

## Peer-to-Peer ファイル共有システムにおける最近傍制御方式の提案

南 裕也      亀井 聡      斎藤 洋

NTT サービスインテグレーション基盤研究所

〒180-8585 東京都武蔵野市緑町 3-9-11

光アクセス系のキラーアプリケーションとして、Peer-to-Peer (P2P) ファイル共有システムが注目されている。P2P ファイル共有システムでは、利用が進むにつれノード間でファイルが複製されて行くため、目的のファイルを取得する際に通信相手となり得るノードが IP 網上で広域に分散配置されることになる。このような状況では、利用者の利便性を向上させる点からも網負荷の増大を抑制する点からも、通信相手に「近い」ノードを選ぶことが望ましい。本稿では、目的のファイルを保有するノードが複数ある場合にノードの「近さ (proximity)」に基づいて通信相手を選択する最近傍制御方式を提案し、シミュレーションを通して提案方式の有効性を確認する。

### A Proposal of Node Selection Based on Proximity for Peer-to-Peer File Sharing Systems

Hiroya Minami Satoshi Kamei Hiroshi Saito

NTT Service Integration Laboratories

3-9-11 Midori-Cho, Musashino-Shi, Tokyo 180-8585 Japan

Peer-to-Peer (P2P) file sharing systems have gained attention as a killer application on the optical access network. In this kind of systems, files are replicated among the nodes that join the systems. Under such conditions, every node should select nearby node which has a target file as a server of the file. In this paper, we propose a node selection mechanism based on proximity. And we evaluate the effect of the proposed mechanism through a simulation.

#### 1. はじめに

近年、常時接続かつ広帯域なアクセス回線の普及とエンドユーザの端末性能の向上により、Web 等のクライアント/サーバシステムにおいて特定のサーバへ負荷が集中する問題が顕在化している。これに対する一つの解決策として、Peer-to-Peer (P2P) が注目されている。P2P はシステムに参加する各ノードが同等の役割を担い相互に通信しあう分散システムである。P2P は、利用者が各自の端末資源の一部をシステムに提供するため、ノードの容量やアクセス帯域の点で十分なスケラビリティがあると言える。

現在は、Gnutella[1][2]やNapster[3]等のP2Pによる

ファイル共有システムがインターネットで流行している。このシステムは、エンドユーザによる情報発信を容易にし情報流通を促進するとして、双方向広帯域の光アクセス系のキラーアプリケーションとしても期待されている。しかし一方で、現状のP2Pファイル共有システムは、Webと比較してIP網上でより広範囲に大容量のファイルを送受しており、網への負担が大きいことが分かっている[4]。システムの安定運用を考える上で、利用者の増加に伴う網負荷の増大が懸念される場所である。

現状のP2Pファイル共有システムが網へ高い負担をかける原因の一つとして、IP網の負荷情報や経路情報を有効に使う仕組みが備わっていないことが考えられる。そこで本稿では、IP網上に構築されるP2Pファイル共有シ

システムを対象に、利用者の利便性向上と網設備の効率的利用のためノード間の「近さ(proximity)」に基づいて通信相手を選択する最近傍制御方式を提案する。

以下、2章で既存のP2Pファイル共有システムの概要を解説し、3章で提案する最近傍制御の枠組みについて説明する。4章では実現例として実装が容易でかつ有効性が高いと思われる方式を挙げ、シミュレーションを通して提案方式の有効性を確認する。最後に5章でまとめと今後の課題について述べる。

## 2. P2P ファイル共有システム

P2P ファイル共有システムでは、システムの参加者同士が互いの保有するファイルを参照し合う。各ノードはファイルを参照するにあたって、目的のファイルをシステム内で検索し、当該ファイルを保有するノードからダウンロードして複製する。本稿では、目的のファイルを検索して複製するノードを検索ノードと呼び、当該ファイルを既に保有しているノードを保有ノードと呼ぶ。

検索ノードは、目的のファイルの有無を各ノードに直接確認するか、システム内のファイルの所在を管理する中央サーバに問合せる。前者の代表例が Gnutella であり、後者の代表例が Napster である。

### 2.1. Gnutella

Gnutella 型のファイル共有システムは、ノード間で目的のファイルの有無を確認しあうために、IP 網の上にオーバーレイで構築されるアプリケーション層の独自ネットワークを介して、目的のファイルに関するメッセージを交換し合う。本稿ではこのファイル検索用の独自ネットワークを P2P ネットワークと呼ぶ。Gnutella で構築されている P2P ネットワークは GnutellaNet と呼ばれる。P2P ネットワークの構築は、ノード同士が TCP コネクションなど仮想コネクションの設定により論理的に接続されることで実現される。P2P ネットワーク上で接続された相手のノードを隣接ノードと呼ぶ。

ファイル検索を行うときは、検索メッセージを全ての隣接ノードへ送信する。検索メッセージは、隣接ノード以外のノードへも中継されて届く。各ノードは検索メッセージを受信すると、当該メッセージの送信元のノードへ回答を返信するとともに他の隣接ノードへ検索メッセージを転送する。個々の検索メッセージには、無限に転送が繰り返されることを防ぐための Time To Live(TTL)と重複して到着したメッセージの転送を回避するための識別子 Globally Unique Identifier(GUID)が与えられる。この TTL は、メッセージが P2P ネットワーク上で転送される回数を制限するものであり、IP 網を通過する際のルータホップ数を制限するものとは異なる。検索メッセージの転送回数が TTL により制限される

ため、ファイル共有は P2P ネットワーク上で検索メッセージが転送される範囲のノード間でのみ可能となる。回答のメッセージは、対応する検索メッセージの送信元ノードへの返信を繰り返しながら検索メッセージの転送経路を逆に辿り、最終的には検索メッセージを生成したノードに返される。検索メッセージの転送経路を逆に辿る返信方法を send back という(図1)。

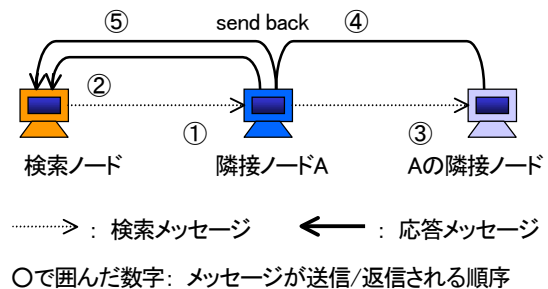


図1: P2P ネットワークによるファイル検索

各ノードはシステムへ参加するにあたり、既に P2P ネットワークに参加しているノードから一つ以上の任意のノードを選んで隣接ノードとする必要がある。Gnutella プロトコルでは、各ノードのシステムへの参加が断続的であることを考慮して、システムに参加中のノードを検索するメッセージも用意されている。ノード検索のメッセージは周期的に送信され、検索メッセージが転送される範囲にある参加中のノードを確認する。ノード検索のメッセージは上記の TTL および GUID の他に特に情報を運ばない。一方、これに対する回答のメッセージは、返信したノードの IP アドレスや接続を待ち受けるポート番号、アクセス回線速度の情報等を運ぶ。このノード検索の仕組みにより、各ノードはシステムへ参加している他のノードの一つ知っていれば、残りのノードを芋づる式に発見することができる。

ファイル検索のメッセージは目的のファイルについての条件を伝える。現状の殆どの実装では、具体的な検索条件は単純なキーワードで指定されている。Gnutella プロトコルでは、ファイル検索のメッセージに対する回答は、キーワードに一致するファイルの保有ノードだけが返信し、適当なファイルを保有しないノードは検索メッセージの転送のみを行う。回答メッセージには、返信したノードの IP アドレスとダウンロード時の接続を待ち受けるポート番号、条件に適合するファイル名等が記録される。

P2P ネットワークを利用したファイル検索では、目的のファイルの保有ノードが複数ある場合に、各保有ノードから別々に回答があることを利用して、回答が到着する順番に基づくノード選択が可能である。他にも Gnutella プロトコルでは、回答のメッセージに記録されている情報に基づいてダウンロードの相手を選択でき

る。回答メッセージが運ぶ有用な情報としては、アクセス回線速度やP2Pネットワーク上で回答メッセージが何回転送されたかのホップ数が挙げられる。Gnutella プロトコルに独自の拡張を施したソフトウェアも多数あり、その内の一つ Gnutella[5]は、アップロードの履歴情報が回答メッセージに添付している。

## 2.2. Napster

Napster 型のファイル共有システムは、システム内の情報の所在を一括管理する中央サーバを用意している。システムへの参加は、各ノードが中央サーバへ接続することで達成される。Napster では、中央サーバは実際には一つではなく、複数のサーバで負荷分散を行っている。そのため、ファイル共有は同じ中央サーバに接続しているノード間でのみ可能となっている。

各ノードはシステムへの参加が完了すると、自分が保有するファイルを中央サーバに登録するため、ファイル名やダウンロード時の接続を待ち受けるポート番号などファイル共有に必要な情報を中央サーバに送信する。

各ノードはファイル検索時に、目的のファイルのキーワードを中央サーバに送信する。これに対して中央サーバは、キーワードに一致するファイルとその保有ノードの IP アドレスおよび接続先のポート番号を回答する。

Napster には、中央サーバから案内された保有ノードに対して、ping で送られる ICMP echo パケット[6]の Round Trip Time(RTT)を測定する仕組みが備わっている。これにより、ファイル検索を実行した時点の保有ノードの可到達性が確認できるとともに、同一ファイルの保有ノードが複数ある場合に RTT に基づくノード選択が可能になっている。

ファイル検索に中央サーバを用いる Napster 型のシステムは、P2P ネットワークによる Gnutella 型のシステムに比べて検索に伴うメッセージの数を低く抑えられる。しかし一方で、システムの規模が中央サーバの容量で制限されることになり、また中央サーバに障害が発生した場合はファイル共有システム全体が停止してしまう。

## 3. 最近傍制御

P2P ファイル共有システムでは、システムの利用が進むにつれ、各ファイルはノード間で次々と複製されていく。これは同一のファイルの保有ノードが IP 網上で広域に分散配置されることを意味する。このような状況では、ファイル検索時に複数の保有ノードを発見し、その内の一つをダウンロードの相手を選ぶ場合は、利用者の利便性を向上させる点からも網設備を効率的に利用する点からも、IP 網上で「近い」ノードを選ぶことが望ましい。ノード間の proximity に基づいて最適なノードを選択する制御を本稿では最近傍制御と呼ぶ。

最近傍制御では、利用者及びネットワークの立場から、各々以下の事項が重要である：

- a) ダウンロードの所要時間を短縮すること
- b) ダウンロードに伴い無駄な網負荷が発生するのを抑制すること

具体的なノード選択方法は、探索ノードと保有ノード間の proximity を客観的に評価する尺度と機能配備により分類できる。

### 3.1. proximity の評価尺度

最近傍制御の効果はダウンロードの所要時間と網負荷によって決定される。このため、proximity の評価尺度には、これら2つの要因に対応したものが求められる。尺度の候補としては、実効スループット、網トポロジ上の距離、応答時間、アクセス回線速度、そしてこれらの組み合わせが考えられる。

#### 3.1.1. 実効スループットによる評価

ダウンロード時間を短縮するには、IP 網の通過経路の帯域や保有ノードの負荷状況が反映される実効スループットに基づくノード選択が有効である。そのためには、候補となる保有ノードからのダウンロードを試行して実効スループットを測定する必要がある。実効スループットは品質が保証されない IP 網では時間変動が大きい。そのため、測定はノード選択の直前に実行することが望ましい。P2P ファイル共有システムでは、回答メッセージが保有ノードから直接検索ノードに返されることがないため(Gnutella 型のシステムで全ての保有ノードが隣接ノードである場合を除く)、試験メッセージを別途ダウンロードすることになる。

ノード選択時に実効スループットの測定を逐次実行する場合、利用者の利便性が悪化するのを避けるため、測定の処理遅延を短く抑えなければならない。そのためには、試験メッセージのサイズはできる限り小さくしたいという要求がある。しかし、ダウンロードに TCP を用いるシステムでは、ウィンドウ制御の特性上、試験メッセージのサイズを数百 KBytes 程度とする必要がある[7]。実効スループットによる評価には測定負荷が大きくなる欠点があると言える。

なお、回答メッセージが保有ノードから直接検索ノードへ返されるシステムは、検索ノードの側で仮想コネクションの数が急激に増加し、検索ノードの過負荷を引き起こす可能性もあるため好ましくない。

#### 3.1.2. 網トポロジ上の距離による評価

ダウンロードに伴う網負荷は、取得するファイルのサイズが同じであっても、ノード間の網トポロジ上の距離によって増減する。網トポロジ上の距離は、ダウンロードパケットが IP 網を通過する経路のルータホップ数や

經由 AS 数等で表される。ルータホップ数が最小のノードを常に選択する場合、IP 網の各リンクの負荷が平均的には最少化され、網全体の負荷が低減される。IP 網の経路は網の負荷状況に応じて変化するため、経路情報の変化を常に監視しておくか、ノード選択時に網トポロジ上の距離を測定する必要がある。

ルータホップ数は traceroute[8] という一般的な測定ツールによって測定可能であるが、セキュリティ等の理由でこのツールに回答しないネットワークも多い。確実な測定のためには、P2P ファイル共有システムに独自のルータホップ数測定機能を実装し、アプリケーション層の通信として測定することが望ましい。4.1 節で、システムに測定機能を組み込む場合のルータホップ数測定法を具体的に説明する。

web では、BGP 等の経路制御プロトコルから得られる情報により經由 AS 数を把握しサーバ選択に利用する方式も見られる[9]。しかしながら、經由 AS 数はサーバ選択の尺度としては粒度が粗く不十分であることが指摘されている[10][11]。IP 網の経路制御プロトコルから得られる情報によりノードの proximity を適切に判断する手法が求められる。

Gnutella 型では、P2P ネットワーク上のホップ数も利用できる。しかし、これは IP 層の網トポロジ上の距離と全く異なるため、最近傍制御において特に参考になるものではない。

### 3.1.3. 応答時間による評価

Napster で採用されているように、ping の RTT から IP 網の通過経路の負荷状態を判断する方法は実装が容易である。しかし、セキュリティ等のため ICMP パケットをフィルタリングするネットワークもあるため、RTT を確実に測定するには、やはり P2P ファイル共有システムに独自の応答時間測定機能を実装し、アプリケーション層の通信の中で測定する必要がある。

Gnutella 型では、ファイル検索時に各保有ノードから別々に回答メッセージが届くため、この応答時間が利用できる。ただし、回答メッセージが send back で IP 網を通過する経路はダウンロードパケットが通過する経路とは異なることに注意しなければならない。

ノード選択において応答時間から判断できることは、パケットの通過経路に過負荷状態のノードあるいは網設備があるか否かという程度である。

### 3.1.4. アクセス回線速度による評価

アクセス回線が性能のボトルネックとなる環境では、アクセス回線速度に基づくノード選択も有効である。アクセス回線速度は固定値であるから、ノード選択時に測定を行う必要がない。しかし、ネットワークの負荷状況を反映しないため、最適なノードを選択するには情報と

して不十分である。また、現状では、アクセス回線速度を利用するにはエンドユーザによる手入力が必要であり、アクセス回線の変更などにより入力値と実際の速度の不整合が起きやすいという問題がある。

### 3.1.5. 総合的な評価

これまでに挙げた proximity の評価尺度を個別に扱うのではなく、総合的に評価して最適なノードを選択する仕組みも有用である。種類の異なるデータを同時に扱うには、各々の値を規格化する必要がある。

任意の尺度  $x$  による保有ノード  $i$  の評価値を  $X_i$  で表すことにする。各保有ノードの評価値の分布が正規分布に従うと仮定すると、 $X_i$  の値は次の関数で規格化できる。

$$f(X_i) = (X_i - mx)/\sigma x$$
ここで  $mx$  は一回のノード選択で候補となる全ての保有ノードの尺度  $x$  の評価値の平均、 $\sigma$  は標準偏差である。

このように評価値を規格化した上で、各々の尺度の重要度に応じて重み付けして加算することで総合的な評価ができる。たとえば、保有ノード  $i$  の proximity を 2 種類の評価尺度を用いて評価し、各々の尺度による評価値が  $A_i$  と  $B_i$  である場合、総合評価値  $C_i$  は以下のように決定できる。

$$C_i = \alpha \cdot f(A_i) + \beta \cdot f(B_i), (\alpha + \beta = 1)$$
各尺度の重要性を表す  $\alpha$  および  $\beta$  の値は利用者またはファイル共有システム単位で定められる。

## 3.2. 機能配備

最近傍制御では、候補となる保有ノードの proximity を正確に評価するために測定を行う機能（以下、proximity 測定機能）と、各保有ノードの proximity 評価の結果を比較して最適なノードを選択する機能（以下、ノード選択機能）が P2P ファイル共有システムに組み込まれる。ここでは、これらの機能配備により制御方式を分類する。

### 3.2.1. proximity 測定機能

proximity 測定機能が配備される候補としては、検索ノード、保有ノード、中央サーバが挙げられる。proximity の測定では、測定対象となる保有ノードの IP アドレス以外にアプリケーション層で扱う知識を必要としないため、中央サーバのかわりに P2P ファイル共有システムの外部にある第三者を利用しても良い。

#### 3.2.1.1. 検索ノードによる測定

proximity の評価は、同一ファイルの保有ノードが複数発見された時に候補となる保有ノードに対してのみ行えば良い。検索ノードはファイル取得先の候補となる保有ノードの所在が把握できるので、検索ノードに測定機能を配備する制御方式は、測定対象を必要最低限に絞ることで測定負荷を低く抑えられる。

### 3.2.1.2. 保有ノードによる測定

保有ノードの側で、回答メッセージを返信する直前に測定を行う方式も考えられる。測定結果は回答のメッセージに添付されて送られる。この方式では、ノード選択時の測定負荷は分散されるが、候補となるか否かに拘らず保有ノードが常に測定を実行することになり、無駄な測定負荷が発生してしまう。

### 3.2.1.3. 中央サーバによる測定

ノード間の proximity を中央サーバまたは第三者が測定することにより、利用者の端末は proximity 測定の負荷から解放される。

複雑なネットポロジを持つ IP 網では、任意の 2 ノードを通過する経路の実効スループットやルータホップ数等を他の地点から測定するのは非常に困難であるため、proximity 測定機能を中央サーバあるいは第三者に配備する場合は、評価尺度として IP 網の経路制御プロトコルから得られる情報を利用する方法が有効である。

### 3.2.1.4. proximity 測定機能が不要の場合

アクセス回線速度や履歴情報のように測定を必要としない評価尺度のみを用いる場合は、proximity 測定機能が不要となる。

## 3.2.2. ノード選択機能

ノード選択時の処理遅延と処理負荷を低減するため、ノード選択機能の配備先としては以下の知識を持つものが望ましい。

- ・ファイル毎の保有ノードの数および所在
- ・各ノードの proximity 評価の結果

両方の知識を持ち得るのは、ファイル検索の仕組みから、Gnutella 型では検索ノードのみが、Napster 型では検索ノードと中央サーバが該当する。ノード選択の処理は、ノード毎の proximity 評価の優劣を単純に比較するだけなので非常に軽く済む。そのため、ノード選択機能を配備することによる検索ノードあるいは中央サーバの負担は小さいと言える。

### 3.2.2.1. 検索ノードによる選択

検索ノードにノード選択機能を配備する場合、proximity 評価に必要な測定結果をノード内で受け渡しするために、測定負荷が問題とならない限りは proximity 測定機能も検索ノードに配備すべきである。

### 3.2.2.2. 中央サーバによる選択

中央サーバにノード選択機能を配備する方式においても、proximity の評価尺度が中央サーバで測定可能なものに限定できるのであれば、proximity 測定機能を中央サーバに配備すべきである。

Napster 型のシステムで、proximity 測定機能とノード選択機能を中央サーバに配備する場合は、利用者の端末側に新たな機能を追加する必要がなく、最近傍制御を利用者に透過的に提供することが可能になる。

### 3.2.2.3. ノード選択機能が不要の場合

既存の P2P ファイル共有システムのように、利用者が最適なノードを判断する場合は、検索ノードが必要な情報を収集するだけでよく、ノード選択機能が不要となる。

## 3.3. 方式比較

最近傍制御の各方式を、既存のファイル共有システムで利用者に提示されている情報に基づくノード選択法と定性的に比較し整理する(表1)。

	制御方式				
	A	B	C	D	E
ダウンロード時間短縮	◎	△	△	△	△
網負荷の低減	△	◎	○	△	×
測定負荷	×	○	○	○	—
処理遅延	△	○	○	—	—
実現容易性	容易	容易	困難	既存	既存

表1：方式比較

表中の A ~ F に対応する各方式における proximity の評価尺度と proximity 測定機能の配備先の組み合わせは以下の通りである：

- A) 実効スループット・検索ノード測定
- B) ルータホップ数・検索ノード測定
- C) 経路制御プロトコル情報・中央サーバ測定
- D) RTT・検索ノード測定
- E) アクセス回線速度・測定不要

上の A ~ C では、proximity 測定機能とノード測定機能は同じノードに配備されるとした。評価尺度と各機能の配備先の組み合わせにより他にも方式が考えられるが、表には有効性が高いと思われるもののみを示した。

## 4. 実現例

最近傍制御の実現例として、保有ノードの proximity をルータホップ数のみで評価し、proximity 測定機能およびノード選択機能を検索ノードに配備する方式を詳細に説明する。本方式は、表1の方式Bに対応するものである。

### 4.1. ルータホップ数測定の実現

本方式では、アプリケーション層の通信の中でルータホップ数を測定する独自機能を P2P ファイル共有システムに組み込む。本方式のルータホップ数測定は以下のように行う(図2)。

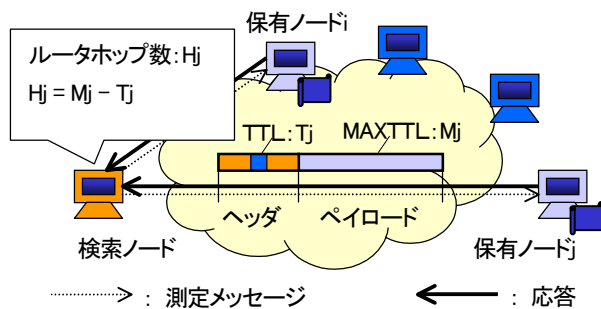


図2：ルータホップ数測定法

- 1) 候補となる保有ノードへ測定メッセージを出す。
- 2) 測定メッセージを受けた保有ノードは応答を返すが、その際、送出パケットのIPヘッダのTTLフィールドに設定するMAXTTLの値をペイロード部にも記録する。
- 3) 応答のパケットを受信した検索ノードは、ペイロードのMAXTTLの値とIPヘッダ内のTTL(ルータを経由する毎に1ずつ減少する)の値の差分からルータホップ数を求める。

#### 4.2. 制御フロー

最近傍制御は同一ファイルの保有ノードが複数あることが検出された場合に実行すれば良いが、具体的な実行契機はGnutella型とNapster型で異なる。これは、Gnutella型では、各ファイルの保有ノードを全て網羅するのに、どれだけの回答メッセージを受信すれば良いかわからないためである。本方式は、Gnutella型では、以下の何れかの条件が満たされた時点で実行を開始する：

- A) 検索メッセージを送信後、一定時間経過する
- B) 候補とする保有ノードの数に制約を設け、発見された保有ノードの数がファイル毎に閾値を超えたとき

一方、Napster型では単純に、中央サーバの回答メッセージから同一ファイルの保有ノードが複数あることが検出されたときに実行を開始する。

実行開始後の制御フローを以下に示す。

1. 候補となる保有ノードのIPアドレスをproximity測定機能に渡す
2. proximity測定機能は前節のルータホップ数測定法の通りに測定を実行する。
3. proximity測定機能は各候補のルータホップ数をノード選択機能に渡す
4. ノード選択機能は、ファイル毎に、候補となる保有ノードからルータホップ数が最小のものを選択する。ルータホップ数最小のノードが複数ある場合は、ランダムに一つ選ぶ。

#### 4.3. 評価

実現例として挙げたルータホップ数が最小の保有ノ

ードを選択する方式(提案方式)を、NapsterのようにpingのRTTが最短の保有ノードを選ぶ方式(従来方式)と比較した場合のルータホップ数削減効果を評価するシミュレーションを行った。シミュレーションでは、入力パラメータとして実際のP2Pネットワークから得た情報を使用した。

##### 4.3.1. シミュレーションの手順

シミュレーションに先立ち、現状で最大級のP2PネットワークであるGnutellaNetを観測した。具体的には、国内でインターネットに接続した端末において、Gnutellaプロトコルの処理にパケットをダンプする機能を追加したソフトウェアであるgnutを起動し、通過するパケットを24時間キャプチャした。さらに、GnutellaNet上に現れたノードに対してtracerouteとpingを実行し、ルータホップ数とRTTを測定した。

シミュレーションでは、GnutellaNetの観測結果をそのまま使用した。ただし、ルータホップ数については、主に北米に位置しているノードを国内から観測したため、北米に到達するまで10ホップ程度必要となることを考慮し、実測値から10減算することにした。

今回のシミュレーションでは、以下に示す方法でノード選択を10万回模擬した。

- 1) ノード選択の候補となる保有ノードとしてGnutellaNet上に現れたノードからランダムにN個を選択する。
- 2) Nはポアソン分布に従う確率変数とし、ノード選択を模擬する度に値を決定する。分布の平均には任意の値を与える。
- 3) 選択されたN個のノードを対象に、ルータホップ数またはRTTが最小のノードを選択する。

##### 4.3.2. 結果

図3は保有ノード数の分布Nの平均を40程度としたときの数値例である。

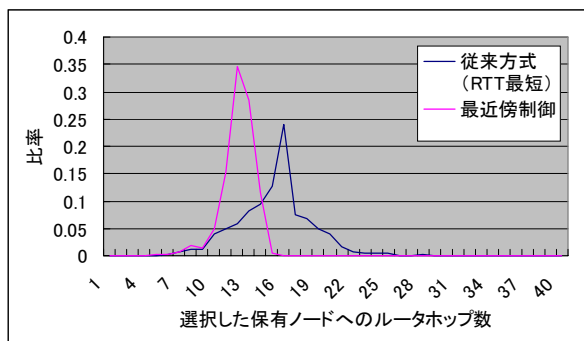


図3：最近傍制御の効果

提案方式により、選択したノードへのルータホップ数が著しく減少することが分る。

図4は分布の平均を4程度としたときの結果である。

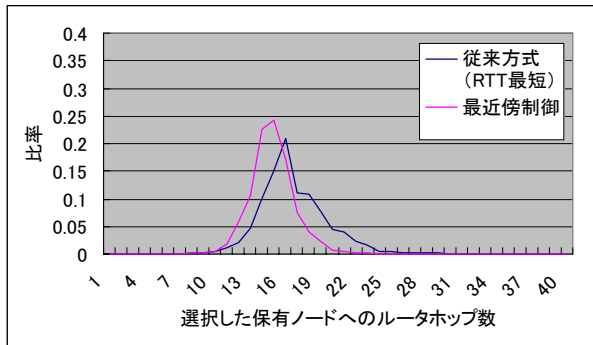


図4：候補の数が少ないときの最近傍制御の効果

この場合、ルータホップ数は従来方式に比べ若干減少する程度である。この結果から、最近傍制御は候補の数があがる程度多く、ノード選択の幅が広い場合に効果が顕著に現れることが分る。ただ何れにせよ、従来方式に比べてルータホップ数を減少させており、提案方式の有効性が確認できた。

## 5. まとめと今後の課題

本稿では、P2P ファイル共有システムを対象に、目的のファイルの保有ノードが複数ある場合に、ノード間の proximity に基づいてダウンロードする相手を選択する最近傍制御方式を提案した。さらに、実現例として proximity をルータホップ数のみで評価する方式を挙げ、実際のP2Pネットワークから得た情報を入力パラメータとしたシミュレーションを通して提案方式の有効性を確認した。

今後の課題としては、proximity の評価尺度にルータホップ数以外を用いた場合の効果を定量化することや、評価シミュレーションの中で候補となる保有ノードの数を決めている確率分布をより妥当なものに置き換えることなどが挙げられる。

## 6. 参考文献

- [1] Gnutella. <http://gnutella.wego.com>.
- [2] 井口圭一, "ネットワーク管理者のためのGnutella入門", [http://www.atmarkit.co.jp/fwin2k/experiments/gnutella\\_for\\_admin/gnutella\\_for\\_admin\\_0.html](http://www.atmarkit.co.jp/fwin2k/experiments/gnutella_for_admin/gnutella_for_admin_0.html), 2000.
- [3] 井口圭一, "ネットワーク管理者のためのNapster入門", [http://www.atmarkit.co.jp/fwin2k/experiments/napster\\_for\\_admin/napster\\_for\\_admin\\_1.html](http://www.atmarkit.co.jp/fwin2k/experiments/napster_for_admin/napster_for_admin_1.html), 2000.
- [4] 亀井聡, 木村卓巳, "大規模P2Pネットワークの測定と考察", 情報処理学会第63回全国大会, 2001.
- [5] Gnutella. <http://www.gnutella.com/>.
- [6] J. Postel, "INTERNET CONTROL MESSAGE PROTOCOL", RFC792, 1981.

[7] 上山憲昭, 山田慈朗, "IP 網における実効スループット簡易測定手法に関する一考察", 電子情報通信学会ソサイエティ大会, 2001.

[8] traceroute. <ftp://ftp.ee.lbl.gov/pub/traceroute.tar.Z>.

[9] Kevin Delgadillo, "Cisco DistributedDirector", [http://www.cisco.com/warp/public/cc/pd/cxsr/dd/tech/dd\\_wp.htm](http://www.cisco.com/warp/public/cc/pd/cxsr/dd/tech/dd_wp.htm), 2000.

[10] Kirk L. Johnson, John F. Carr, Mark S. Day and M. Frans Kaashoek, "The Measured Performance of Content Distribution Networks", 5th International Web Caching and Content Delivery Workshop, 2000.

[11] Balachander Krishnamurthy and Jia Wang, "Topology Modeling via Cluster Graphs", ACM SIGCOMM Internet Measurement Workshop, 2001.