

P2Pシステムにおけるブルームフィルタを利用した オーバーレイネットワークの構築

高橋佑典[†], 泉泰介^{††}, 角川裕次[†], 増澤利光[†]

[†] 大阪大学大学院 情報科学研究科

{yusuke-t, kakugawa, masuzawa}@ist.osaka-u.ac.jp

^{††} 名古屋工業大学大学院 工学研究科

t-izumi@nitech.ac.jp

ブルームフィルタは、誤検出を許容することにより、データアイテム集合のインデックス情報を少ないメモリ領域で表現できる手段の一つである。そのため、ブルームフィルタを用いることで、P2P ネットワーク上での資源検索を効率的な手法を実現する手法が提案されている。一般的に、ブルームフィルタを用いた既存の検索手法は任意のトポロジ上で適用可能であるため、ネットワークのトポロジ構造に一切の仮定をおかないモデル上で考察される場合が多い。しかし、ある特定のトポロジ構造に対してブルームフィルタを利用した検索が非常に有効に働き、かつそのような検索に適したトポロジを少ないコストで構築できれば、さらに効率的な検索が実現できると考えられる。そこで本研究では、ブルームフィルタを利用した検索に適したオーバーレイネットワークを構築する手法、および、構築したネットワークを利用した場合転送手法を提案する。また、提案手法を用いることで、既存手法よりも効率的な検索が実現できること、および、少ないコストで構築したトポロジ上でも十分効率的な検索が実現できることをシミュレーション実験によって示す。

Peer-to-peer Overlay Construction Based on Bloom Filter

Yusuke TAKAHASHI[†], Taisuke IZUMI^{††}, Hirotugu KAKUGAWA[†], Toshimitsu MASUZAWA[†]

[†] Graduate School of Information Science and Technology, Osaka University

{yusuke-t, kakugawa, masuzawa}@ist.osaka-u.ac.jp

^{††} Graduate School of Engineering, Nagoya Institute of Technology

t-izumi@nitech.ac.jp

The use of Bloom filters is one of the most popular approach to improve lookup efficiency in P2P networks. A bloom filter is a representation of data item indices, which achieves small memory requirement by allowing one-sided errors (false positive). Usually, Bloom-filter-based lookup methods assume nothing about overlay structure because their primary advantage is that they can be adopted on any overlay topology. However, it can be expected that the efficiency of bloom-filter-based methods can be improved only if we slightly control the topological structure of overlay networks. Thus, in our research, we propose an overlay construction strategy that is appropriate to Bloom-filter-based lookup methods and a query routing method on it. We also show by simulation that our method can achieve better lookup performance than existing ones and we realize sufficiently efficient lookup on a topology constructed with low cost.

1 まえがき

ピア・ツー・ピアネットワーク (以下 P2P ネットワーク) とは、ネットワーク内の計算機がサーバを介さずに直接通信する分散システムである。P2P ネットワークにおいて、テキストファイルやプログラムといったデータ (データアイテム) は多数のピアに分散して配置されるため、ファイル共有などのアプリケーションを動作させる場合、目的となるデータアイテムを持つピアを探し出す事が必要となる。この問題は検索問題と呼ばれており P2P ネットワークにおける重要な基本問題の一つとして考えられている。検索問題の解法を効率化する手法のひとつとして、ブルームフィルタを使用した検索手法が提案されている [2]。ブルームフィルタは自分の所持するデータアイテムの集合の要約であり、多少の誤りを許容することで、所持するデータアイテム集合をコンパクトなインデッ

クス情報として表現することができる。ブルームフィルタを利用する検索において、各ピアはあらかじめ、自分が所持するデータアイテムの集合に対するブルームフィルタを作成し、近隣ピアに散布する。あるデータアイテムの検索を行いたい場合は、そのデータアイテムに対応する問合せを作成し、この問合せを目的のデータアイテムを所持するピアまで転送する。このとき、あらかじめ配布しておいたブルームフィルタを手がかりにすることで、問合せを短い経路で目的のデータアイテムを所持するピアに到達させることができる。ただし、ブルームフィルタは誤りを含むため、問い合わせが目的のピアへと近付かない方向へと転送される可能性がある。一般的に、この誤り確率は、要約するデータアイテムの個数がフィルタサイズに対して大きくなるにつれて高くなる。

ブルームフィルタを利用した従来の検索手法において、問合せが目的となるデータアイテムの情

報を持つブルームフィルタを発見した場合、そのフィルタを手掛かりにすることで、速やかに目的のアイテムをもつピアまで転送することができる。ただし、あらかじめ配布しておくブルームフィルタは一定の領域にしか散布されないため、問合せを目的ピアまで転送するには、まずは、ブルームフィルタが散布されている領域まで到達させる必要がある。このとき、従来の検索手法では、フィルタが散布された領域がどこに存在するかがわからないため、ランダムウォーク等でこの領域を探すことが一般的である。しかしながら、このような手法は必ずしも効率的であるとはいえない。一方、P2P ネットワークは仮想的なリンクにより接続されるオーバーレイネットワーク上で動作するため、仮想リンクによる接続関係をシステム自身が自由に変更することが可能である。そのため、ブルームフィルタを利用した検索においても、検索に適したネットワークトポロジを前処理として構成することで、さらなる性能向上が期待できる。しかし、既存手法 [3, 4, 5] においては、ブルームフィルタの散布手法を改良することは考慮されているが、検索に適したネットワークを構築することに關してはあまり考慮されていない。

そこで本研究では、従来の検索手法には改良の余地があると考え、更なる検索効率の向上を達成するために、検索に適したオーバーレイネットワークを構築する手法、および、構築したネットワークの構造を利用した問合せ転送手法を提案する。提案する手法では、各ピアが構成するブルームフィルタを問合せ転送の手掛かりに利用するだけでなく、探索したい資源を保持するピアが存在する位置にある程度の見当をつけながら問合せの転送ができるように、周囲のピアが構成するブルームフィルタを参照してトポロジを構築する。これにより、従来の手法よりも効率的な探索を実現する。本研究では、提案した手法を利用した場合、トポロジ構築を考慮しない従来の手法と比べて探索の効率が向上すること、および、提案手法は局所的な情報のみを利用してトポロジを構築することで、十分効率的な探索が実現できることをシミュレーション実験により示す。

以下、2 節で諸定義を述べ、3 節でブルームフィルタを利用したオーバーレイネットワーク構築手法、および、構築したネットワークの構造を利用した問合せ転送手法について説明する。また、4 節でシミュレーションによる評価を行い、最後に、5 節で本研究をまとめる。

2 諸定義

2.1 ネットワークと検索問題

本稿では P2P ネットワークを考える。ネットワークは無向グラフ $G = (V, E)$ によりモデル化する。ここで V はピア集合、 E はリンク集合である。リンクで繋がっている 2 つのピアはメッセージを交換することでお互いに通信を行うことが可能である。以下、ピア $v, w \in V$ 間のリンクを (v, w) で

表す。なお、以降特に断りなくネットワークに対してグラフの用語を用いる。

本稿ではトポロジの構築手法を提案するが、提案手法によりトポロジを構築する前は、何かしらのトポロジがオーバーレイ上に構築されていると仮定する。また一般に、P2P ネットワークにおけるトポロジでは、ピア集合、および、それらの接続関係は時間とともに動的に変化するが、本稿では説明を簡単にするため、ネットワーク上に存在するピアやリンクは常に一定であるとし、新しいピアやリンクが発生したり、既に存在するピアやリンクが離脱することはないものと仮定する。なお、これら仮定は議論を単純化するために導入するものである。システムが非同期であっても、 α -シンクロナイザ [1] を用いることで、通信コストを増加させることなく簡単にラウンド同期システムを実現できる。また、ピアの参加・離脱があるような動的ネットワーク上においても、提案手法は実現することが可能である。

2.2 ブルームフィルタ

ブルームフィルタは、ある要素集合のインデックス情報を表現するための省スペースなデータ構造である。要素集合 X に対するブルームフィルタ $F(X)$ は、ある要素 x が X に含まれるかを調べるための手続き $find(x)$ を提供する。フィルタ F に対する x の $find$ 操作を $F.find(x)$ とする。この問い合わせ結果は片側誤りである。すなわち、 $x \in X$ であれば必ず $find(x)$ は真を返すが、 $x \notin X$ であっても $find(x)$ は偽を返すとは限らない。

以下、 a 個の要素の集合 $X = \{x_1, x_2, \dots, x_a\}$ を表すブルームフィルタ $F(X)$ の構成法を説明する。ブルームフィルタは m ビットの配列から成り、配列における各ビットの初期値は全て 0 である。この配列の各ビットを $F[1] \sim F[m]$ で表す。また、ブルームフィルタでは複数個のハッシュ関数を利用する。以下、 k 個のハッシュ関数を用意し、これらを $h_i (1 \leq i \leq k)$ で表す。 h_i は任意の要素 x を引数として取り、1 以上 m 以下の値を返す。つまり、 \mathcal{X} を可能な全ての要素の集合とすると、 $h_i: \mathcal{X} \rightarrow [1, m]$ である。 X に対するブルームフィルタ $F(X)$ は、 X に含まれる全ての要素 x について、 $F[h_1(x)], \dots, F[h_k(x)]$ の値を全て 1 に変更することで構成される。また、構成されたブルームフィルタ $F(X)$ に対する問い合わせ $F.find(y)$ は、 $F[h_1(y)], \dots, F[h_k(y)]$ の論理積の結果として返す。すなわち、 $F[h_1(y)], \dots, F[h_k(y)]$ のうち、0 であるビットが 1 つでも存在した場合、 $y \notin X$ と判断され、全ての値が 1 だった場合、 $y \in X$ と判断される。このとき、実際に $y \in X$ であった場合は、 $F[h_1(y)], \dots, F[h_k(y)]$ の値は必ず 1 になるため、 $find$ 操作の結果は必ず真となる。一方、実際は $y \notin X$ であるにもかかわらず、 $find$ 操作が真を返す場合がある。一般的にこの性質を false positive 性という。 $y \notin F$ にもかかわらず $F.find(y)$ が真

を返す確率 (= 誤り確率) は以下ようになる。

$$\left(1 - \left(1 - \frac{1}{m}\right)^{ka}\right)^k \approx (1 - e^{-ka/m})^k = p^k \quad (1)$$

ここで、 $p = 1 - e^{-ka/m}$ としている。このとき、 p は F の全ビット中に 1 が占める割合と等しくなる。上記の式から、ブルームフィルタ中に含まれる 1 の割合が多い場合、誤り確率は高くなってしまいうことがわかる。

2.3 ブルームフィルタを用いた検索手法

P2P ネットワークに参加しているピアは、それぞれ固有のデータアイテムを所持している。検索を行う前処理として、各ピアはあらかじめ自分の所持するデータアイテム集合を表すブルームフィルタを構築し、これを D -近傍に存在する全ピアに散布する。以下、この D を散布半径と呼ぶ。ピア v がデータアイテム x の検索を行う場合、 x を検索したい旨を記した問合せを作成し、この問合せを、ピア間で繰り返し転送することで、最終的に x を所持するピアまで問合せを到達させることを目標とする。以下、検索したいデータアイテムを保持するピアを目的ピアと呼ぶ。問合せ転送の過程において、隣接ピアから問合せを受信したピアは、所有しているフィルタに対して $find(x)$ を実行する。そして、その返り値が真になるフィルタが存在した場合、そのピアに向かって問合せを転送する。この処理を繰り返すことで、最終的に問合せを目的ピアに到達させることができる。

一般的に、フィルタの散布半径が広ければ広いほど、所持するデータアイテム集合のインデックス情報を多くのピアに知らせることができるが、逆に周囲のピアから受信するフィルタの量が指数的に増加する。大量のフィルタを扱うのはコストがかなり管理が困難になるので、通常、これらのフィルタを受信したピア毎に論理和を取って統合する。以下、ピア v が隣接ピア w から受信した全てのフィルタについて論理和を取って統合したものを F_{vw} とする。しかし、フィルタの統合を行うことで、フィルタ中に立つ 1 の割合は増加してしまうため、誤り確率の増加が引き起こってしまう。また、既存の検索手法においては、 $F_{vw}.find(x) = 1$ となる隣接ピア w が存在しない場合、 x のインデックス情報を含むブルームフィルタを発見できるまで、問合せをランダムな方向に転送するのが一般的である。しかし、これでは $F_{vw}.find(x) = 1$ となる隣接ピア w を発見するまでに時間を要してしまうため、非効率的であるといえる。

3 提案手法

2.3 節で述べたように、P2P ネットワークはオーバレイ上で構成されるため、そのトポロジをある程度自由に構築することができるという特徴を持つ。しかし、既存手法においては、効率的な検索

を行うために検索に適したネットワークを構築する、ということに関してはあまり考慮されていない。そこで本研究では、従来の検索手法に改良の余地があると考え、ブルームフィルタを利用したオーバレイネットワークの構築手法、および、構築したネットワークの構造を利用した問合せ転送手法を提案する。

3.1 提案手法の方針

提案手法の基本アイデアは、問合せ転送時、目的ピアの位置におおよその見当をつけられるように、あらかじめ、各ピアの保持するデータアイテムによってピア同士の隣接関係が決定するようなトポロジを構築しておくところにある。

各ピアの構成するブルームフィルタは、そのピアが保持しているデータアイテムを元に構成されるものである。そこで、各ピアが構成するブルームフィルタをそのピアのアドレスを表している識別子と考える。以下、この識別子をアドレス ID と呼ぶ。このとき、ブルームフィルタは $\{0, 1\}$ のビット列で表される。そこで、各ピアのアドレスを多次元 $\{0, 1\}$ -ベクトルで表すとすると、各ピアは多次元の $\{0, 1\}$ -ベクトル空間に配置できる。 $\{0, 1\}$ -ベクトル空間上に配置された 2 点間の距離は対応する 2 つのビット列のハミング距離によって定義する。提案する手法において、各ピアは自身のアドレス ID と他のピアのアドレス ID とを比較し、それらのハミング距離が最も小さくなるピアを順に t 個選択し、それらのピアに対してリンクの接続要求を送信する。以上の操作を行うことで、アドレス ID が類似したピア同士を近接したトポロジを構築することができる。これにより、問合せ転送時、アドレス ID の情報を参照することで、探索したい資源を保持するピアが存在する位置にある程度の検討をつけることができると考えられる。ただし、アドレス ID を参照しただけでは、目的のデータアイテムを持つピアの正確な位置を把握することは出来ない。そこで、目的ピアのおおよその位置に見当をつけた後は、ブルームフィルタを利用した従来の手法と同様に、予め散布されたブルームフィルタの情報を参照して、問合せを目的のデータアイテムを持つピアに到達させる。以上の操作により、従来手法における $F_{vw}.find(x) = 1$ となる隣接ピア w を発見するまでは問合せをランダムウォークせざるをえないという点を改善し、検索効率を向上させることが可能になる。

一般的に、ブルームフィルタのサイズは十数キロビット以上及ぶため、ピアが構成するブルームフィルタをそのままの形でピアのアドレス ID とみなしてしまうと、ベクトル空間のサイズが莫大なものになってしまう。そこで、構成したブルームフィルタを OR 操作で縮約したものをそのピアのアドレス ID と解釈することで、ネットワーク空間のサイズを適切なものに調整する。なお、アドレス ID の全ビット中に 1 が占める割合を p' は以下のようになる。

$$p' = 1 - (1 - p)^c \quad (2)$$

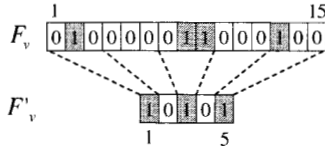


図 1: F'_v の作成例

3.2 トポロジ構築手法

以降の節では、提案手法の概要を述べる。本節では、トポロジ構築プロトコルの概要を説明する。

従来の検索手法と同様に提案手法においても、各ピアが構成するブルームフィルタ F は m ビットの配列と k 個のハッシュ関数から成る。配列の各ビットを $F[1] \sim F[m]$ で表し、ピア v が構成するブルームフィルタを F_v で表す。ピア v は最初に、2.2 節で説明した手法を用いてブルームフィルタ F_v を構成する。つまり、各ピアが保持するデータアイテムを $X = \{x_1, x_2, \dots, x_n\}$ とすると、ピア v が構成するブルームフィルタは、 X に含まれる全要素 x について、 $F[h_1(x)], \dots, F[h_k(x)]$ の値を全て 1 に変更したものである。また、ピア v のアドレス ID は、構成したフィルタ F_v を参照し、先頭のビットから順番に c ビットずつ OR 操作を行うことで縮約したものと定義する (図 1)。以下、ピア v のアドレス ID を F'_v とし、 c を縮約率と呼ぶ。以上の操作により、アドレス ID のサイズは m/c となり、アドレス空間全体のサイズは $2^{m/c}$ となる。

全てのピアがアドレス IDF' を構成した後、各ピアはリンクの接続要求を送信するピアを決定するために周囲のピアのアドレス ID を参照する。ピア v がリンクを作成する場合、ピア v の D -近傍に存在する全ピア w のアドレス IDF'_w を参照し、 F'_v と F'_w のハミング距離が最も小さくなるピア w を順に t 個選出する。そして、選択した t 個のピアに対してリンク接続要求を送信し、これらのピアと双方向リンクを張る。以上の処理を行うことで、アドレス ID の類似したピア同士が近接しているという構造を持ったトポロジを構築することができる。これにより、資源探索時にアドレス ID を参照することで、探索したい資源を持つピアが存在する領域におおよその見当をつけながら問合せの転送を行うことができる。

3.3 フィルタの散布

ネットワーク構築後は、2.3 節で説明した従来の手法と同様の手法でブルームフィルタを散布する。フィルタを散布する範囲は距離 D までのピアとする。したがって、ピア v が受信するブルームフィルタは、ピア v から距離 i ($1 \leq i \leq D$) の全ピアが構成するブルームフィルタとなる。このとき、ピア v は、受信した各フィルタを隣接ピア毎に分類し、OR 操作で統合する。

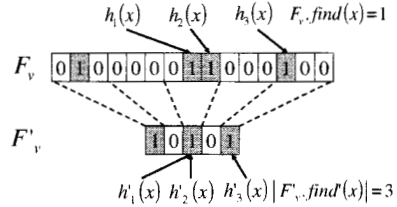


図 2: $find(x)$ と $find'(x)$ の関係

3.4 問合せ転送手法

問合せの転送を行う場合、従来の検索手法同様に、まずは $find(x) = 1$ となるフィルタを持つピアを探す。提案手法ではこのフィルタを持つピアを効率よく探し当てるために、3.2 節で構築したトポロジの構造を併せて利用する。

あるデータアイテム x の検索を行う場合、従来手法と同様に真偽値を返す手続き $find(x)$ が提供される。これに加えて提案手法においては、アドレス IDF' に対して定義される手続き $find'(x)$ も併せて提供される。以下、アドレス IDF' に対する x の $find'$ 操作を $F'.find'(x)$ とする。 $F'.find'(x)$ の返り値は k 個のハッシュ関数 h'_i を用いることで算出される。また、ハッシュ関数が $h'_i(x)$ は $h_i(x)$ を c で割った商に 1 を加えた値を返すと定義する。なお、 $h'_i(x)$ の値域は $[1, m/c]$ となる。このように h'_i を定義したとき、 $F'.find'(x)$ は $F'.find'(x) = \{i | F'[h'_i(x)] = 1\}$ と定義される。つまり、 $F'.find'(x)$ は要素集合を返す。また、 $F'.find'(x)$ が返す要素数を $|F'.find'(x)|$ と表す。以上のように $F'.find'(x)$ を定義すると、アドレス IDF' はフィルタ F を c ビットずつ OR 操作で縮約することで得られるビット列であるため、 $F.find(x)$ が真を返す場合、 $|F'.find'(x)|$ の値は必ず k となるという性質が生まれる (図 2)。したがって、 $F.find(x)$ が真を返すピア集合は、 $|F'.find'(x)| = k$ となるピア集合に含まれることになる。また、構築したトポロジはアドレス ID のハミング距離が小さいピア同士を近接させているため、 $|F'.find'(x)| = k$ となるピアの集合は、お互いに隣接している可能性が非常に高い。以下、 $|F'.find'(x)| = k$ となるピア集合を探索ピア集合と呼び、探索ピア集合が存在する部分空間を探索部分空間と呼ぶ。さらに、探索部分空間に近接しているピアは、 $|F'.find'(x)|$ の値が k に近い値であると考えられ、逆に探索部分空間から離れた位置に存在するピアは、 $|F'.find'(x)|$ の値が 0 に近い値であると考えられる。

以上のトポロジ構造を利用して、 $|F'_w.find'(x)|$ の値がなるべく大きくなるようなピア w へ問合せの転送を繰り返し、問合せを探索部分空間まで到達させることができると期待できる。その上、目的ピアは近隣に自身が構成したブルームフィルタを散布しているため、探索部分空間内のピアは $F.find(x) = 1$ となるブルームフィルタを保持している可能性が非常に高い。そこで、問合せが探索部分空間に到達した後は、探索ピア集合に含まれるピアを重点的に訪問し、 $F.find(x) = 1$ とな

表 1: c と p' の関係

p'	0.1	0.2	0.3	0.4	0.5	0.6	0.7
c	40	90	140	210	280	360	480

るブルームフィルタを探し出す。そして最終的に、このブルームフィルタを参照することで実際に x を保持するピアを探し当てることが可能になる。

4 シミュレーション評価

本章では、本稿で提案したネットワーク構築手法、および、問合せ転送手法の有用性を検証する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、提案手法と従来の検索手法の性能を比較するために、以下に示す 2 つの手法について比較・評価を行った。まず、1 つ目の手法は、提案したネットワーク構築手法と問合せ転送手法を併用した検索手法（以下、提案手法と呼ぶ）であり、2 つ目の手法は、ネットワークはフラットボロジを用い、かつ、問合せ転送手法は従来のものを用いた検索手法（以下、従来手法と呼ぶ）である。

提案手法においてはネットワークを構築する際に D -近傍のピアからアドレス ID を収集する。このとき、 D の値が大きければ大きいほど、ネットワークに存在する多くのピアのアドレス ID を収集できるため、 $|F'.find'(x)| = k$ となるピア集合がお互いに隣接している可能性が非常に高いトポロジを構築できると考えられる。しかし、このように大域的にアドレス ID を収集しようとすると、アドレス ID の収集に費やされる通信量が膨大なものになってしまい、効率的な検索が実現できるとはいえない。しかしながら、実際のシステムでは、アドレス ID の（ネットワーク上における地理的な）分布に極端な偏りがあるケースは少ないと考えられるため、 D の値を抑え、局所的な範囲のピアのみからアドレス ID を収集することでネットワークを構築したとしても、 $|F'.find'(x)| = k$ となるピア集合がある程度隣接しているトポロジの構築が期待できる。この場合、当然ながら検索効率は大域的にアドレス ID を収集した場合よりも劣る可能性があるが、仮に削減した通信量に見合った性能を発揮できたとしたら、提案手法は非常に有用であるといえる。なお、本実験においては、検索に用いるブルームフィルタ F を散布する場合の半径と、ネットワーク構築時に収集するアドレス ID F' を散布する場合の半径とを別のものとして定義する。以下、ブルームフィルタ F における散布半径を D と表し、フィルタ散布半径と呼ぶ。また、アドレス ID における散布半径を D' と表し、ID 散布半径と呼ぶ。以下、提案手法においてアドレス ID を収集する際に、 D' を無限大に設定してネットワーク中の全ピアからアドレス ID を収集する場合を大域手法、 D' を小さく設定して周囲のピアのみからアドレス ID を収集する場合を局所手法と呼ぶ。

4.1 実験設定

提案手法において構築するネットワークは 10000 ピアで構成し、フィルタサイズ m を 10000 に、ハッシュ関数の個数 k を 5 に設定する。全ピアはデータアイテムを一律 5 個ずつ保持する。これらのデータアイテムに関して複製は存在せず、種類は全て異なるとする。つまり、ネットワーク上には 50000 個の異なるデータアイテムがそれぞれ 1 個ずつ存在する。ネットワーク構築時、ID 散布半径 D' を大域手法では無限大、局所手法では 3 とする。また、フィルタ散布半径は D を 3 に、リンク接続要求を送信するピアの数 t は 5 に設定する。このとき、 p' の値を変化されるために縮約率 c の値を変化させている。なお、各 p' に対応する縮約率 c の値は表 1 のようになる。また、極端に次数が多いピアが発生するのを防ぐ目的で、リンク接続要求を送信したピアの次数が x 以上であった場合、そのピアへのリンク接続要求を破棄し、代わりにして次にハミング距離の小さいピアに改めてリンク接続要求を送信するという仕組みを取り入れた。本実験において、この x の値を 16 に設定した。以上の設定で、アドレス ID の各ビットに 1 の立割割合を示す p' の値を 0.1 から 0.7 まで約 0.1 刻みで変化させ、それぞれの場合の性能を調査する。

一方、従来手法で用いるフラットボロジは、提案手法において構築するネットワークと同様、10000 ピアで構成され、フィルタサイズ、ハッシュ関数の個数、データアイテムの配置も、提案手法と同様に設定した。以上の設定は、公平を期するために、各ピアの次数の平均がほぼ一定になることを意図して設定されている。その理由は、一般的に検索問題を解くにあたり、各ピアの次数が多ければ多いほど周囲により多くの情報を伝達できるため、検索性能は良くなることがわかっているからである。

以上の設定で、提案手法における大域手法および局所手法、そして従来手法の 3 手法を比較した。なお、これらのシミュレーション実験において、問合せは各実験において 5000 個生成する。また、各手法における実験では、問合せ生成シナリオ、ハッシュ関数等は同一のものを用いている。評価した項目は、各ピアの次数の平均値、各問合せが目的ピアに到達するまでの平均ホップ数、検索成功率である。検索成功率は、生成された問合せに対して、TTL 内に目的ピアに到着した問合せとの割合と定義する。加えて、提案手法（大域手法と局所手法）に関しては、各データアイテム x に対する探索ピア集合に含まれるピア数の平均（以下、探索ピア数と呼ぶ）と、各問合せが初めて探索部分空間に到達するまでの平均ホップ数（以下、探索空間到達ホップ数と呼ぶ）も調査している。

4.2 実験結果

4.1 節で説明した設定に基づいて実行したシミュレーション実験の結果を表 2 に示す。この結果を参照すると、提案手法における平均ホップ数・検索成功率は大域手法・局所手法ともに、従来手法

表 2: 実験結果

評価項目	評価手法	p' の値						
		0.1	0.2	0.3	0.4	0.5	0.6	0.7
平均回数	大域手法	6.40	6.23	6.05	5.87	5.82	5.91	6.08
	局所手法	6.25	6.16	6.05	5.95	5.91	5.96	6.14
	従来手法	6.00						
平均ホップ数	大域手法	45.42	46.78	52.26	57.59	57.88	55.49	67.32
	局所手法	60.88	59.54	64.23	67.56	67.85	62.10	53.19
	従来手法	81.29						
検索成功率	大域手法	0.994	0.998	0.995	0.993	0.992	0.993	0.984
	局所手法	0.982	0.992	0.987	0.986	0.986	0.991	0.995
	従来手法	0.965						
探索空間到達ホップ数	大域手法	45.10	35.13	16.79	6.24	3.17	1.88	1.32
	局所手法	60.47	47.69	24.22	7.60	3.18	1.51	0.62
探索ピア数	提案手法	1.0	1.8	27.0	186.6	341.2	1153.0	2427.6

よりも優れていることがわかる。また、提案手法における平均回数は従来手法のものに比べてあまり変わらない。これにより、提案手法における戦略が生かされたことによって、性能が向上していると考えられる。以上より、提案手法は提案手法よりも検索効率が優れていることがいえる。

次に、大域手法と局所手法の結果を比較すると、大域手法の方がやや検索効率が良い。しかし、そこまで効率に大きな差はあるわけではない。したがって、局所的なピアのみからアドレス ID を収集した場合でも、削減した通信量に見合った、十分な検索効率を実現できるといえる。また、提案手法において p' を変化させた場合の結果を参照すると、提案手法においては、 p' の値を変化させることにより、検索性能に変化が生じることがわかる。具体的には、大域手法においては p' の値が小さいほど検索性能が良く、また、局所手法においては p' の値が小さい、または、大きいほど検索性能が良いことがわかる。

5 まとめ

本稿では、ブルームフィルタを利用したオーバレイネットワークの構築手法、および、構築したネットワークの構造を利用した問合せ転送手法を提案した。この手法では、問合せ転送時に目的ピアが存在する位置におおよその見当がつけられるように、各ピアのアドレス ID を参照してリンクを張っていく。また本稿では、大域手法、局所手法、従来手法の比較・評価をシミュレーションにより行った。その結果、提案手法を用いることで効率的な検索を実現できる他、トポロジを構築する際に、大域的な情報を参照せずに、局所的な情報を参照するだけでも十分効率的な検索が実現できることをシミュレーション実験により示した。

今後の研究課題としては、提案手法の動的環境への適応性の調査、および、提案手法とフィルタ散布手法を改良した既存手法とを併用した場合の検索効率の調査などが挙げられる。

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 基盤研究 (B)17300020、若手 (B)19700058)、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)、および、(財) 栢森情報科学振興財団によるものである。

参考文献

- [1] B. Awerbuch. Complexity of network synchronization. *Journal of the ACM (JACM)*, Vol. 32, No. 4, pp. 804–823, October 1985.
- [2] A. Broder and M. Mitzenmacher. Network applications of bloom filters: A survey. *Internet Mathematics*, Vol. 1, No. 4, pp. 485–509, 2004.
- [3] J. Kubiawicz and S. C. Rhea. Probabilistic location and routing. In *Proceedings of the 21st Conference of the IEEE Communications Society (INFOCOM)*, Vol. 3, pp. 1248–1257, June 2002.
- [4] A. Kumar, J. Xu, and E. W. Zegura. Efficient and scalable query routing for unstructured peer-to-peer networks. In *Proceedings of the 24th Conference of the IEEE Communications Society (INFOCOM)*, Vol. 2, pp. 1162–1173, March 2005.
- [5] 高橋佑典, 泉泰介, 増澤利光. P2P ネットワークにおけるブルームフィルタを利用したインデックス情報散布法の改良. 電子情報通信学会技術研究報告 (NS2006-21), 第 106 巻, pp. 1–4, May 2006.