

モバイルP2Pにおけるレーン構造を用いた資源探索手法

小林正幸, 呉エキ, 大下福仁, 角川裕次, 増澤利光
大阪大学大学院情報科学研究科 コンピュータサイエンス専攻

モバイルアドホックネットワーク上でのP2PシステムであるモバイルP2P(MP2P)が注目されている。MP2Pにおける効率的な資源探索手法として、GPSによる地理情報を用いた手法が注目されている。本稿では通信コストの削減のため、動的なレーン構造を利用した資源探索手法を提案する。提案手法は、座標軸すなわち x 軸と y 軸をもつフィールドにおいて、資源を保持するノードが x 軸と平行に資源情報を散布し、資源情報を所持する端末でレーンを構成する。これにより、資源を探索する際は、探索メッセージを y 軸と平行に転送することで、必ずレーンに到達し、効率的な探索が可能である。さらに、相対速度を考慮してレーンを構築するため、動的な環境においても通信コストの抑制が可能である。また、シミュレーション実験を行い、既存手法との比較により提案手法の有効性を示す。

Object search using lane structure in mobile P2P systems

Masayuki Kobayashi, Yu Wu, Fukuhiro Ooshita,
Hirotetsu Kakugawa, Toshimitsu Masuzawa

Graduate School of Information Science and Technology, Osaka University

A mobile P2P system(MP2P) is a P2P system over a mobile ad hoc network. In MP2P, object search method using GPS is popular. In this paper, we propose a new object search method using lane structure. We allocate object information to nodes in parallel with x -axis. When we request an object, object search message is transferred in parallel with y -axis. In this way, we certainly discovery object information with small search cost. Additionally, object information is allocated to nodes which have about the same velocity. In this way, communication cost will be small. Simulation results show that our algorithm is effective.

1 はじめに

Peer-to-Peer システム(以降 P2P システム)とは、ネットワーク内の端末(ノード)がサーバを介さず直接通信するネットワークである。P2P システムは負荷分散、システムの拡張性、故障耐性といった優れた特性を有しており、ファイル共有システムやIP 電話、分散計算など様々な分野に応用されている。P2P システムにおいてファイルなどの資源は膨大な数のノードに分散して配置されている。そのため、ファイル共有などのアプリケーションでは、目的の資源を持つノードを探し出す必要がある。この問題は**探索問題**と呼ばれ、P2P システムにおける最も基本的な問題のひとつである。

近年、無線通信技術の急速な発展やPDA、携帯電話といった小型かつ高性能な計算機の普及にともない、モバイルアドホックネットワークが広く利用されるようになった。モバイルアドホックネットワークとは、有線リンクや基地局を持たず、移動ノードのみで構成される無線ネットワークである。モバイルアドホックネットワークの普及に伴い、モバイルアドホックネットワーク上でのP2Pシステムすなわち**モバイルP2P**(以降、MP2P)が注目されている。MP2Pにおいて探索問題を解決する場合、有線ネットワークにおけるP2Pシステムで有効であった手法は必ずしも有効ではない。これは、P2PシステムがTCP/IP上のオーバーレイ構造であるのに対し、MP2Pは、ノード間における通信の可否は物理位置に依存し、ノードの移動によりトポロジが動的に変

化するため、オーバーレイ構造を保つために必要なメッセージが非常に多くなることが原因である。また、モバイルノードの記憶領域は小さいため、多くのノードや資源の情報を扱うことはできない。したがって、MP2Pに適した探索手法が必要である。

MP2Pでの探索問題を解決する手法として、送信可能な全ノードに対して、資源を探索するためのメッセージであるクエリを転送し続けるフラッディングを基盤とした手法がある。しかし、探索に要するメッセージ数(以降、探索コスト)が探索範囲の増加に対して大幅に増加するため、大規模ネットワークには向かないという問題がある。そこで、資源を所持するノード(以降、ソースノード)の地理座標を含む情報である**インデックス**を用いた探索手法GHT[1]が提案されている。GHTでは、GPS(Global Positioning System)によりインデックスを特定の地理座標に配置し、該当座標付近のノードがインデックスを管理する。インデックスを管理するノード(以降、インデックスノード)にクエリが到達すると、インデックスを元にクエリはソースノードへ転送される。GHTは、特定の地理座標へのみクエリを送信し、その地理座標付近に存在するインデックスノードがソースノードへクエリを転送するという仕組みであるため、資源探索コストの抑制を実現している。しかし、インデックスを配置する地理座標を固定するため、配置する地理座標付近にノードがない場合、インデックスを配置する領域を広げるもしくは別の地理座標に配置する必要がある。また、ノードはフィールド

上を自由に移動するため、インデックスノードの交代処理やインデックスの更新処理も頻発する。このため、GHTには構造を維持し続けるために必要なメッセージ数(以降、維持コスト)が多いという問題がある。

本研究では、資源探索成功率が高く探索・維持コストが低い手法を提案する。想定するフィールドは長方形であり、座標軸すなわち x 軸と y 軸をもつ。提案手法では、ソースノードからフィールド端まで x 軸と平行にインデックスを配置する。これをレーン構造と呼ぶ。そして、クエリを y 軸と平行に順次転送する。クエリはインデックスノードに到達後、ソースノードへ向けてレーン上を転送される。二次元平面であるフィールドに対し、 x 軸方向にのみインデックスを散布するため、インデックスノードは少ない。 x 軸方向のインデックス散布がフィールド両端まで到達している状況では、二次元平面であるフィールドに対し、提案手法では、 x, y 軸方向に直線状にクエリを転送するだけで資源探索が可能であるため、探索コストの抑制が期待できる。また、提案手法では、インデックスを散布する際、相対的な位置と速度に着目し、相対速度の小さいノードによりレーンが構成されるように、散布先のノードを選択する。これにより、インデックスノードの交代処理やインデックスの更新処理の頻度を抑制することができ、維持コストの抑制が期待できる。

2 諸定義

2.1 システムモデル

MP2Pでは、ノード集合 $V = \{p_1, p_2, \dots, p_n\}$ で構成される。各ノードは固有の識別子を持つ。各ノードは二次元平面上のフィールドに配置されており、フィールド上を移動する。各ノードは、通信範囲内に存在する他のノードとのみ、無線通信によるメッセージの送受信が可能である。本稿では、各ノードの通信範囲は、そのノードを中心とする半径 R の円とする。以下、 R を通信半径と呼ぶ。ノード p_i, p_j が互いに無線通信可能であるとき、 p_i, p_j は隣接するという。本研究では、ノード p_i が隣接ノード p_j に送信したメッセージは必ずノード p_j が受信すると仮定する。つまり、通信の衝突によるメッセージの消失、およびメッセージの改変は起こらないとする。

本稿で対象とするフィールドは $X \times Y$ の長方形とし、座標軸すなわち x 軸と y 軸を持ち、四辺のうち二辺は $y = 0 (0 \leq x \leq X)$, $x = 0 (0 \leq y \leq Y)$ である(図1)。各ノードはフィールド上でGPSを用いることにより、自身の地理座標と速度情報を把握している。ノード p_i の地理座標を $(x_i, y_i) (0 \leq x_i \leq X, 0 \leq y_i \leq Y)$, 速度を v_i と表記する。各ノードは自身と隣接するノードの座標や速度情報を把握し、ネットワーク中の平均速度の大きさ $|v|$ と最低速度の大きさ $|v_{min}|$ は与えられていると仮定する。フィールド上において、 $x = 0$ を x 軸の負方向のフィールド端、 $x = X$ を x 軸の正方向のフィールド端、 $y = 0$

を y 軸の負方向のフィールド端、 $y = Y$ を y 軸の正方向のフィールド端とする。

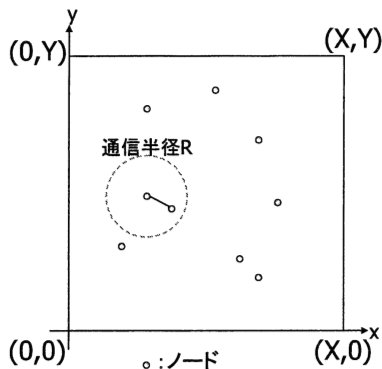


図1: フィールド

2.2 資源探索問題

MP2Pでは、ファイルなどの資源がネットワーク中のノードに分散して配置されているため、資源を要求するノードはMP2P中から、その資源を持つノードを探索しなければならない。本稿では、資源を要求するノードをリクエストノード、該当する資源を所持するノードをソースノードと呼ぶ。ソースノードは、リクエストノードが資源を発見しやすくなるように、位置や速度などの情報をインデックスとして複数のノードに配置しておく。インデックスを所持するノードをインデックスノードと呼ぶ。リクエストノードは、クエリと呼ばれるメッセージを発信することで資源の探索を開始する。本稿では、クエリがソースノードに到達することを資源探索が成功したという。

資源探索アルゴリズムの評価指標として、探索成功率、探索時に要したメッセージ数である探索コスト、インデックスの更新・削除処理に要したメッセージ数である維持コストが挙げられる。GPSを用いたMP2Pにおける維持コストには、更新コストと交代コストがある。更新コストとは、資源を所持するノードが移動した際、その資源に対するすべてのインデックスの更新に必要なメッセージ数である。交代コストとは、インデックスを管理するノードが移動した際、管理するノードの交代もしくは追加に必要なメッセージ数である。

3 提案手法

3.1 概要

ソースノードは、インデックスノードを少なくするために、図2のように、ソースノードから x 軸の正負両方向のフィールド端までできるだけ直線的にインデックスノードを選出する。ソースノード、そのインデックスノードをまとめてレーンと呼ぶ。リ

クエストノードはクエリの転送回数を抑えるために、クエリを y 軸の正負両方向に発信する。 x 軸の負方向のフィールド端から正方向のフィールド端までレーンを構築した場合、クエリは必ずレーンに到達（ヒット）する。レーンにヒットしたクエリはレーン上をソースノード方向へ転送され、最終的にソースノードに到達し、資源探索は成功する。また、維持コストを抑制するため、インデックスノードを選出する際、相対的な位置や速度を考慮し、ノードを選出する。以下、提案手法の詳細を述べる。

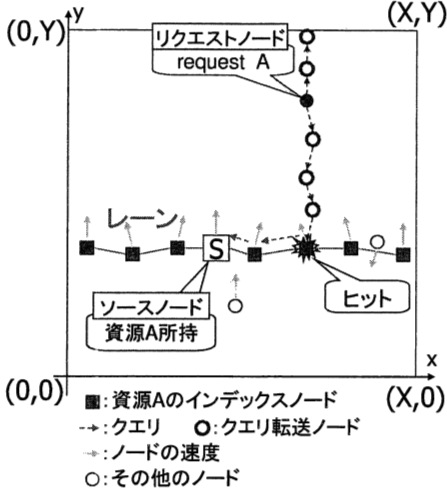


図 2: レーン構造

3.2 レーンの構築・維持手法

ソースノード p_S が構築したレーン上の全ノード集合を L_S とし、 x 軸の正方向に構築したレーン上の全ノード集合を L_S^+ 、負方向に構築したレーン上の全ノード集合を L_S^- とする。つまり、図 3 のようにレーンが構築されたとき、 $L_S^+ = \{p_A, p_B, p_C\}$ 、 $L_S^- = \{p_D, p_E, p_F\}$ 、 $L_S = \{p_S, L_S^+, L_S^-\}$ である。

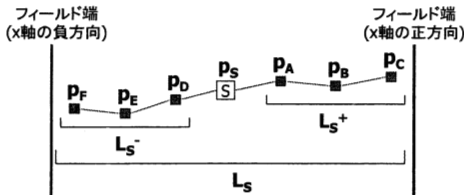


図 3: L_S , L_S^+ , L_S^- の関係

3.3 レーンの構築手法

レーン末端のインデックスノードは、ソースノードとレーン上のインデックスノードを除く隣接ノ

ドの中からインデックスノードを選出する。レーン上のあるインデックスノード p_i の位置 (x_i, y_i) が $0 \leq x_i \leq \sqrt{3}R/2$ 、 $0 \leq y_i \leq Y$ を満たすとき、このレーンは x 軸の負方向のフィールド端に到達したという。あるインデックスノード p_j の位置 (x_j, y_j) が $X - \sqrt{3}R/2 \leq x_j \leq X$ 、 $0 \leq y_j \leq Y$ を満たすとき、このレーンは x 軸の正方向のフィールド端に到達したという。インデックスノードの選出処理を、フィールド端に到達する、もしくは、条件に合致する隣接ノードが存在しなくなるまで繰り返し行うことにより、レーンを構築する。 x 軸の正負両方向においてフィールド端にレーンが到達している場合、このレーンはフィールドを横断しているという。

インデックスノード選出の際には、評価式を用いて隣接ノードの中から最も評価値が高いノードをインデックスノードに選出する。レーン末端のインデックスノード p_A が隣接ノード p_C を評価する状況を考える（図 4）。 p_A を中心とした半径 R の円を C_A とし、ノード p_C の p_A に対する相対速度の延長線上にある円 C_A との交点を D とする。 D と p_C 間の距離を $rdist_{AC}$ とする。

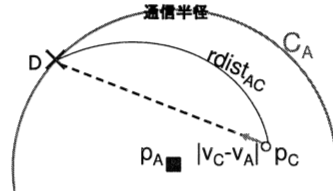


図 4: 評価モデル (p_A が p_C を評価)

提案手法では、インデックスノードを選出する際、相対的な位置と速度を考慮してノードを評価する。

相対的な位置を考慮する際、探索コスト抑制のため、レーンを構成するインデックスノードの数は少ないことが望ましい。したがって、相対的な位置に関する評価指標 $E_d(p_A, p_C)$ は以下の式である。

$$E_d(p_A, p_C) = \begin{cases} x_C - x_A & (\text{正方向にレーン構築}) \\ x_A - x_C & (\text{負方向にレーン構築}) \end{cases} \quad (1)$$

インデックスノード p_A に対し、 p_C がフィールド端に近い位置に存在するほど、 $E_d(p_A, p_C)$ は大きい。

相対速度を考慮する際、インデックスノードの交代コスト抑制のため、選出されるインデックスノードは隣接インデックスノードの通信半径内に長く留まることが望ましい。したがって、相対速度に関する評価指標 $E_v(p_A, p_C)$ は以下の式である。

$$E_v(p_A, p_C) = \frac{|\vec{v}| \cdot rdist_{AC}}{|v_C - v_A| + |v_{min}|} \quad (2)$$

相対的な位置と相対速度を考慮し、それぞれの評価指標に重み α ($0 < \alpha \leq 1$) を加えた以下の式

$E(p_A, p_C)$ をインデックスノード選出の評価式として用いる。

$$E(p_A, p_C) = \alpha E_d(p_A, p_C) + (1 - \alpha) E_v(p_A, p_C) \quad (3)$$

3.4 インデックスノードの追加及びレーンの再構築手法

各ノードはフィールド上で自由に移動するため、レーン構築後、何も処理をしなければレーンは切断されてしまう。したがって、レーン上のインデックスノードの追加処理もしくは交代処理によりレーンを維持する必要がある。そこで、本節よりレーン維持手法について述べる。

あるレーン上で隣接関係にあるインデックスノード p_A, p_B (p_A は p_B よりもソースノード側) 間のリンクが切れることでレーンが切断される状況を考える。このとき、 p_B よりもソースノード側である p_A はレーンの切断が発生することを防止するために、レーン維持処理を行う。レーン維持処理には、状況によって2通りの手法がある。1つ目は、 p_A, p_B の通信範囲内の重複部分にノードが存在する状況である。2つ目は、該当部分にノードが存在しない状況である。

まず、 p_A, p_B の通信範囲内の重複部分にノードが存在する状況について記す。この状況では、重複部分に長く留まるノードをインデックスノードとして選出し、レーンに追加する。選出されたインデックスノードを p_C とすると、 p_A, p_C 間と p_C, p_B 間において、レーン上での隣接関係を構築する。

このインデックスノード追加手法について詳細を述べる。 p_A, p_B の両方と隣接関係にあるノードを候補ノードと呼ぶ。各候補ノード p_C に対し、図5のように、 p_C が p_A, p_B の通信半径内に留まる時間を算出する。算出された2つの時間のうち小さい方の値を p_C の評価値とする。すなわち、評価値 $Value(p_C)$ は以下の式で表せる。

$$Value(p_C) = \min\left(\frac{rdist_{AC}}{|v_C - v_A|}, \frac{rdist_{BC}}{|v_C - v_B|}\right) \quad (4)$$

候補ノードの中で最も評価値の高いノードをインデックスノードとして選出する。

次に、 p_A, p_B の通信半径内の重複部分にノードが存在しない状況について記す。このとき、 p_A からフィールドの端に向けて、評価式 (3) を用いてインデックスノードの選出処理を順次行う。この処理をレーン再構築と呼ぶ。

レーン再構築の過程において、再構築の際に選出されたインデックスノード p_D が既存レーン上のインデックスノード p_E の通信半径内に存在した場合、レーン再構築処理を終了し、 p_D, p_E 間でレーン上での隣接関係を構築する。そして、 p_B は、 p_A がレーンを再構築する際に発信するメッセージを受信してから一定時間後に削除メッセージを送信し、 p_B と p_B, p_E 間にあるインデックスノードをレーンから削

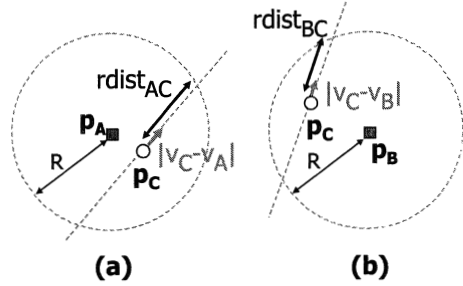


図5: 候補ノード p_C の評価モデル

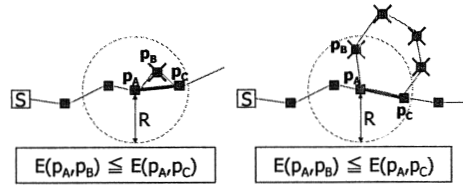


図6: 不要インデックスノード削除例

除する。この処理は、 p_B から p_E までレーンに沿って削除メッセージを転送することで行う。既存レーンと合流しなかった場合も同様に、 p_B 以降のフィールド端まで展開しているインデックスノードをレーンから削除する。

3.5 レーンの冗長部分削除手法

インデックスノードが移動してもインデックスノードの追加によりレーン構造は保たれるが、追加手法を繰り返すことによりインデックスノードが増加してしまう。インデックスノードの移動により、レーン上にリング状の経路が構築された場合、レーンの冗長部分の削除を行う。

各インデックスノード p_A は、自身の通信半径内に $p_B, p_C \in L_A^+$ または $p_B, p_C \in L_A^-$ を満たす p_B, p_C を発見した場合、評価式 (3) を用いて p_B, p_C を評価し、評価値が高いノードを隣接インデックスノードとする。図6のように、インデックスノード p_A に対し、現在のレーンで隣接しているインデックスノード p_B が p_B, p_C の中で最大の評価値を持たない場合、 p_A, p_C 間でレーン上での隣接関係を構築し、 p_B と p_B, p_C 間のインデックスノードをレーンから削除する。

3.6 探索手法

リクエストノードは探索を行う際、 y 軸の正負両方向にクエリを送信する。 y 軸方向のクエリ転送は、転送ノードに対し y 軸方向に通信半径内で最も遠いノードが順次行う。図7において、リクエストノード p_R がクエリを発信した場合、隣接ノードのうち y 軸の正方向に最も遠いノードは p_B 、負方向に最も遠いノードは p_C であるため、 p_B, p_C がそれぞれの方向へクエリを転送する。転送処理は、クエリが探

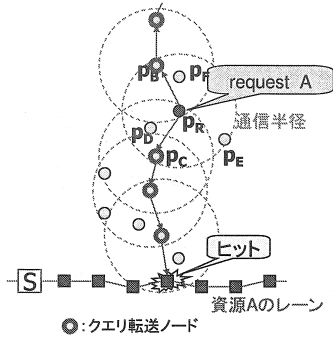


図 7: y 軸方向のクエリ転送

索資源のインデックスを所持するレーンに到達する、もしくは、 y 軸方向のフィールド端に到達するまで繰り返される。レーンにヒットしたクエリはレーン上のインデックスノードによりソースノードまで転送され、資源探索は成功する。

y 軸方向のクエリ転送において、通信半径内の全ノードはクエリを受信するため、レーンがフィールドを横断している場合、 y 軸方向のクエリ転送を失敗しない限り、クエリは必ずレーンにヒットする。

4 シミュレーション実験

本章では、提案手法の性能を示すためにシミュレーション実験を行う。また、地理情報を用いた GHT を MP2P 向けに改良した MPI(Multi-level Peer Index)[2] と提案手法を比較評価し、提案手法の有効性を示す。

4.1 評価環境

シミュレーション実験は、 $1600\text{m} \times 1600\text{m}$ のフィールド、ノードの通信半径は $R = 280\text{m}$ とした。各ノードの初期位置はフィールド上の点を一様ランダムに選択して決定した。各ノードはユニークな資源を 1 つ所持する。シミュレーション実験で、各ノードはラウンドにより動作する。シミュレーション時間を 300 ラウンドとし、ノードの移動モデルにはランダムウェイポイントモデルを用いる。ランダムウェイポイントモデルでは、各ノードがランダムに目的地と速度を決定し、目的地まで一定速度で移動し、目的地に到着後、一定時間停止という動作を繰り返す。本シミュレーションでは、速度を $1 \sim 10\text{m}$ /ラウンド、 $1 \sim 3\text{m}$ /ラウンド、 $8 \sim 10\text{m}$ /ラウンドの 3 種類で実験し、停止時間は 0 ラウンドとした。また、ノード数 100, 200, 400、提案手法の α の値は、0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0 と設定し、それぞれの環境において 10 回ずつ実験を行った。

MPI はフィールドを一辺 $\sqrt{2R}$ の正方形であるセルでグリッド状に分割する。同じセルに属する任意の 2 ノードは直通信可能である。MPI では、各資源に第 1 ハッシュ関数を適応してセルに対応させる。ソースノードは GPSR を用いてインデックスを第

1 ハッシュ関数が出力したセルへ転送する。セル内でインデックスを管理することにより、ノードの移動による維持コストを抑制する。リクエストノードは、同じ第 1 ハッシュ関数を用いて、第 1 ハッシュ関数が出力したセルへクエリを転送し、探索する資源のインデックスを取得する。また、MPI では近隣の複数のセルによって、正方形のグループを構成し、仮想的階層構造を構築することで、効率的な探索を行っている。ソースノードによるインデックス配置、リクエストノードによる探索ともに第 1 ハッシュ関数が出力したセルにノードが存在しない場合、第 2 ハッシュ関数を用いて同様の処理を行う。

本シミュレーションでは、フィールド $1600\text{m} \times 1600\text{m}$ 、 $R = 280\text{m}$ であるため、フィールドは 64 個のセルによって構成されている。また、第 2 ハッシュ関数の実装方法が定義されていないため、第 2 ハッシュ関数を用いることによる情報配置・インデックス更新の際に発生するメッセージを維持コストに換算せずに評価する。

4.2 評価基準

シミュレーション実験では探索成功率、探索コスト、維持コストについて提案手法と MPI を比較評価する。探索成功率とは、全クエリのうち資源探索に成功したクエリの割合である。探索コストとは、リクエストノードから発信されたクエリのうち、資源探索に成功したクエリがソースノードに到達するまでに要したメッセージ数の平均である。維持コストとは、1 ラウンド当たりのインデックスの更新・削除処理に要した総メッセージ数である。

4.3 評価結果

探索成功率 図 8 は MPI と提案手法の資源探索成功率である。横軸は α 、縦軸は資源探索成功率である。レーン (100) とはノード数 100 における提案手法のグラフである。両手法とも、ノード数が 100 である場合、探索成功率は低い。これはノードが少ない場合、MPI ではノードが存在しないセルが多く、提案手法では隣接ノードの減少によりフィールドを横断できないレーンが多くなるためである。ノード数が 200 以上では、両手法とも 100% に近い値で資源探索は成功する。

探索コスト 図 9 は MPI と提案手法の探索コストである。横軸は α 、縦軸は探索コストである。提案手法は MPI よりも探索コストが低いことが分かる。これは、MPI がハッシュ関数を用いてインデックス配置セルを固定するため、リクエストノードとソースノードとインデックスを配置するセルの位置関係によっては、冗長な探索経路を構築するためである。つまり、リクエストノードからソースノードに対して遠くなる方向であろうとも、まずはインデックス配置セルへクエリを転送し、そのセルからソースノードへクエリを転送するという処理が行われているためである。一方、提案手法は x, y 軸方向のみの直線

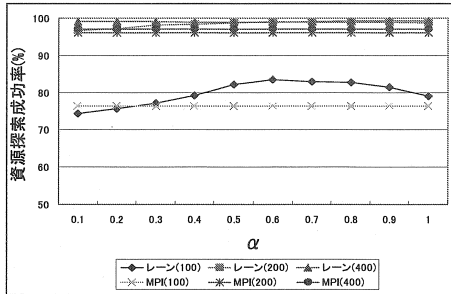


図 8: 資源探索成功率

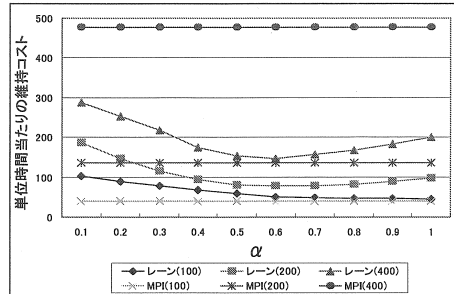


図 10: 維持コスト

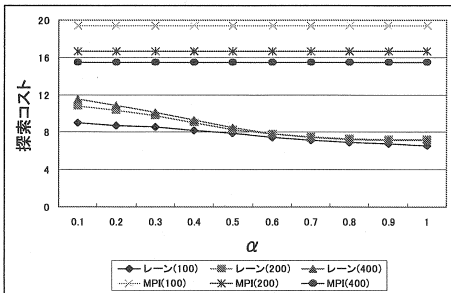


図 9: 探索コスト

動的な探索経路を構築するため、効率的な探索である。

MPI(100),MPI(200),MPI(400)を比較した場合、MPIはノードが少ないとき、ノードが多いときよりも探索コストは大きい。これは、ノードが少ないとき、第1ハッシュ関数で指定されたセルにノードが存在しない状況が頻発し、第2ハッシュ関数の使用頻度が高くなるためである。このとき、クエリは第1ハッシュ関数で指定されたセルから第2ハッシュ関数で指定されたセルへ転送されるため探索コストは大きくなる。

提案手法は α が小さくなるに従って、探索コストが大きくなる。これは α が小さいときはレーンが直線形状ではない可能性が上がるため、レーン上のインデックスノードが多く、レーン上のクエリ転送処理に多くのメッセージが必要であるためである。

維持コスト 図10はMPI,提案手法の単位時間当たりの維持コストである。両手法はノード数の増加に伴い、維持コストが大きくなる。ノード数200と400において、提案手法とMPIを比較した場合、提案手法の方が維持コストは小さい。ノード数100では、MPIの方が維持コストは小さい。しかし、ノード数100のとき、資源探索成功率は低いため、ノードが存在しないセルが多いことが考えられる、すなわち、第2ハッシュ関数の利用頻度が高いため、実際の維持コストは結果よりも高い。MPIはセルという固定された領域でインデックスを管理しているため、インデックスノードとソースノードの移動により、インデックスノードの交代とインデックスの更新

が頻発する。一方、提案手法は相対的な位置と速度に着目した直線状のレーン内でインデックスを管理しており、更新コストや交代コストを抑制している。

提案手法では、 α が小さいとき、レーン長は長くなりレーンを構成するインデックスノードの数も増加するため維持コストは大きくなる。 α が大きいつき、隣接インデックスノードとの相対速度が大きく通信半径に近い位置に存在するインデックスノードが選出される可能性が増加する。このとき、レーン長は短いインデックスノード間のリンクは切断しやすく、維持コストは大きくなる。

5 まとめ

本稿では通信コストの削減のため、動的なレーン構造を利用した資源探索手法を提案した。シミュレーション結果から、提案手法手法が既存手法に比べ、探索コスト・維持コストともに小さいことを示した。

謝辞 本研究の一部は、文部科学省グローバルCOEプログラム(研究拠点形成費)の研究助成、日本学術振興会科学研究費補助金(基盤研究(B)19300017,若手研究(B)18700059)、文部科学省科学研究費補助金(特定領域研究16092215)、総務省戦略的情報通信研究開発推進制度(SCOPE)、(財)栢森情報科学振興財団、中島記念国際交流財団によるものである。ここに記して謝意を表す。

参考文献

- [1] S.Ratnasamy, B.Karp, S.Shenker, D.Estrin, R.Govindan, and F.Yu. Data-Centric Storage in Sensornets with GHT, a Geographic Hash Table. In Proceedings of the Mobile Networks and Applications(MONET), 8(4):pp.427-442,2003.
- [2] M.Li, W-C.Lee, and A.Sivasubramaniam. Efficient Peer-to-Peer Information Sharing over Mobile Ad Hoc Networks. In Proceedings of MobEAI, 2004