

P2P 型ファイル検索における高スループット・ピアの自動選択機構

渡辺 元晴† 河野 健二†† 岩崎 英哉†† 益田 隆司††

† 電気通信大学大学院電気通信学研究科情報工学専攻

†† 電気通信大学情報工学科

電子メール : watanabe@zeus.cs.uec.ac.jp , {kono, iwasaki, masuda}@cs.uec.ac.jp

要旨

本稿では、P2P 型ファイル検索において、ファイルの取得時間が短いと期待されるピアを自動的に選択する手法を提案する。既存の P2P 型ファイル検索では、最適なファイルの取得先は分からない。提案する選択手法では、ピアは自身が計測するスループット情報と他のピアが提供するスループット情報から、ファイル取得時間が最短となると期待されるピアを自動的に選択する。スループット情報の交換は、通常のファイル検索メッセージに付与する形で実現でき、本手法を用いても通信回数が増加することはない。定評のあるネットワーク・シミュレータである ns-2 を用いたシミュレーションでは、30 回のファイル検索、5 回のファイル取得を行えば、80%以上の確率で最適なピアが選択できることを確認した。

Selecting a High Throughput Peer in Peer-to-Peer File Search

Motoharu Watanabe† Kenji Kono†† Hideya Iwasaki†† Takashi Masuda††

†Course in Computer Science and Information Mathematics,
Graduate School of Electro-Communications, University of Electro-Communications

††Department of Computer Science, University of Electro-Communications

E-mail: watanabe@zeus.cs.uec.ac.jp , {kono,iwasaki,masuda}@cs.uec.ac.jp

Abstract

This paper shows a technique that automates selection of optimal peers in Peer-to-Peer (P2P) file search systems. In P2P file search systems, each peer cannot determine which peer are able to provide the highest throughput for that peer. In the proposed method, each peer measures effective throughput of file uploading/downloading, and exchanges this throughput information among the peers. Using this information, each peer selects the optimal peer. Since this information is piggybacked on normal P2P messages, the number of messages does not increase in our method. Simulation results demonstrate that over the 80% peers can find the optimal peer after 30 file searches and 5 downloads.

1 はじめに

安価に利用可能な常時接続網の普及により、従来のクライアント・サーバ型のシステム・モデルとは異なる Peer-to-Peer (P2P) 型のシステム・モデルに基づいた分散アプリケーションが広く利用されるようになっていく。このような分散アプリケーションには、Gnutella [1], Freenet [5], Napster [2], WinMX [4] などがある。

P2P 型システムでは、個々のピアがサーバとクライアントの両方の役割を担い、各ピアは不特定多数のピアと直接情報交換ができる特徴を持つ。P2P 型システムの利点を次に示す。1) ピア数の増減に応じて性能や機能を柔軟に向上させることのできるスケラビリティがある。2) サーバのような特定のピアに依存することがないので、特定のピアが故障するなどの障害が起きても P2P ネットワーク全体は影響を受けずに機能する耐故障性、可用性がある。3) 高価かつ高性能なサーバは必要無く、個人が持つような PC で P2P ネットワークを構成することができるので、大規模な分散システムを安価に構築できる。

P2P ネットワークを用いた一般的な分散ファイル検索は、接続しているピアに検索メッセージを送信する。検索メッセージを受信したピアは、送信元以外の接続ピアにそのメッセージを転送すると同時に、そのメッセージが要求するファイルが自ピア上にあるかどうか調べる。ファイルがあれば送信元にファイルが見つかったことを知らせるメッセージを返信する。複数のピアでファイルが見つかった場合、検索要求したピアは、複数の検索結果の中からファイルの取得先を手動で選択する必要がある。

本稿では、検索後のファイル取得時に高スループットが期待できる最適なピアを自動的に選択する方法を示す。ここでいう最適なピアとは、ユーザにとって最短時間でファイルを取得できるピアのことをいう。

最適なピアを選択するひとつの方法は、各ピアまでのスループットを実際に計測し、最適なピアを見つける方法である。しかし、この方法ではピアの台数の増加に伴い、通信量が爆発してしまう。本稿で示す手法では、各ピアに関するスループット情報を収集し、最適ピア選択の判断材料として用いる。スループット情報の収集時にはすべてのピアとは通信せず、低スループット・ピアを早期に排除し、段階的に候補を絞り込んでいくようにする。さらに、各

ピアからのスループット情報は、通常のファイル検索やダウンロード時のメッセージの中に含ませることにする。こうすることにより、通信回数を増加させることなく最適なピアを発見することができる。

本稿では、P2P 型ファイル検索用のプロトコルのひとつである Gnutella プロトコルを拡張し、提案する最適ピア選択アルゴリズムを実現した。また、ネットワークシミュレータ ns-2 を用いたシミュレーションを行い、その有効性の検証を行った。ピアの総数 100 の P2P ネットワークを構築し、そのネットワーク上で 5 回のファイル取得を行えば、80% 程度のピアが最適なピアを発見できていることを確認した。

2 P2P 型ファイル検索と問題点

2.1 P2P 型ファイル検索

Gnutella などで行われている基本的な P2P 型ファイル検索の仕組みを図1に示す。

1. 検索を行うピア A は隣接するピア B, C に Query を送信する。
2. Query を受信したピア B, C は Query の転送元はピア A と記憶する。ピア B は隣接するピア E に Query を転送し、ピア C は隣接するピア D, E に転送する。ここで、ピア E はピア B, C から Query を受信するが、同一の Query のため、転送元はピア B と記憶するそして、ピア E はピア F に Query を転送する。
3. ピア F はローカル内に Query 対するファイルを保持しているか検索した結果、自ピアがファイルを保持していることが分かる。
4. ピア F は、見つかったファイル情報 (QueryHit) を Query の送信元ピア E に送信する。
5. QueryHit を受信したピア E は 2 で記憶した Query の転送元ピア B へ QueryHit を転送する。ピア B でも同様に Query の転送元ピア A へ転送する。
6. ピア A は受信した QueryHit からピア F の情報を取得する

Freenet [5] や WinMX [4] など他の検索システムでは、具体的な検索方法やダウンロード方法は異なる。

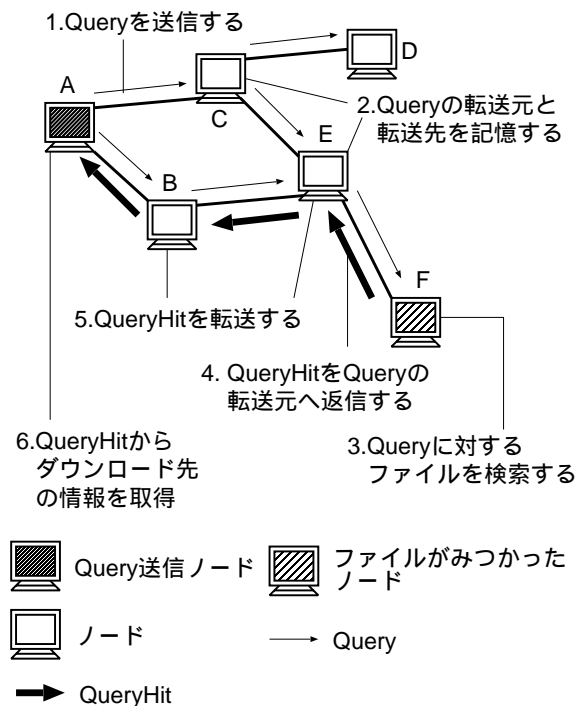


図 1: 基本的な P2P 型ファイル検索

るが、検索結果に対して複数のピアがファイル取得先として見つかり、その中から手動で実際のファイル取得先を選択しなければならない点では同様である。

2.2 P2P 型ファイル検索の問題点

P2P 型ファイル検索では、検索要求を波紋状に転送するため、いつ検索が終了し、すべての応答がいつ返ってくるのをあらかじめ知ることができない。また、検索結果を複数のピアを中継するため、検索に時間を要する。この問題に焦点を当て、キャッシングやグループ化などを行い検索を高速化する方法が提案されている [9, 7, 11, 10]。

これらの研究とは異なり、本論文ではユーザにとってファイルの取得に要する時間が短時間になることが期待される最適なピアを見つけることを目的とする。ネットワーク上のランドマークを利用することで最適なピアを見つける研究 [9, 6] もあるが、我々の方式はランドマークを必要としない点に特徴がある。

以下、ある検索に対してヒットしたファイルを所有するピアをサーバ・ピアと呼び、実際にそのファ

イルを取得しようとしているピアをクライアント・ピアと呼ぶこととする。

ある検索を行った結果として、複数のサーバ・ピアからファイルが取得可能であった場合、それらのピアから最適なピアを見つける最も単純な方法は、各サーバ・ピアと実際に通信を行いスループットを実測する方法である。しかし、この方法では多数のピアと通信しなければならない。さらに、サーバ・ピアやネットワークの負荷は常に変動するため、最適なピアを見つけるには定期的に計測をやり直す必要がある。そのため、この方法ではネットワークに与える負荷が爆発的に増え、P2P 型のネットワークには適用できない。

提案手法では取得先の候補となるサーバ・ピアを選ぶために、実際に通信を行ってスループットを計測するようなことはしない。スループットの実測対象となるピアを効果的に限定し、かつスループットの実測はファイル取得時にのみ副産物として行うようにし、通信回数を増加させることなく最適なピアを発見できる手法を示す。あるピアにとって最適となる可能性が低いピアをあらかじめ測定対象からはずし、最適ピアとなる可能性のあるピアを対象を絞りこむ。

Andy ら [8] の調査によれば、クライアント・ピアからみてどのサーバ・ピアがよりよいスループットを供給するかという順位付けを行ったとすると、その順位は長期にわたり変動することがない。すなわち、一度最適なピアを発見することができれば、長期にわたりそのピアが最適である確率が高い。したがって、ファイルの取得対象となるピアを段階的に絞り込んでいけば、最適なピアを発見できると期待される。なお、Andy らの調査は、米国内の 47 のサーバに総数 50,000 に及ぶファイルを取得する実験を行った結果である。

3 最適ピアの自動選択機構

目的を達成するためにクライアント・ピアは各サーバ・ピアのスループット情報を収集する。そのスループット情報には (1) 各サーバ・ピアが報告してくるスループット、(2) クライアント・ピアが計測するスループットがある。これらの情報を用いて最適なサーバ・ピアを選択する。以下、3.1 節、3.2 節でその具体的な方法を示す。

3.1 低スループット・ピアの排除

3.1.1 潜在的スループットを用いる方法

狭帯域のネットワークに接続されているなどの理由で、そもそも高スループットを出し得ないサーバ・ピアをあらかじめ選択対象から排除するようにする。サーバ・ピアは他のピアと通信した時のスループットを計測し、最大のスループットを記録しておく。このスループットは、そのピアが供給しうる最大のスループットであり、これを潜在的スループットと呼ぶ。サーバ・ピアはクライアント・ピアに潜在的スループットを通知する。これによって、クライアント・ピアは潜在的スループットの低いピアをファイル取得先の候補から排除することができる。

3.1.2 利用可能スループットを用いる方法

潜在的スループットだけを用いる方法では、実際に高スループットを出すピアを選択できるとは限らない。なぜなら、サーバ・ピアの負荷を考慮しておらず、サーバ・ピアが他のピアと行っている通信を無視しているためである。

そこで、サーバ・ピアが実際に供給できるであろうスループットを用いて負荷の高いピアを排除する。このスループットを利用可能スループットと呼ぶ。利用可能スループットとは、潜在的スループットから他のクライアント・ピアとの通信に使用しているスループットを引いたものである(図2を参照)。

潜在的スループットが高くとも利用可能スループットの低いピアは、その時点では低スループット・ピアと判断してよい。しかし、そのサーバ・ピアは潜在的に高いスループットを供給しうるため、選択候補から完全に除外することはせず、後に他のサーバ・ピアと比較して排除するか否か決める。

3.2 遠距離ピアの排除

3.1節で示した方法ではネットワーク距離を考慮していなため、遠距離にあるサーバ・ピアを選択し

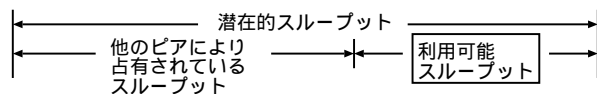


図 2: ピアのスループットの内訳

てしまい、高い実効スループットが得られない可能性がある。遠距離にあるピアを排除するため、クライアント・ピアはサーバ・ピアからファイルを取得する際に、そのファイル取得時のスループットを計測する。これを実測スループットと呼ぶ。

サーバ・ピアが報告してきた利用可能スループットに比べ、クライアント・ピアで実測したスループットがはるかに低い場合、クライアント・ピアはサーバ・ピアを遠距離ピアと判断し、それ以降のファイル取得時にはそのサーバ・ピアを候補から排除する。なぜなら、サーバ・ピアとクライアント・ピアの間に帯域の狭い通信路が存在し、実効スループットが低かったのであろうと推測できるからである。

実測スループットが低かった場合であっても、1)サーバ・ピアの潜在的スループットと利用可能スループットに大きな差があり、かつ2)クライアント・ピアが計測した実測スループットが利用可能スループットに近い場合、そのサーバ・ピアは一時的に過負荷になっていると推測できる。したがって、次にファイルを取得する際にはその負荷が低下し、サーバ・ピアの潜在的スループットに近い実測スループットが得られる可能性がある。したがって、このようなサーバ・ピアはファイル取得の候補からはずすことはしない。

3.3 選択アルゴリズム

3.1節、3.2節で述べた手法を実現するために、サーバ・ピアから収集したスループット情報を管理する表が必要である。この表をスループット管理表と呼ぶ。スループット管理表は、1)サーバ・ピアから得られる潜在的スループットと利用可能スループット、2)そのサーバ・ピアに対する実測スループットの最大値、3)その最大実測スループットを計測したときの、サーバ・ピアの潜在的スループット・利用可能スループットを持つ。

スループット管理表の更新は次のように行う。あるサーバ・ピアと通信を行うたびに潜在的スループットと利用可能スループットとを更新する。また、あるサーバ・ピアに対して最大の実測スループットを計測したとき、実測スループットの最大値を更新し、その時点でのサーバ・ピアの潜在的スループットと利用可能スループットとを更新する。

本稿で提案するサーバ・ピア選択アルゴリズムは次のように動作する。P2Pネットワーク上でサーバ・

表 1: クライアント・ピアが保持するスループット情報表

サーバ ピア	サーバ・ピアが 報告した情報 (KB/秒)		クライアント・ピアが計測した時の サーバ・ピアの情報 (KB/秒)			期待値 (KB/秒)
	潜在的 スループット	利用可能 スループット	スループット の最大実測値	潜在的 スループット	利用可能 スループット	
A	30	30	—	—	—	30
B	250	10	—	—	—	10
C	200	140	20	180	150	20
D	350	200	120	300	150	200

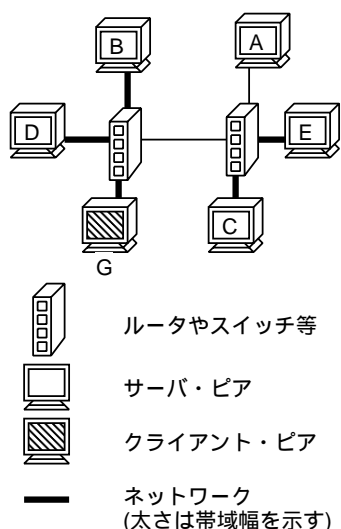


図 3: サーバ・ピアとクライアント・ピアのネットワーク上の関係

ピアとクライアント・ピアが図3に示した関係にある場合を例として説明する。クライアント・ピア G は D とは広帯域の通信路で接続されているが、A, B, C とは間に狭帯域の通信路があり高いスループットでファイルを取得することはできない状況である。

このとき、スループット管理表は表1に示したようになっているものとする。クライアント・ピア G は、サーバ・ピア A に対し一度もファイルの取得を行っていないため、スループットを実測していない。そのため、サーバ・ピア A からファイルを取得した場合、サーバ・ピア A の報告する利用可能スループットでファイルが取得できるものとみなす。サーバ・ピア B も同様である。このとき、サーバ・ピア A の潜在スループットが他のピアに比べて著しく低いため、サーバ・ピア A はあらかじめファイルの取得先候補から排除する。サーバ・ピア B に関しては、現時点での利用可能スループットは低い

ものの、潜在スループットは高いため、ファイル取得先の候補として残しておく。

クライアント・ピア G は、サーバ・ピア C, D からファイルを取得したことがあり、それぞれのピアに対する実測スループットの最大値がそれぞれ記録されている。サーバ・ピア C に対する最大値は 20、D に対する最大値は 120 である。また、その最大値を計測したときに、サーバ・ピアが報告してきた潜在スループットと利用可能スループットの値が記録されている。

サーバ・ピア C に対して最大の実測スループット 20 を計測したときの、C の利用可能スループットは 150 であり、実測スループットのほうがはるかに小さい。そのため、サーバ・ピア C はネットワーク上で遠距離にあるピアと推定される。そのため、C はファイル取得の候補から除外する。

サーバ・ピア D に対して最大の実測スループット 120 を計測した時、D の利用可能スループットは 150 であり、実測スループットが利用可能スループットに近い値を示している。そのため、サーバ・ピア D はネットワーク的に近距離にあるピアと推定される。この場合、表1により、クライアント・ピア G にとって期待値 120 のサーバ・ピア D が最適なピアとなる。

最後に、サーバ・ピア B の負荷が下がり、利用可能スループットが大きくなった場合には、サーバ・ピア B はファイル取得の候補になる。表1では、サーバ・ピア D の最大実測スループット 120 よりも、サーバ・ピア B の利用可能スループットが大きくなった場合に、サーバ・ピア B を最適ピアと判断する。

図4に選択アルゴリズムの疑似コードを示す。初めに 1 行目ではクライアント・ピアが実測値があるか判断する。無い場合は、15 行目にてサーバ・ピアの申告する利用可能スループットが期待値となる。

```

1  if (実測値があるか)
2    if ( * (最大実測値を計測した時のサーバ・
ピア
3      の利用可能スループット )
4      < スループット最大実測値 )
5    if (サーバ・ピアが申告する
6        利用可能スループット)
7      < スループット最大実測値 )
8      期待値 = サーバ・ピアが申告する
9              利用可能スループット
10   else
11     期待値 = スループットの最大実測値
12   else
13     期待値 = スループットの最大実測値
14 else
15   期待値 = サーバ・ピアが申告する
16           利用可能スループット

```

図 4: 選択アルゴリズムの疑似コード

ある場合は、2 行目にて最大実測値とそれを計測した時の利用可能スループットから明らかに遠距離にあるピアを判断する。最大実測値と利用可能スループット間にどれくらいの比率から遠距離ピアか判断するためのパラメータとして、この時、2 行目では利用可能スループットに $\alpha = 1/3$ をかけている。

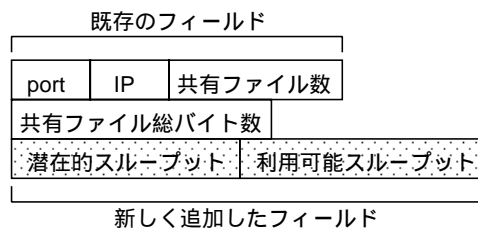
13 行目では遠距離ピアの場合、利用可能スループットが最大実測値より大きくても、最大実測値以上はでないため、期待値は最大実測値とする。そうでない場合、遠距離ピアではないと判断されるために、5 行目ではサーバ・ピアが申告した利用可能スループットと最大実測値を比較し、8, 11 行目では値の小さい方を期待値とする。

3.4 スループット情報の収集

前節で述べた方式の実現するために、Gnutella プロトコルに拡張を行った。サーバ・ピアからスループット情報を収集するために、通常のファイル検索やファイル取得に用いられている通信メッセージにそれらの情報を埋め込み、本方式を適用しても通信回数が増大しないようにした。これは、P2P 型のファイル検索では、ピア数の増大とともに通信回数が増大する傾向にあるため、通信回数を増やすことは好ましくないからである。

スループット情報が収集できるように、Gnutella プロトコルを次のように拡張した。2.1 節で述べたように、通常の Gnutella ではピア情報を収集するために定期的に Ping メッセージを送信している。その返

拡張したPongパケット



拡張したQueryHitパケット

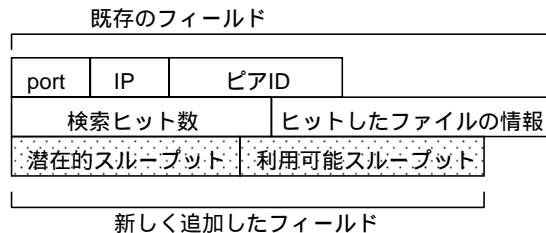


図 5: 拡張した Gnutella パケット

答である Pong メッセージを拡張し、サーバ・ピアの潜在的スループットと利用可能スループットとを返信できるようにした。また、ファイル検索用のメッセージである Query に対する返信である QueryHit にも同様の拡張を行った。標準の Pong, QueryHit に対する拡張を図5に示す。

3.5 スループット管理表

スループット管理表は、Pong や QueryHit を受け取ったすべてのサーバ・ピアの情報を保持しなければならない。ピア数の増大とともに爆発的にエントリ数が増加する。このようなエントリ数の増大を防ぐため、一定数よりもエントリ数が増加しないように不要なエントリを削除しなければならない。

クライアント・ピアは、各サーバ・ピアの情報に期限切れになる時刻 (expire) を設定する。QueryHit から新たにサーバ・ピアが見つかった場合に、expire を設定する。さらに、サーバ・ピアからファイルを取得する時に、選択したサーバ・ピアの expire の期限を延長するために一定時間先に再設定しなおす。これにより、QueryHit を返信するサーバ・ピアがあるとしても、低スループットの選択されないサーバ・ピアの情報が削除されるため、高スループットが計測できる可能性のあるサーバ・ピアの情報が残る。

4 シミュレータによる実験

ネットワークシミュレータである ns-2[3] を用いて、本稿で示した最適ピア選択アルゴリズムの有効性の検証を行った。ns-2 はカリフォルニア大学バークレー校 (UCB) で開発され、多くの研究プロジェクトで利用された実績のあるシミュレータである。

ns-2 上でのシミュレーションを行うため、3.4節で述べた通りに Pong, QueryHit メッセージを拡張した Gnutella プロトコルを実現した。ns-2 上の各ノードはスループット情報表を保持するようにし、Pong, QueryHit メッセージを受信するたびにスループット情報表を更新する。ファイル取得時には、3.3節で述べた方法でサーバ・ピアを選択する。

提案するアルゴリズムが最適なピアを選択しているか確認するために、各ピアが定期的にファイルを取得するシミュレーションを行い、ピアのファイル取得時のスループットを計測した。

ネットワークは帯域の広い基幹部があり、ピアはその基幹部に接続する形態になっている。ネットワークの帯域は接続する各ピアによって異なる。また、ピア間のレイテンシやホップ数を考慮し、各ピア間で通信に必要なホップ数の分布は 1~36 までの正規分布にならう。

シミュレーション環境は、ピア総数 400 のうち 100 ピアが Query やファイル取得要求を送信する。残りの 300 ピアは P2P ネットワークに参加せずに、Web や Ftp などによりネットワークの帯域を占有しているピアを想定しているためである。

図6は、先のシミュレーション環境で、ネットワークの帯域幅 24Mbps のピアが 180 秒毎にファイルを取得すると同時に計測したスループットのグラフである。縦軸はピアが計測した実効スループット、また、横軸はシミュレーション時間を示す。実効スループットは、ピアがファイル取得時に実測するスループットである。スループットの理想値とは、他の通信が一切行われておらずネットワークを占有した状態で得られるスループットの値である。シミュレーション時間 10 分までは、サーバ・ピアの情報がなく、色々なピアからスループットを計測するために、実効スループットが安定しない。10 分以降は最適なピアが見つかるために、スループットの理想値に近い実効スループットを計測している。

次に、図7に P2P ネットワーク内において、全 100 ピアに対してどれくらいのピアが最適なピアを選択

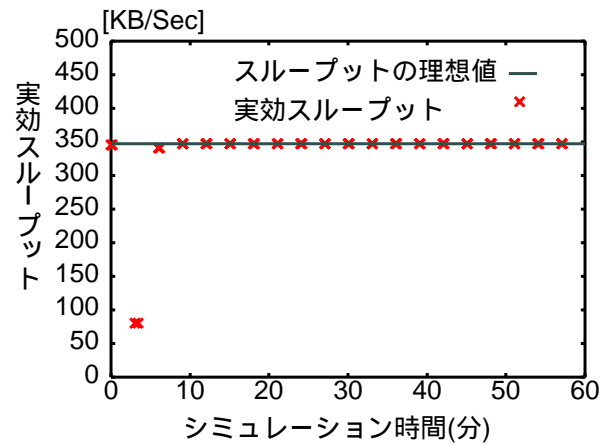


図 6: 帯域幅 24Mbps のピアが計測したスループット

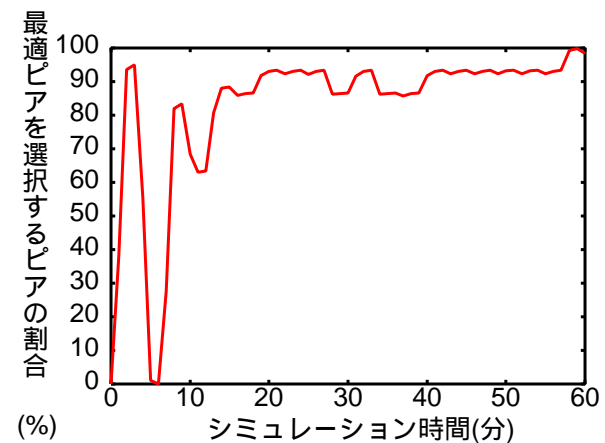


図 7: 最適ピアを選択するピアの割合

しているかの割合を示す。ここでいう最適なピアとは、実効スループットがスループットの理想値と同じ値である場合に、その実効スループットを計測できたサーバ・ピアのことを指す。図7の縦軸は、P2P ネットワーク内において最適なピアを選択するピアの割合、また、横軸はシミュレーション時間である。

図7を見るとシミュレーション時間 15 分までは、各ピアはサーバ・ピアの情報を収集するために、高スループット・ピアではない可能性のあるピアからもスループットを実測する。その場合は、最適なピアからファイルを取得していないと判断するので、最適なピアを選択するピアの割合は低くなる。しかし、15 分以降になるとサーバ・ピアの情報が集まり、80%のピアは最適なピアを見つけることが確認できる。

5 関連研究

Binning[9] は、ノードが構成する P2P ネットワークを物理ネットワークに近づけることで、ノードが計測するスループットを向上させることが目的としている。その方法は、ランドマークノードまでのレイテンシを計測することにより、ノードの位置を推測し、物理的に近いノード同士をグループ化する。この方式では、ランドマークを用いるために、そのノードは常に安定していなければならないという欠点がある。我々の方式は、ランドマークを必要としない利点がある。

Peer Communities[7] は、ノードのグループ化の研究だが、同じ目的・趣味をもつピア同士で同一のコミュニティ(グループ)を構築し、検索などの効率化を提案している。我々の方式は、ファイル取得時に計測したサーバ・ピアのスループットを比較することで、最適なピアを見つける。その結果によって、ファイル取得に要する時間が短縮できる可能性がある利点があり、Peer Communities[7] のファイル検索の高速化を目的とした研究とは違う。

6 まとめ

本稿では、P2P 型ファイル検索において最適なピアを自動的に選択する手法を提案した。P2P 型ファイル検索では、検索にヒットした複数のピアからファイルの取得先ピアを選択しなければならず、通常、この選択は手動で行われている。そのため、必ずしもファイル取得時間が最短となるピアが選択できるとは限らない。

本稿で示した手法では、各ピアが実測するスループット情報と他のピアが実測したスループット情報とを交換し、ファイル取得時の実効スループットが低いと予測されるピアをファイル取得先候補から段階的に排除する。スループット情報の交換は通常のファイル検索メッセージを用いて行われるため、本方式を組み込んでも通信回数が増加しない。シミュレーションによる検証の結果、30 回程度のファイル検索、5 回程度のファイル取得を行えば、80% 以上の確率で最適なピアが選択できることがわかった。

今後の課題として、検索やファイル取得のメッセージの送信間隔の調整を行った上で、アルゴリズムの検証を行う必要がある。実際の環境に近いネットワークの構築や P2P アプリケーション以外の通信

によりネットワークにかかる負荷を考慮したシミュレーションや、実際のネットワーク上で実験をする点である。

参考文献

- [1] The gnutella protocol specification v0.4
http://www.stanford.edu/class/cs244b/gnutella_protocol_0.4.pdf.
- [2] napster. <http://www.napster.com/>.
- [3] The network simulator ns-2
<http://www.isi.edu/nsnam/ns/>.
- [4] Winmx. <http://www.winmx.com/>.
- [5] I. Clarke, O. Sandberg, B. Wiley, and T. W. Hong. Freenet: A distributed anonymous information storage and retrieval system. *Lecture Notes in Computer Science*, 2009:46–66, 2001.
- [6] J. D. Guyton and M. F. Schwartz. Locating nearby copies of replicated internet servers. *ACM SIGCOMM*, pp. 288–298, 1995.
- [7] M. Khambatti, P. Dasgupta, and K. Ryu. Peer-to-peer communities: Formation and discovery. *Fourteenth IASTED International Conference on Parallel and Distributed Computing and Systems Cambridge*, pp. 166–173, November 2002.
- [8] A. Myers, P. Dinda, and H. Zhang. Performance characteristics of mirror servers on the internet. *IEEE INFOCOM'99*, 1:304–312, March 1999.
- [9] S. Ratnasamy, M. Handley, R. karp, and S. Shenker. Topologically-aware overlay construction and server selection. *IEEE INFOCOM'02*, June 2002.
- [10] P. Reynolds and A. Vahdat. Efficient peer-to-peer keyword searching. *ACM/IFIP/USENIX International Middleware Conference*, 2003.
- [11] B. Yang and H. Garcia-Molina. Efficient search in peer-to-peer networks. *IEEE Int'l Conf. on Distributed Computing Systems*, 2002.