNodeIDとIPアドレスが伝わる

ノードがインターネットへの接続性を失い離脱すると、隣接ノードがその事を知り、離脱ノードが抜けた部分を修復する。ここではNodeID14のノードがNodeID12のノードの離脱を知った場合を例に取り、

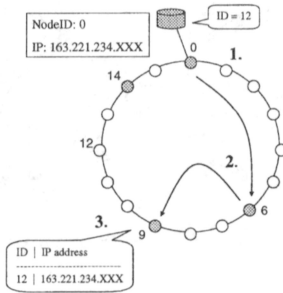


図4 情報の登録

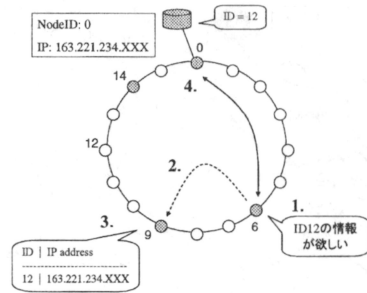


図5 情報の検索

修復時の動作を示す。図3にオーバレイネットワークからの離脱の様子を図示する。

1. NodeID14のノードは隣接したNodeID12のノードが離脱したことを知る
2. NodeID9のノードに新しく隣接ノードになったことを伝える
3. NodeID9は担当領域を再設定する
4. NodeID14のノードはNodeID8のノードを、NodeID9のノードはNodeID1のノードをそれぞれ二つ隣のノードとして登録する

### 3.3 位置に基づく情報の管理・検索

#### 3.3.1 情報の管理・検索アルゴリズム

ノードはセンサデバイスで温度や速度を測定したり、レストランやガソリンスタンドなどの店舗のサービスを調べたり、カメラ等で画像を撮影したりなど何らかの情報を取得する能力を有しているとする。ノードは情報を取得したと同時に位置検出デバイスを用いて取得した情報と位置情報とを関連付ける。そしてその位置情報からIDを生成し、そのIDを担当領域内にもつノードを検索し、IDとIPアドレスを登録する。ここでは例としてNodeID0のノードがID12の情報を取得したとして、動作を示す。NodeID0のノードはIPアドレスが163.221.234.XXXとする。図4に情報登録の様子を図示する。

1. 情報取得時に位置(2次元座標)からIDを生成(ここではID12)
2. ID12を担当領域内にもつノード(NodeID9)を検索

3. NodeID9のノードにIDと情報を取得したノード(NodeID0)のIPアドレスを登録

次に検索アルゴリズムを示す。ノードが位置に基づいた情報を得たい場合、まず位置からIDを生成する。そのIDを”KeyID”と呼び、KeyIDに基づいて検索する。自分の場所から右回りにクエリーを伝播させ、KeyIDを担当しているノードを探し出す。KeyIDを担当しているノードが分かったら、そのノードが持っているIDとIPアドレスのテーブルから実際に情報を持つノードのIPアドレスを調べる。次に検索元のノードと実際に情報を持つノードが直接通信し、情報を取得する。ここでは例としてNodeID6のノードが位置情報(KeyID12)に関連する情報を検索する様子を図5に示す。ここではNodeID0のノードのIPアドレスが163.221.234.XXXだとする。

1. 情報を取得したい位置からKeyIDを生成する(ここではKeyID12)
2. ID12を担当しているノードを右回りにたどり、探し出す
3. ID12を担当しているのはNodeID9のノードであり、ID12の情報を持っているノード(NodeID0)のIPアドレス(163.221.234.XXX)を検索元に教える
4. 検索元のノード(NodeID6)は情報を持つノード(NodeID0)と直接通信をし、情報を取得する

#### 3.3.2 領域の検索

実際に情報を検索する際にどこか一点のKeyIDに対応する情報を探すことは少なく、特定領域内の検

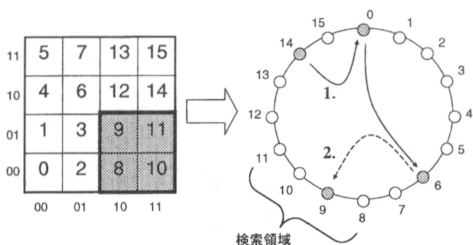


図6 領域の検索

索が機能として望まれる。提案システムでは

- ID を 1 ずつ増加させると、対応する 2 次元平面では正方形を形作る
- 各ノードは昇順に連続した ID を担当領域として管理している

といった特徴があり、そのために正方形の領域検索に対して特に力を発揮する。

領域を検索する場合は、連続した ID を一つのクエリーとする。連続する ID の先頭 ID を”StartKeyID”，最後の ID を”EndKeyID”とする。まず StartKeyID を担当領域に持つノードを検索し、その後は右隣のノードをたどることにより領域検索が可能である。ここでは図 6 における右下の正方領域を検索する場合を例に取り、領域検索の様子を示す。検索元は NodeID14 のノードである。このノードが右下の正方領域である KeyID8,9,10,11 に関連した情報を検索するとする。この場合 StartKeyID は 8, EndKeyID は 11 となる。ここでは NodeID6 のノードが ID8 に関連する情報を、NodeID9 のノードが ID9,10,11 に関連する情報を管理している。領域検索の様子を図 6 に示す。

1. StartKeyID8 を持つノード (NodeID6) を検索する
2. 右隣のノード (NodeID9) へクエリーを伝播させる

上記で述べた領域検索の例は最良の性能を示す例で、中央部分の正方領域 (ID3,6,9,12) を選んだ場合は ID が連続しないため 4 つの検索クエリーを送信する必要がある。具体的には一つ目のクエリーが StartKeyID3, EndKeyID3 となり、二つ目のクエリーは StartKeyID6, EndKeyID6 と 4 つのクエ

表 3 検索回数の比較

	最良の検索回数	最悪の検索回数
提案手法	1	4
座標の連結	$2^{62}$	$2^{62}$

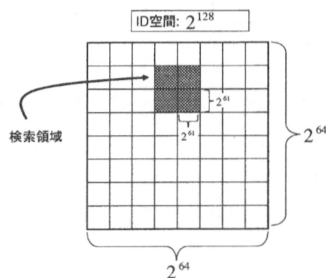


図7 領域の検索

リーが生成される。しかしこれは提案システムを否定する重大な要素ではない。

ここでは 2 次元の座標を単純に連結した ID 生成法と比較する。例えば座標が (123, 789) ならば x 座標と y 座標を連結させ ID は 123789 となる。このような ID の生成方法では ID を 1 ずつ増加させるにしたがって、2 次元平面では y 座標の値が増えていき y 軸の最大値を越えると x 座標の値が一つ増え y 座標の値が 0 になるという動作を繰り返す。ID 空間が  $2^{128}$  の空間を用意して検索する場合を例にとつて考えると、上記の x 座標と y 座標を連結させた ID の生成方法は検索効率が悪い。

図 7 において塗りつぶされた 4 マスの領域を検索する場合を考える。座標の連結で ID を生成した場合は検索領域の内部で ID が連続せず、検索領域の y 方向の最大値を何度も超える。検索回数の比較を表 3 に示す。図 7 において座標を連結する手法で ID を生成すると ID が  $2^{62}$  回分断され、領域を検索するのに  $2^{62}$  の検索クエリーを送信する必要がある。これは検索ノードやネットワークに多大な負荷を与える原因になる。一方で提案する ID の生成方法では任意の正方 4 マスを検索領域とした場合、最良で 1 回の検索クエリーを送信すればよく最悪でも 4 回のクエリーで済む。なぜならば ID を 1 ずつ増加し続けると対応する 2 次元平面では小さな正方形から大きな正方形を描くようになっているためである。図 7 で 64 分割

表4 NodeID18の持つルーティングテーブル

ホップ数	NodeID	IP アドレス
1	20	163.221.10.AAA
2	21	163.221.22.BBB
4	25	163.221.135.CCC
8	31	163.221.234.DDD

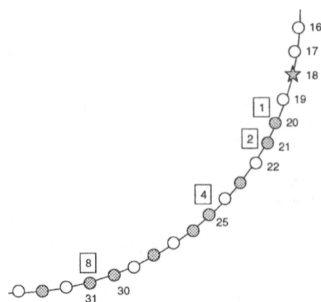


図8 オーバレイネットワークの状態

された1マスを検索する場合、必ずIDが連続することが保証されており、クエリーを1つ送信すれば検索可能となる。

各桁を交互に入れ替える手法では検索回数がID空間の大きさに影響を受けない。しかし座標の連結手法では検索回数がID空間の大きさに強く影響を受ける。このため提案手法は正方検索に適した手法だと言える。これが提案手法で桁の入れ替えによってIDを生成する手法を採用している主たる理由である。

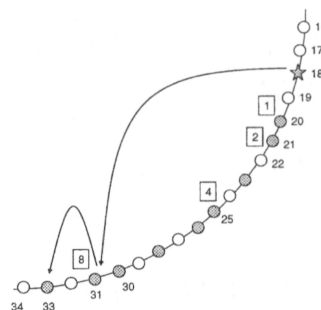


図9 情報の検索

### 3.3.3 検索アルゴリズムの改良

右回りの検索アルゴリズムは非常に単純であり、ノードは右隣の情報を保持するだけでよい。そのために実装が容易という長所があり、ノード数が数十程度ならば良好に動作すると考えられる。しかしノード数が増えるにつれ検索にかかるホップ数が線形に増加し、スケーラビリティに欠ける。そこで円状の双方向スキップリストを利用して検索アルゴリズムの改良を図る。スキップリストのアルゴリズムはDHTで代表的なChordなどで取り入れられており、比較的単純なアルゴリズムで高速な検索が可能となっている。

周りのノードと通信をし、ルーティングテーブルの情報を作成する。ノードはオーバーレイネットワークへ参加した時点で二つ隣りまでのノードの情報を保持。つまりホップ数1,2に関しては最初から情報を保持。二つ隣りのノードに対して、そのノードの二つ隣りのノードの情報を要求する。このことにより、四つ隣りのノードの情報が得られる。次に四つ隣りのノードに対してそのノードの四つ隣りのノードの情報を要求する。そして八つ隣りのノードの情報を得る。要求した情報が得られなかった場合はランダムな時間待ってから再要求する。これを繰り返してルーティングテーブルを更新する。円周で構成されるID空間の半周分を越えるルーティングテーブルの情報は追加しない。つまり自分の場所から半周分を右回りのルーティングテーブルを利用して検索し、もう半分は左回りのルーティングテーブルを用いて検索する。

#### ルーティングアルゴリズム

各ノードは右回りと左回りのルーティングテーブルを持つ。ルーティングテーブルの中身はホップ数、IPアドレス、NodeIDの3つの項目から構成される。表4にNodeID18のノードが持つ右回りのルーティングテーブルの情報を示し、図8にその時のオーバーレイネットワークの状況を図示する。検索元であるNodeID18のノードは星型(★)、オーバーレイネットワークに参加しているノードは黒丸(●)で示す。

ルーティングテーブルを用いることにより右回りにノードをたどる線形検索比べ、ルーティングテーブルを保持するという新たな負荷が生じる。しかしルーティングテーブルの大きさは最大で $\log N$ であ

ホップ数に関しては1,2,4,8,16と二乗の数だけ離れたノードの情報を保持する。各ノードは定期的



表5 動作環境

CPU	Pentium4 2.4GHz
Memory	1GB
OS	WindowsXP SP2
ノード数	10 → 2560
ID 空間 (領域)	$2^{24}$ (4096 × 4096)
ノードの移動アルゴリズム	ランダムウォーク

り、参加ノードが増加してもルーティングテーブルの大きさを抑えることができる。

#### ルーティングテーブルを用いた検索

実際の検索では取得したい情報に関連した ID とルーティング情報の ID を比較して最も近いノードに検索クエリーを送信する。この事を繰り返して目的のノードを探し出す。ここでは例として表4で示したルーティングテーブルを使って NodeID18 のノードが ID34 に関連した情報を検索する様子を図9に示す。

1. ルーティングテーブル (表 4) を参照し、KeyID34 に一番近いノードは NodeID31 の場所に存在すると確認する。そして NodeID31 に検索クエリーを送信する
2. NodeID31 のノードも同様にルーティングテーブルを参照しクエリーを伝播させる
3. KeyID34 に関する情報は NodeID33 のノードの担当領域内にあると分かる

この後は NodeID33 のノードに実際に情報を持っているノードを教えてもらい、そのノードと直接通信することによって、情報を得ることができる。

## 4 シミュレーション評価

提案システムを評価するために Java 2 SDK ver1.4.2-05 上にシミュレータを実装した。OS は WindowsXP-SP2 を用いた。以下に動作環境および評価結果を示す。

### 4.1 動作環境

表5に動作環境とシミュレータの環境設定を示す。シミュレータの利用は提案システムの動作確認およ

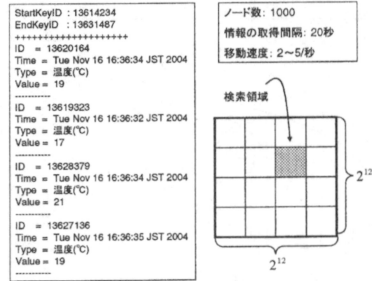


図10 領域検索の検索結果例

びアルゴリズムの特性を測定することを目的としている。そのためにノード数を変化させて検索速度やノード間で交換されるメッセージの数を測定した。計算機の能力のため最大ノード数は 2560 とした。提案システムの特性を算出するには十分な数であった。領域については 4096 × 4096 をとり ID として  $2^{24}$  の空間を割り当てた。これはノード数に対して十分に大きな値であり、ID の衝突などが生じない値となっている。

### 4.2 動作例

ここではノードが移動しながら一定の時間間隔で温度センサによって温度情報を取得し、その情報を位置に基づいて P2P ネットワーク上で流通させるというシナリオを設定した。図10に実際の検索結果例を示す。図中で網掛けをしてある領域を検索した結果、領域内の情報を持ついくつかのノードから温度情報を取得できた。図10の検索結果を見ると全ての ID は StartKeyID と EndKeyID の間にあることが分かる。これは検索対象としている領域内の情報が正確に取得できていることを示している。実際には取得した温度情報を用いて特定の場所の平均気温を算出したり、回りの温度から突出した値を見つけることで車載器などの異常や火災などの発見に利用できる可能性があると考えられる。

### 4.3 検索速度

ランダムにノードを選び任意の領域を検索し情報を得るといった条件の下で、ノードの増加に伴い検索ホップ数がどのように変化するかを計測した。ノード数の増加に対する検索ホップ数の推移は図11

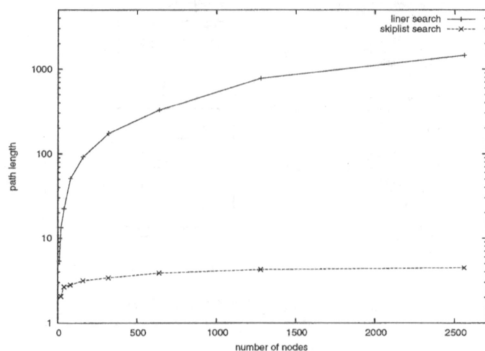


図 11 ノード数と検索ホップ数の関係

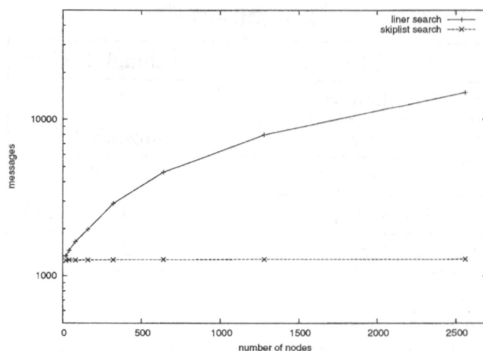


図 12 ノード数とノード一つあたりが処理するメッセージ数の関係

のようになっている。右回りの線形検索ではノードの増加に伴い検索ホップ数が線形に増加している。一方でスキップリストを用いて領域検索した場合は  $O(\log N)$  の検索が可能となっており、ノード数が増加しても高速な検索が維持される。表 2(p.4) に代表的な DHT のアルゴリズムとの比較を示す。DHT は Gnutella や Freenet などのフラッディングを利用するアルゴリズムよりも高速な検索が可能であり、目的とする情報が無い場合は情報が存在しないことが確認できる。一方フラッディングを用いた場合はクエリーの転送を途中で打ち切り一定時間待った後に応答が無い場合は情報が無かったと判断する。つまり本当に情報が無いのかどうかは確認することができない。提案システムは DHT アルゴリズムと同等のオーダーで検索ができ、情報が存在するかの確認もできる (情報が存在しない場合は情報が無いという検索結果を受け取る)。

#### 4.4 ノード間でのメッセージ交換量

ノード間でやり取りされるメッセージの量を測定し、ノード一つあたりが処理するメッセージ数を算出した。メッセージとはノードの生存を定期的に確認するために交換されるメッセージとノードが参加するためであったり離脱したノードが再び参加するために交換されるメッセージの 2 種類から成り立っている。図 12 にシミュレーション結果を示す。線形検索を用いている手法はノードの増加に伴いメッセージ交換量が増加している。生存確認をするためのメッセージ交換量はノード一つ当たり一定であるが、新規参加ノードを参加させるための処理がオーバ

レイネットワークを構成するノード数の増加に伴って線形に増加するためである。一方でスキップリストを利用した場合は新規参加ノードを参加させる処理が少ない。なぜなら、新規参加ノードの NodeID を担当しているノードを  $O(\log N)$  で検索できるためである。このために全体のノード数が増えても一つのノードあたりにかかるメッセージ交換量はほとんど増加していない。サーバクライアント型のサービスではノード数が増えるにつれて線形にメッセージ交換量が増加する。このためにサーバクライアント型に比べて優位性があるといえる。

## 5 今後の課題

検討すべき課題としては以下のものが挙げられる。

- ノードのリンク状態の把握と有効活用
- キャッシュのメカニズム
- モバイルノードに対するプロキシ機能
- 他の P2P との比較
- モバイルアドホックネットワークの利用
- ノードの持つ情報の正当性

### 5.1 ノードのリンク状態の把握と有効活用

現状ではノードのリンク状態は同一という仮定の元にオーバーレイネットワークが構成されている。そのために実際の利用ではネットワーク帯域の狭いところやリンク状態が不安定なところを検索クエリーが流れて、目的の情報を効率的に発見できない事態が

発生すると考えられる。これに対処するためにノードのパラメータに通信メディアの種類(有線, 無線 LAN, PHS など)を追加し, また通信メディアのパケットロス率などを参考にリンクの安定性を判断する。そしてルーティングテーブルを作成する場合にできるだけ状態の良いノードを選択することにより, 実際の検索効率の向上が望める。また取得した情報を登録する際にもメディアの状態を管理して PHS エリア(狭帯域)では情報を貯め続け, 無線 LAN エリア(広帯域)に入ると一気に情報を登録するなどの利用方法も考えられる。

## 5.2 キャッシュのメカニズム

現在の提案手法ではノードがインターネットとの接続を失うとそのノードの持っている情報を取得できなくなる。そのためにキャッシュのメカニズムが必要である。オーバーレイネットワーク上で近くのノードが情報を互いにキャッシュしたり, ネットワークのリンク状態や記憶媒体の量を加味して情報をキャッシュするなどの方法が考えられる。特に無線メディアを使用した場合はインターネットとの接続性が失われる可能性が高い。そのために情報に到達できない事態が発生すると考えられる。しかし上記のように冗長性を確保することにより, 目的の情報を取得する可能性が大幅に向上すると考えられる。

## 5.3 モバイルノードに対するプロキシ機能

ノードの種類はさまざまであり, 特にモバイル機器は性能が乏しいものも少なくない。そのために多くの情報を保持することや多くの検索クエリーを処理することが困難な場合がある。そこでモバイルノードに対してプロキシ機能を提供することが有効であると考えられる。具体的にはモバイルノードは有線で繋がっている自宅のマシンなどに担当領域の管理や検索クエリーの処理などを任せする方法などが考えられる。プロキシ機能を提供することで非常に乏しい計算機資源しか持たないノードでも P2P ネットワークに参加することが可能になる。

## 5.4 他の P2P との比較

現状ではフラッディングや DHT を用いた P2P のシステムをシミュレータ上に実装中である。今後は他の P2P の手法をシミュレータ上に実装し, 同一の環境で比較検討していく必要がある。

## 5.5 モバイルアドホックネットワークの利用

P2P ネットワークにおける課題の一つとして初期ノードをどのようにして発見するかという問題がある。その問題の解決策としてモバイルアドホックネットワーク [10] の利用が有効であると考えられる。オーバーレイネットワークへの参加を希望するノードはアドホックネットワークを利用してオーバーレイネットワークに既に参加しているノードを発見しようと試みる。アドホックネットワークを利用すると今までノードの参加にかかっていた計算量が  $O(\log N)$  から  $O(1)$  に軽減される。そのために高速にオーバーレイネットワークに参加できることに加え, ネットワークに対する負荷も軽減でき, 初期ノードのリストを保持していなかったとしてもオーバーレイネットワーク参加することが可能となる。

## 5.6 ノードの持つ情報の正当性

ノードは情報を取得すると, 情報を取得した位置と関連付けてオーバーレイネットワークに情報を登録する。この場合, どのノードが登録した情報も正しいものとして管理されている。そのために悪意のあるノードが偽ったデータを登録することが可能となる。そこで認証局などを用いて事前にオーバーレイネットワークに参加するノードに対して認証を行い, その後で情報の検索および登録を可能にするという方法が考えられる。悪意のあるノードが発見された場合はオーバーレイネットワークから強制的に切断し, もう一度参加を試みた場合は認証の段階で参加を拒否するなどの方法が考えられる。

## 6 おわりに

位置情報サービスへの要求の高まりや移動ノードの増加を受けて, 位置に関連した情報の管理・検索手

法を提案した。本提案のモデルを利用することにより、位置に基づいた情報検索と位置に基づいた情報の蓄積が可能となる。そのため、現在サーバ・クライアント型で行っている統計情報のサービスをP2Pネットワークで行うことができ、負荷分散の効果があると考えられる。また2次元を1次元の円周にマッピングし、一意な検索を可能としているので、フラッディングを用いる他のP2Pシステムと比べてネットワークに対する負荷が少なく、 $O(\log N)$ での検索が可能で、情報の有無を確認できるといった優位性がある。

シミュレーションの結果から、提案システムは既存のDHTと同等の計算量で検索が可能である事が明らかとなった。また検索にかかる計算量やノード一つ当たりの負荷が全体のノード数 $N$ に対して $\log N$ であることから、ノード数の増加に強いシステムであることが明らかとなった。

今後の課題としては実環境に近いエミュレート環境を構築したり、実機を使った実環境での測定を行う必要がある。また実環境に適応するためにネットワークや計算機資源を加味し、システムを改良する。

位置に基づいた情報の蓄積を行うことで、情報の2次利用を通してより集約化が進み、個々のノードの持っている情報から多様な集約情報が生成されると考えられる。

## 参考文献

- [1] 日本自動車研究所 (JARI: Japan Automobile Research Institute)  
(<http://www.jari.or.jp/>)
- [2] Napster(<http://www.napster.com/>)
- [3] Gnutella(<http://www.gnutella.com/>)
- [4] Freenet(<http://freenetproject.org/>)
- [5] S. Ion, M. Robert, K. David, K. Frans M and B. Hari: "Chord: A Scalable Peer-to-Peer Lockup Service for Internet Applications." ACM of the 2001 conference on Applications, technologies, architectures, and protocols for computer communications, pp. 149-160, 2001.
- [6] R. Sylvania, F. Paul, H. Mark, K. Richard and S. Scott: "A Scalable Content-Addressable Network," ACM SIGCOMM,

pp. 161-172, 2001.

- [7] B. Zhou, D. A. Joseph, J. Kubiatowicz, "Tapestry: a fault tolerant wide area network infrastructure," Sigcomm 2001 poster and UC Berkeley Tech. Report UCB/CSD-01-1141.
- [8] A. Rowstron and P. Druschel, "Pastry: Scalable, Decentralized Object Location and Routing for Large-Scale Peer-to-Peer Systems," IFIP/ACM Int'l Conf. Distributed Systems Platforms (Middleware), ACM Press, 2001. pp. 329-350
- [9] 金子雄, 福村真哉, 春本要, 下條真司, 西尾章治郎, "モバイル環境における端末の位置情報に基づくP2Pネットワークの提案と評価", 電子情報通信学会第15回データ工学ワークショップ (DEWS2004) 論文集, March 2004.
- [10] Mobile Ad-hoc Network  
(<http://www.ietf.org/html.charters/manet-charter.html>)

本 PDF ファイルは 2005 年発行の「第 46 回プログラミング・シンポジウム報告集」をスキャンし、項目ごとに整理して、情報処理学会電子図書館「情報学広場」に掲載するものです。

この出版物は情報処理学会への著作権譲渡がなされていませんが、情報処理学会公式 Web サイトに、下記「過去のプログラミング・シンポジウム報告集の利用許諾について」を掲載し、権利者の検索をおこないました。そのうえで同意をいただいたもの、お申し出のなかったものを掲載しています。

[https://www.ipsj.or.jp/topics/Past\\_reports.html](https://www.ipsj.or.jp/topics/Past_reports.html)

#### 過去のプログラミング・シンポジウム報告集の利用許諾について

情報処理学会発行の出版物著作権は平成 12 年から情報処理学会著作権規程に従い、学会に帰属することになっています。

プログラミング・シンポジウムの報告集は、情報処理学会と設立の事情が異なるため、この改訂がシンポジウム内部で徹底しておらず、情報処理学会の他の出版物が情報学広場 (=情報処理学会電子図書館) で公開されているにも拘らず、古い報告集には公開されていないものが少からずありました。

プログラミング・シンポジウムは昭和 59 年に情報処理学会の一部門になりましたが、それ以前の報告集も含め、この度学会の他の出版物と同様の扱いにしたいと考えます。過去のすべての報告集の論文について、著作権者 (論文を執筆された故人の相続人) を探し出して利用許諾に関する同意を頂くことは困難ですので、一定期間の権利者搜索の努力をしたうえで、著作権者が見つからない場合も論文を情報学広場に掲載させていただきたいと思います。その後、著作権者が発見され、情報学広場への掲載の継続に同意が得られなかった場合には、当該論文については、掲載を停止致します。

この措置にご意見のある方は、プログラミング・シンポジウムの辻尚史運営委員長 ([tsuji@math.s.chiba-u.ac.jp](mailto:tsuji@math.s.chiba-u.ac.jp)) までお申し出ください。

加えて、著作権者について情報をお持ちの方は事務局まで情報をお寄せくださいますようお願い申し上げます。

期間：2020 年 12 月 18 日～2021 年 3 月 19 日

掲載日：2020 年 12 月 18 日

プログラミング・シンポジウム委員会

情報処理学会著作権規程

<https://www.ipsj.or.jp/copyright/ronbun/copyright.html>