

P2P ネットワークにおけるデータアクセス頻度を考慮した更新伝播法

渡辺 俊貴^{†1} 神崎 映光^{†1}
原 隆浩^{†1} 西尾 章治郎^{†1}

筆者らの研究グループではこれまでに、データが複製され、それぞれのデータに更新が発生する P2P ネットワークにおいて、更新伝播時の負荷分散と遅延減少を目的とした木構造に基づく更新伝播法を提案している。本論文では、これまでに提案した更新伝播法を拡張し、各ピアのデータのアクセス頻度に応じて伝播させるデータを変更する手法を提案する。提案手法では、データのアクセス頻度の高いピアには、更新データを伝播させ、データのアクセス頻度の低いピアには、複製が古くなったことのみを通知する小さなメッセージを送信する。これにより、更新データを伝播させるピア数を削減し、更新伝播時の負荷や遅延を抑えることができる。

An Update Propagation Strategy Considering Data Access Frequency in Peer-to-Peer Networks

TOSHIKI WATANABE,^{†1} AKIMITSU KANZAKI,^{†1}
TAKAHIRO HARA^{†1} and SYOJIRO NISHIO^{†1}

We have proposed an update propagation strategy using a tree structure for load balancing and delay reduction assuming a P2P network in which each data is replicated. In this paper, we extend our previous strategy to propagate update information selectively considering the data access frequency of each peer. The extended strategy propagates the updated data to peers which frequently access the data, whereas it propagates only a small message informing that the replica has become invalid to peers which rarely access the data. This approach further reduces the load and delay for update propagation since the number of peers that receive the updated data decreases.

1. はじめに

近年、計算機の高性能化やネットワークのブロードバンド化により、Peer-to-Peer (P2P) ネットワークを用いたデータ共有に関する研究が注目されている。ピアどうしが対等な接続形態をとり、互いにサービスを提供しあう P2P ネットワークでは、検索効率やデータ可用性の向上、負荷分散のためにデータを複製し、ネットワーク上の複数のピアに配置することが有効である^{4),5)}。一方、リアルタイムの天気予報やニュース速報、分散 Web コンテンツなどの分散ファイルシステムサービスでは、複数のピアで共有しているデータに更新が発生する環境が想定される。データに更新が発生した場合、複製を所持するピアに対して、即座に更新情報を通知する必要がある。筆者らの研究グループではこれまでに、更新伝播時の負荷分散と遅延減少を目的として木構造に基づく複製更新伝播法 (UPT-FT 法: Update Propagation Tree with Fault Tolerance) を提案している¹⁶⁾。この手法では、データのオリジナルを所持するピア (オリジナルノード) を根とし、複製を所持するその他のピアを内部節点とした n 分木の論理ネットワーク (更新伝播木) を構成し、この更新伝播木に沿って更新データを伝播する。

ここで、複製を所持するピアの中には、頻繁にそのデータにアクセスするピアと、あまりそのデータにアクセスしないピアが混在することが考えられる。UPT-FT 法は、これらのピアに同等に更新データを伝播するため、あまり最新のデータを必要としないピアにまで更新データを伝播し、余分な負荷が発生してしまう。

そこで本論文では、UPT-FT 法を拡張し、各ピアのデータへのアクセス頻度に応じて更新データを伝播させるピアを変更する手法として、UPT-HL 法 (UPT with H & L peers) を提案する。UPT-HL 法では、データへのアクセス頻度が高いピアにのみ更新データを伝播させ、アクセス頻度の低いピアには、データが古くなったことを通知する小さなメッセージ (無効化情報) を送信する。これにより、更新データを伝播させるピア数を削減し、更新伝播時の負荷や遅延を抑えることができる。

以下では、2 章で関連研究について述べ、3 章で、UPT-HL 法について説明する。4 章で UPT-HL 法の性能評価を行い、最後に 5 章で本論文のまとめと今後の課題について述べる。

^{†1} 大阪大学大学院情報科学研究科マルチメディア工学専攻

Department of Multimedia Engineering, Graduate School of Information Science and Technology, Osaka University

2. 関連研究

P2P ネットワークサービスに関する研究分野では、データの複製を配置する研究はさかんに行われている^{(6),(8),(10),(11)}が、複数のピアで共有されるデータに更新が発生する環境を考慮した研究は、あまり行われていない。しかし実環境では、データに更新が発生することが一般的である。このような環境では、複製を所持するピアに対する更新データの通知が遅れると、更新前の古いデータにアクセスしてしまう可能性が高くなる。そのため、複製を所持するすべてのピアに対して、即座に更新データを通知する機構が必要となる。以下では、提案手法と類似したアプローチをとるマルチキャストに関する研究、および、従来研究における複製更新伝播法について、その概要と問題点について述べる。

2.1 木構造を用いたマルチキャスト

P2P ネットワークにおいて、木構造を用いたマルチキャストに関する研究もさかんに行われている^{(7),(13),(14)}。これらの研究では、データが木に沿って伝播される点で提案手法と類似している。しかし、これらの研究では、木に参加する各ピアが多くの情報を管理し、各ピアに付随している子ノードの数や位置などを考慮してピアの参加位置を決定するなど、よりバランスのとれた最適な木を構築、維持することを目的としている。このような手法では、木構造が変化した場合に、各ピアが管理する情報を更新するために多くのメッセージ交換が行われる。そのため、複製が頻繁に置き換えられ、更新伝播木への参加や脱退が繰り返されるような本論文の想定環境には適していない。一方、提案手法では、各ピアが少数の情報を管理し、単純な操作を行うことで、可能な限りバランスのとれた木を構築、維持している。

2.2 複製（インデックス）の配置を考慮した更新伝播

文献 12) で提案されている CUP 法では、クエリを発生させたピアからクエリに回答したピアまでのパス（クエリパス）上にあるすべてのピアに複製（インデックス）を配置し、更新が発生した場合には、クエリパスに沿って更新情報を伝播させる。また、CUP 法を改善した DUP 法⁽¹⁷⁾では、パス上のすべてのピアにはインデックスを配置せず、パス上にあるピアを選別してインデックスを配置することにより、更新情報を經由させるピアを減少させ、更新伝播にかかる遅延を小さくしている。これらの手法は、複製（インデックス）の配置と更新伝播を組み合わせることで、更新を伝播すべきピアの管理を容易にしている。しかし、複製の配置数や配置場所、さらに更新を伝播させる経路を柔軟に変更できない。

2.3 構造型 P2P ネットワークにおける更新伝播

Freenet⁽³⁾では、構造型 P2P ネットワークにおいて、コンテンツから生成されるハッシュ

キーをもとにデータの検索を行うという特徴を利用し、効率的な更新データの伝播を実現している。各ピアは、隣接ピアのアドレスとそれに対応するキーのリストを所持しており、自身が要求するデータのキーとより近いピアにクエリを伝播することにより、対象のデータを所持しているピアを効率的に発見することができるため、更新の伝播も容易に行うことができる。しかし、複製を所持するピアが確実に更新データを受け取ることができるという保証はない。

文献 2) では、DHT を利用した構造型 P2P ネットワークにおいて、ID 空間を分割し、それぞれの空間を代表するピアを節点とする木を構築し、更新情報を伝播させている。しかし、構造型 P2P ネットワークを想定していることから、構造の組み換えなどを柔軟に行うことが困難である。

また、これらの手法はともに、各ピアのデータアクセス頻度を考慮して更新を伝播させないため、更新伝播時の負荷が大きくなる。

2.4 非構造型 P2P ネットワークにおける更新伝播

本論文と同様に、非構造型 P2P ネットワークにおける更新伝播に関する研究もいくつか行われている。

2.4.1 確率的な更新伝播法

Datta らは、P2P ネットワーク上の複製を所持するピアに対して、確率に基づいて更新データを伝播させる手法を提案している⁽⁵⁾。この手法では、複製を所持する各ピアは、同じ複製を所持するピアの情報をランダムに保持する。データを更新したピアは、同じデータの複製を所持するピアのうちいくつかを隣接ピアとし、それらのピアに対してある確率（更新伝播率）で更新データを伝播させる。1 度更新を伝播されたピアは、部分リストと呼ばれるリストに加えられ、更新データと一緒に隣接ピアに通知される。

更新データを受け取ったピアは、リストに存在しない隣接ピアに対して、更新伝播率に基づいて更新データを伝播させる。この更新伝播率の値は、更新発生元ピアからの論理ネットワーク上のホップ数が大きくなるにつれて小さくなる。つまり、更新データが多くのピアに伝わるにつれて、隣接ピアに更新データを送る確率が低くなる。しかしこの手法では、すべてのピアに更新が伝播される保証がない。さらに、1 つのピアに対して複数の経路を通して更新が伝播されることがあり、余分な負荷が発生する可能性がある。

2.4.2 チェイン構造を用いた更新伝播法

Wang らは、複製を所持するピアを一直線のチェイン上に配置する論理ネットワークを形成することにより、更新伝播時のオーバーヘッドを抑えつつ、ネットワーク耐性の向上を実現

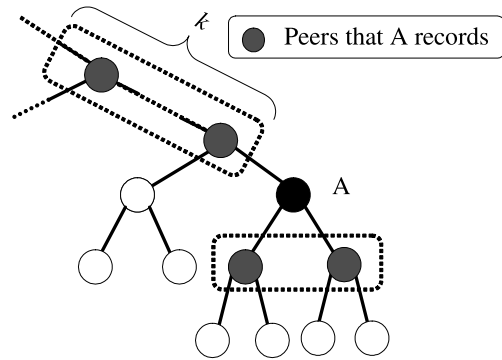


図 1 更新伝播木上のピアの管理情報

Fig. 1 Information managed by a peer in an UP tree.

する手法を提案している¹⁵⁾。この手法では、複製を所持する各ピアは、左右 m 個ずつのピア情報 (IP アドレスなど) を自身の調査ノードとして保持する。チェーン上のピアで更新が発生した場合、更新発生元ピアは、左右 m 個の調査ノードに更新データを送信する。それぞれの方向の m 個の調査ノードのうち、最も遠くにあるネットワーク上に存在する (オンライン) ノードを更新伝播責任ノードとし、更新データが送られてきた方向とは反対方向の調査ノードに対して、同様に更新データを伝播し、次の更新伝播責任ノードを決定する。以上の操作を、左右両方向のピアに対して繰り返すことにより、チェーン上のすべてのオンラインノードに更新データを伝えることができる。しかしこの手法では、複製を所持するピアを一直線上に並べるため、ピア数が増えると、すべてのピアに更新データを伝播させるまでの遅延時間が長くなってしまふ。また、更新データを多くのピアに送信するピアとまったく送信しないピアが現れ、更新伝播時の負荷の偏りが大きくなってしまふ。

2.4.3 木構造に基づく複製更新伝播法 (UPT-FT 法)

筆者らの研究グループではこれまでに、更新伝播時の負荷分散と遅延減少、および障害耐性向上を目的とし、木構造に基づく複製更新伝播法 (UPT-FT 法) を提案した¹⁶⁾。この手法では、検索に用いるネットワークとは別に、各ピアが所持するデータごとに、オリジナルノードを根とし、複製を所持するピアを内部節点とした n 分木の論理ネットワーク (更新伝播木) を形成し、この更新伝播木に沿って更新データを伝播させる。これにより、更新伝播時の負荷分散と遅延減少を両立させる。また、図 1 のように、更新伝播木に参加する各ピアが、親の方向に対して k 個分の先祖ノード、および、子ノードの情報 (IP アドレスなど)

を管理し、新規ピアの更新伝播木上での参加位置の決定や、複製削除などともなうピアの脱退時における更新伝播木の再構成を自律分散的に行う。なお、 k は 1 以上の整数である。

この手法では、ネットワーク障害や機器の故障などによる、周辺ピアへの通知のない退出 (不当な退出) を考慮している。一般に、ピアがネットワークから不当に退出した場合、更新伝播木の修復を行うことができず、更新データをすべてのピアに伝播することができなくなる可能性がある。更新伝播木が分断した場合に、検索用の論理ネットワークを用いてクエリをフラッディングさせ、データ (複製) を所持するピアを発見し、更新伝播木に再参加させる方法も考えられるが、大きなトラフィックが発生してしまう。そこで、この手法では、管理している k 個分の先祖ノードの情報を用いて、更新伝播木の分断が発生したときに、分断した箇所の周辺のピアのみで更新伝播木を修復する。そのため、少ないトラフィックで、最大 $k-1$ 個までの連続した先祖ノードの不当な退出まで修復が可能である。

3. データアクセス頻度を考慮した更新伝播法 (UPT-HL 法)

UPT-FT 法は、更新伝播時の負荷分散と遅延減少を実現し、高いネットワーク耐性を備えている。この手法では、データに更新が発生した場合に、複製を所持するすべてのピアに更新データを伝播することを前提としている。しかし、複製を所持するピアの中には、そのデータへのアクセス頻度が小さく、必ずしも更新データを必要としないピアが存在することが考えられる。このようなピアに更新データを伝播させても、そのデータが利用されない可能性が高いため、更新データの伝播にかかる負荷が余分なものになってしまう。

そこで UPT-HL 法では、UPT-FT 法を拡張し、更新データを伝播させるピアを制限することにより、更新伝播時の負荷や遅延をさらに抑制する。

3.1 想定環境

本論文では、非構造型 P2P ネットワークにおいて、各ピアが、自身または他のピアが持つデータにアクセスする環境を想定する。また、各ピアは、他のピアが持つデータの複製を作成し、自身の記憶領域に配置する。さらに、検索に用いるネットワークとは別に、各データごとに更新伝播木が存在し、複製を配置したピアは、そのデータの更新伝播木に参加する。以降、本論文では、ある 1 つのデータに対する更新伝播木を対象とし、議論を進める。

また、議論の簡単化のため、ピアやネットワークの障害は発生せず、検索用のネットワークや更新伝播木からのピアの不当退出は発生しないものとする。

3.2 更新伝播木の構成

UPT-HL 法における更新伝播木の構成例を図 2 に示す。UPT-HL 法では、複製を所持す

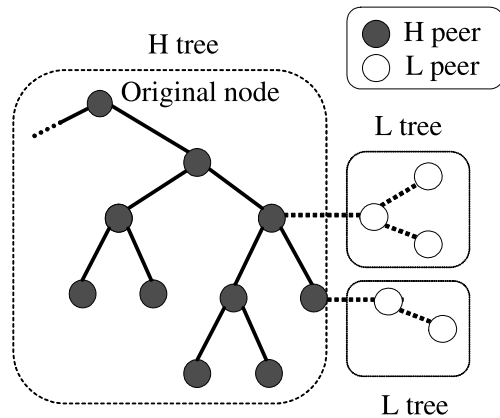


図 2 更新伝播木の構成
Fig. 2 Structure of an UP tree.

るピアを、データへのアクセス頻度が高く、つねに最新のデータを必要とするピア（Hピア）と、データへのアクセス頻度が低く、あまり最新のデータを必要としないピア（Lピア）に分類し、Hピアで構成される更新伝播木（H木）と、Lピアで構成される更新伝播木（L木）に分けて論理ネットワークを構成する。また、1つのデータに対して1つのH木を構成し、Lピアは各Hピアに付随する形で複数のL木を構成する。さらに、複製を所持する各ピアは、更新発生時に最新のデータの必要性を調査し、適宜Hピア、Lピアへの移動を行う。

3.3 ピアの管理情報

Hピアは、H木における自身の上位 $k (\geq 1)$ 個の先祖ノードと子ノード、および、自身に付随するL木の根ノードの情報を管理する。ここで、根ノードは自身より上位の先祖ノード情報は管理しない。また、先祖ノードの数が k 個未満の場合は、先祖ノード情報として根ノードまでのノード情報を管理する。一方、Lピアは、L木における自身の親ノードと子ノード、および、自身が参加するL木が付属しているHピア（責任Hノード）の情報を管理する。

3.4 更新情報の伝播

UPT-HL法における更新情報伝播の様子を図3に示す。データを更新したオリジナルノードは、H木に沿って更新データを伝播させる。更新データを受け取ったHピアは、H

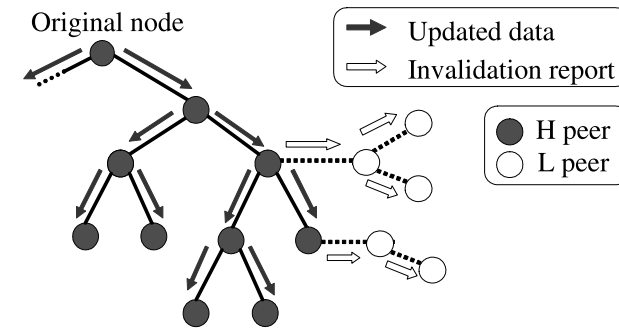


図 3 更新情報の伝播
Fig. 3 Update propagation.

木における自身の子ノードに更新データを送信する。また、自身に付随するL木の根ノードに無効化情報を送信し、データが古くなったことを通知する。無効化情報を受け取ったLピアは、自身の所持する複製を無効化し、L木における自身の子ノードに無効化情報を送信する。データを無効化したLピアが最新のデータを必要とする場合、自身の責任Hノードにデータを要求し、最新のデータを取得する。

このように、UPT-HL法は、更新データを伝播させるピアを制限することにより、更新伝播時の負荷や遅延を抑えることができる。

3.5 新規ピアの参加

新たにデータを取得したピアは、自身の記憶領域にその複製を作成し、そのデータの更新伝播木に参加する。このとき、新たに更新伝播木に参加するピア（新規参加ピア）は、まずHピアとしてH木に参加するものとする。新規参加ピアのH木への参加手順を以下に示す。

- (1) 新規参加ピアからのデータ要求を受け取ったピアがHピアであれば、そのHピアがクエリに応答する。データ要求を受け取ったピアが最新のデータを所持するLピアであれば、そのLピアがクエリに応答し、データと同時にそのLピアの責任Hノードの情報を返す。データ要求を受け取ったピアが無効化情報を受信したLピアであれば、そのLピアは自身の責任Hノードの情報を返す。ここで、新規参加ピアの参加位置の決定を行うピアを責任ノードと呼ぶ。最初は、クエリに応答したHピア、もしくはLピアから通知された責任Hノードが責任ノードとなる。
- (2) 責任ノードは、自身がオリジナルノード（H木の根ノード）であれば、手順(3)に進む。オリジナルノードでない場合、責任ノードは、H木における自身から k 個上

位の先祖ノード（先祖ノードが k 個未満の場合は根）にあたるピアに参加要求を送る．参加要求を受けたピアが新たな責任ノードとなり，手順 (3) に進む．

- (3) 責任ノードは，自身の子ノードの数 x を調べ， $x < n$ の場合は，新規参加ピアをその責任ノードの子として参加させる．その後，新規参加ピアの親ノードとなるピアは，自身の子に関する情報として，新規参加ピアを追加する．一方，新規参加ピアは，親ノードの情報を受け取り，自身の親ノードの情報として記録する．さらに， $k > 1$ の場合は，親ノードから $k - 1$ 個上位まで（先祖ノードの数が $k - 1$ 個未満の場合は根まで）の先祖ノードの情報も同時に受け取り，自身の先祖ノードの情報として記録する． $x = n$ の場合，手順 (4) に進む．
- (4) 責任ノードは，自身の子ノードの中からランダムに 1 つを選択し，その子ノードを新たな責任ノードとする．新たに責任ノードとなったピアは，(3) 以降の手順に従って，新規参加ピアの更新伝播木における参加位置を決定する．

ここで，手順 (2) において， k 個上位の先祖ノードに参加要求を送る（より根に近いピアを最初の責任ノードとする）ことで，更新伝播木を完全 n 分木に近づけることができる．これにより，更新伝播木の高さが減少し，更新伝播時の平均遅延が小さくなる．

3.6 ピアの脱退

複製の置き換えなどにより，複製を削除するピアは，そのデータの更新伝播木から脱退する．このとき，ピアの脱退により更新伝播木が分断されるため，H ピア，L ピアともに，以下の手順に従って，更新伝播木の修復を行う．ここで，更新伝播木から脱退するピアを脱退希望ピアと呼ぶ．

- (1) 脱退希望ピアが更新伝播木の葉ノードの場合，自身の親ノードに木からの脱退を通知する．脱退要求を受けた親ノードは，自身の持つ子ノードに関する情報から脱退希望ピアを削除し，手順 (4) に進む．
- (2) 脱退希望ピアが子ノードを持つ場合，自身の子ノードの中からランダムに 1 つを選択し，脱退要求を伝える．脱退要求を受け取ったピアは，自身が葉ノードでない場合，同様の手順で子ノードを選択し，脱退要求を伝播させる．脱退要求を受け取った葉ノードは，自身と脱退希望ピアの位置を入れ替える．葉ノードの親ノードは，自身の持つ子ノードに関する情報からその葉ノードを削除する．
- (3) 脱退希望ピアの位置に入れ替えられたピアは，新たな位置における親ノードと， k 個下位の子孫ノード（L ピアの場合は子ノード）まで，ピアが入れ替わったことを通知する．通知を受け取った各ピアは，自身の持つ子ノードおよび先祖ノード（L ピアの

場合は親ノード）に関する情報に含まれる脱退希望ピアを，入れ替えられたピアの情報に変更する．

- (4) 脱退希望ピアが H ピアであり，その脱退希望ピアに L 木が付属している場合，位置を入れ替えたピア（脱退希望ピアが葉ノードの場合はその親ノード）にその L 木の付け替え要求を送信する．付け替え要求を受け取った H ピアは，自身に L 木が付属していない場合は，受け取った L 木を自身に付属する L 木とする．自身に L 木が付属している場合，その L 木の根ノードに付け替え要求を送信する．その後，その L 木の根ノードを責任ノードとし，3.5 節と同様の手順に従って L 木に参加する．その後，脱退希望ピアは，先祖ノード（L ピアの場合は親ノード）や子ノードに関する情報をすべて削除し，脱退を完了する．

3.7 ピアの移動

3.2 節で述べたとおり，UPT-HL 法では，複製を所持する各ピアが，そのデータの必要性を調査し，H ピアと L ピアを適宜移動する．

3.7.1 移動基準の決定

複製を所持する各ピアは，H ピアと L ピアの移動基準の決定のために，自身が所持するデータごとに，データアクセス回数と更新発生回数の 2 つの値を保持する．ここで，データアクセス回数とは，自身および他のピアが，自身の所持する複製にアクセスした回数の合計を表す．また，更新発生回数とは，複製を所持している間に更新情報（更新データもしくは無効化情報）を受け取った回数を表す．各ピアは，更新情報を受け取ったときに，自身のデータアクセス回数と更新発生回数を比較し，H ピア-L ピア間を移動するかを決定する．具体的には，データアクセス回数が更新発生回数以上で，自身が L ピアである場合，H ピアへ移動する．一方，データアクセス回数が更新発生回数より小さく，自身が H ピアである場合，L ピアへ移動する．

3.7.2 H ピアから L ピアへの移動

L ピアへ移動することを決定した H ピア（移動ピア）は，まず自身に L 木が付属しているかどうかを調べる．L 木が付属している場合は，その L 木を他の H ピアに付け替えた後，移動ピア自身も L ピアとしてその L 木に参加する．以下に，移動手順の詳細を，図 4 を用いて説明する．

- (1) 移動ピアは，図 4(a) のように，H 木における自身の子ノードに脱退要求を伝播させ，葉ノードと位置を入れ替える（移動ピアが葉ノードであった場合は親ノードに脱退を通知する）ことにより，3.6 節の手順 (2) と同様の手順に従って H 木から脱退する．

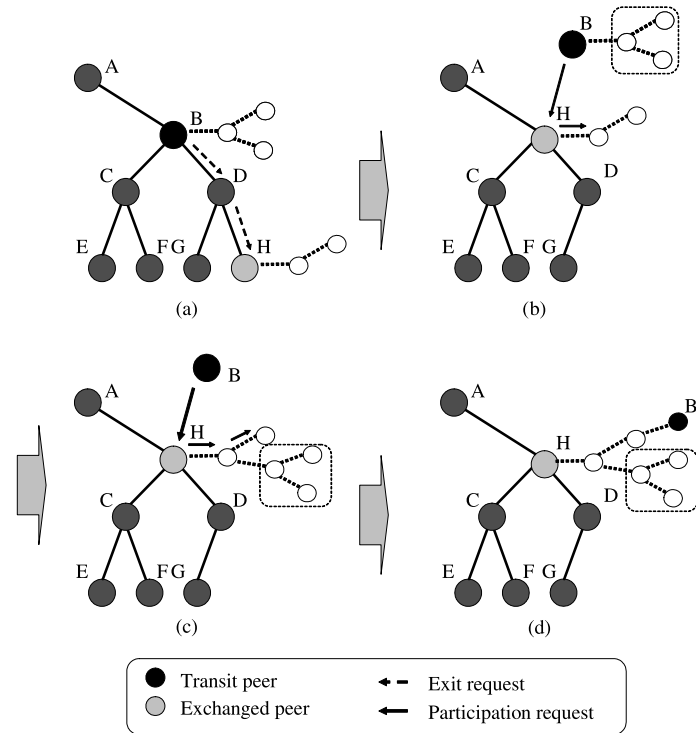


図 4 H ピアから L ピアへの移動
Fig. 4 Transition from H peer to L peer.

- (2) H 木から脱退した移動ピアは、自身に L 木が付属している場合、3.6 節の手順 (4) と同様の手順に従って、図 4 (b) のように、L 木の付け替えを行う。
- (3) 自身に付属する L 木が存在しない、もしくは、自身に付属する L 木の移動を完了した移動ピアは、図 4 (c) のように、自身と位置を入れ替えたピア (移動ピアが葉ノードの場合はその親ノード) に、L ピアとしての参加要求を送信する。参加要求を受け取った H ピアは、自身に L 木が付属していない場合は、移動ピアを自身に付属する L ピアとして参加させる。自身に L 木が付属している場合、その L 木の根ノードに参加要求を送信する。その後、その L 木の根ノードを責任ノードとし、3.5 節の (3) 以降の手順において、 $k = 1$ の場合と同様の動作によって L 木に参加する (図 4 (d))。

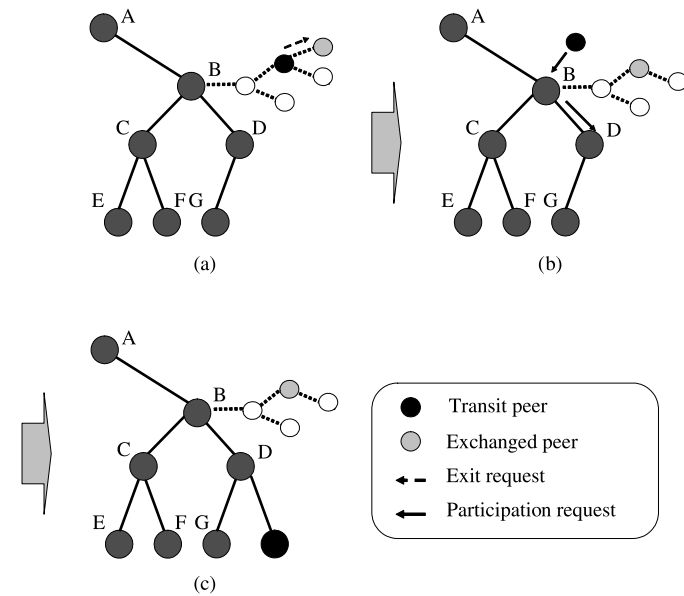


図 5 L ピアから H ピアへの移動
Fig. 5 Transition from L peer to H peer.

3.7.3 L ピアから H ピアへの移動

H ピアへ移動することを決定した L ピアは、図 5 (a) のように、L 木における自身の子ノードに脱退要求を伝播させ、3.6 節の手順 (2) と同様の手順に従って L 木から脱退する。その後、図 5 (b) に示すように、自身の責任 H ノードに参加要求を送信し、3.5 節の手順 (2) ~ (4) と同様の手順に従って H 木に参加する (図 5 (c))。

4. 性能評価

本章では、UPT-HL 法の性能評価のために行ったシミュレーション実験の結果を示す。実験では、非構造型 P2P ネットワーク上でデータ共有を行う環境を想定した。

4.1 シミュレーション環境

シミュレーション実験では、P2P ネットワークに参加するピア数を 1,000 とし、それらがべき法則 (Power-Law Random Graph: PLRG)¹⁾ に従ってネットワークを構成するものとした。ここで、 i 番目のピアの隣接ピア数 d_i を以下の式で与え、一部のピアにリンク

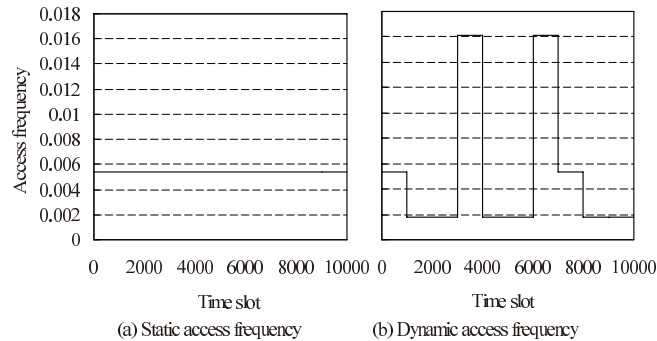


図 6 データアクセス頻度
Fig. 6 Data access frequency.

が集中する環境を実現した．

$$d_i = \lfloor 20 \cdot i^{-0.4} \rfloor \quad (1)$$

データの種別を 100 とし、全ピアのうち、ピア番号が 1 から 100 までのピアがそれぞれ、データ番号 1 から 100 のデータのオリジナルを所持するものとした．各ピアはそれぞれ、1 タイムスロットごとに 0.1 の確率であるデータを要求する．要求するデータ番号の分布は Zipf 分布に従うものとし、データ番号が小さいデータに対する要求ほど頻繁に発生するものとした．具体的には、データ番号 j のデータの要求確率 q_j を、以下の式で与えた．

$$q_j = \frac{j^{-\alpha}}{\sum_{k=1}^{100} k^{-\alpha}} \quad (2)$$

上式において、 α はデータアクセス頻度の偏りを決定するパラメータであり、Zipf 係数と呼ばれる．シミュレーション実験では、Zipf 係数 α は 0.5 とした．ここで、本実験における、1 タイムスロットごとのデータ番号 j のデータの要求確率は $0.1 \times q_j$ で計算される．この式に基づいて計算された、データ番号 1 のデータに対するデータアクセス頻度を図 6(a) に示す．図において、横軸はタイムスロット、縦軸はデータアクセス頻度を表す．また、4.2 節では、データアクセス頻度に偏りが存在する環境においての評価も行った．この評価では、まずデータ番号 1 のデータに対するデータアクセス頻度を図 6(b) に示す値に設定する．なお、データアクセス頻度が変化するタイミングや値は、変化が小さい場合、大きい場合、大きい変化と小さい変化が連続的に発生する場合を網羅することを考慮して、適当に設定した．また、データ番号 1 以外のデータの要求確率は、1 からデータ番号 1 のデータの要求

確率を引いた値を、式 (2) で与えられる各データの要求確率の比と等しくなるように分配する．具体的には、データ番号 1 のデータの要求確率を q'_1 とした場合、データ番号 j のデータの要求確率 q'_j を、以下の式で与えた．

$$q'_j = (1 - q'_1) \frac{q_j}{\sum_{k=2}^{100} q_k} \quad (j \geq 2) \quad (3)$$

図 6(b) から分かるように、データアクセス頻度に偏りがある環境では、一時的にデータ要求が集中して発生する．ここで、データアクセス頻度に偏りがある環境でも、シミュレーション時間全体におけるデータアクセス頻度は、 $0.1 \times q_j$ に従っている．

クエリの伝播にはエキスパンディングリングを用いた．エキスパンディングリングでは、小さな値の TTL でフラッディングを開始し、データを発見できなかった場合は、TTL の値を増加させて再度検索を行う．本実験では、初期 TTL = 1 とし、検索失敗時には TTL の値を 1 ずつ増加させるものとした．

複製の配置方式には、オーナ複製法⁹⁾を用いた．オーナ複製法では、データ要求が成功した際、クエリを発行したピアにのみ複製を配置する．更新伝播木は H 木、L 木ともに 2 分木 ($n = 2$) とした．

各データのサイズはすべて等しく、複製を保有可能な数はすべてのピアで 10 (データのオリジナルを所持するピアは 11) とした．各ピアが複製を作成する際にデータ記憶領域に空きがない場合は、所持していた複製の中で最後にアクセスされた複製を削除し、新たな複製を作成する (LRU) のものとした．また、オリジナルデータは削除しないものとした．オリジナルノードは、1 タイムスロットごとに一定の確率 (更新発生率) でデータを更新するものとした．

以上の環境において、10,000 タイムスロットのシミュレーション実験を行った．なお、実験では、アクセス頻度が最も大きいデータ番号 1 のデータに注目して、このデータに関する評価値のみを採取した．

4.2 複製配置の有無による影響

最初に、オリジナルデータの更新発生率を 0.05 とし、データアクセス頻度に偏りが無い環境とある環境において、UPT-HL 法と複製を作成しない方法における以下の値を評価した．なお、UPT-HL 法において、各 H ピアは H 木上の親ノードまでの情報を保持する ($k = 1$) のものとした．

• オリジナルノードの負荷

10 タイムスロットごとの、オリジナルノードが他のピアからのデータ要求に応えた回

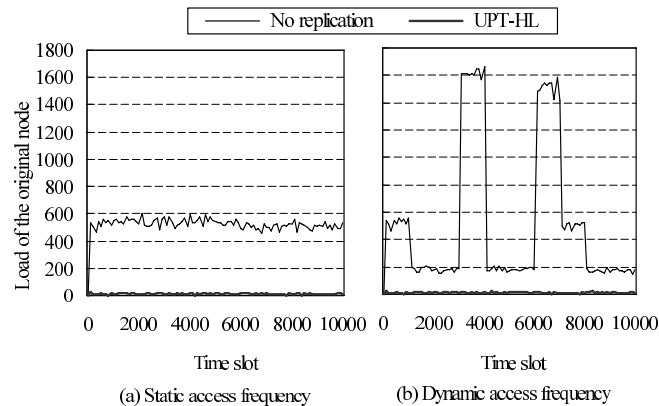


図7 オリジナルノードの負荷
Fig. 7 Load of original node.

数、および、更新が発生した場合に自身の子ノードへと更新を伝播させた回数の合計（複製を作成しない方法では前者のみ）。

シミュレーション実験の結果を図7に示す。図において、横軸がタイムスロット、縦軸がオリジナルノードの負荷を表す。図7より、複製を作成しない方法では、UPT-HL法に比べて、オリジナルノードの負荷が大きくなっていることが分かる。これは、複製を作成しない場合には、データを所持するオリジナルノードにデータ要求が集中してしまう一方で、複製を作成することにより、データ要求が複製所持ピアにも分散されるためである。また、図6(b)および図7(b)より、データアクセス頻度に偏りが存在する場合、複製を作成しない方法では、データアクセス頻度に比例してオリジナルノードの負荷も増減するため、データアクセスが一時的に集中する環境ではオリジナルノードの負荷が極端に増大してしまうことが分かる。一方、UPT-HL法では、データアクセス頻度の変化による影響が小さく、データアクセスが集中したときにも負荷が分散されている。

以上のことから、負荷分散のために複製を配置することが有効であることが確かめられる。

4.3 複製配置方式の比較

続いて、複製を配置することを前提とし、UPT-FT法、UPT-H (UPT with H peers)法、2.4.2項で説明したチェーン構造を用いた更新伝播法（チェーン伝播法）($m = 5$)、および、UPT-HL法の性能を比較した。ここで、UPT-H法とは、データアクセス回数より

も更新発生回数の方が大きい場合にLピアに移動せず、そのピアが持つ複製を削除して更新伝播木から脱退する手法である。これらの実験では、データ番号1のデータアクセス頻度は図6(a)のように一定であるものとし、更新発生率を0.01~0.09の範囲で変化させたときの、以下の値を評価した。なお、UPT-FT法、UPT-HL法、およびUPT-H法において、 $k = 1$ の場合と $k = 5$ の場合の性能を評価した。

- 更新情報伝播負荷

更新発生時に、Hピアに更新データを伝播させることによる負荷、および、Lピアに無効化情報を伝播させることによる負荷のシミュレーション時間全体の合計。ただし、更新データのサイズを100、無効化情報のサイズを1とし、伝播されるデータの総量を負荷とした。

- 論理ネットワーク維持負荷

ピアの参加・脱退時に更新伝播用の論理ネットワーク（木構造、チェーン構造）を修復するために交換されるメッセージ、および、Hピア、Lピアへの移動時に交換されるメッセージの負荷の合計（シミュレーション時間全体）。このとき交換されるメッセージのサイズは、無効化情報と同じく1とした。

- 平均更新伝播遅延

更新伝播時の、H木の根からH木上の各節点までのホップ数の平均値。

- 検索成功率

対象データの検索成功率。

- 検索負荷

エキスパンディングリングによる対象データ検索時において、各TTLで検索が成功した場合に発生する検索メッセージ数（シミュレーション時間全体）、およびその累計。このときのメッセージのサイズも1とした。

ここで、チェーン伝播法は、複製を所持するすべてのピアに更新データを伝播させるという点でUPT-FT法と同じであり、更新情報伝播負荷、検索成功率および検索負荷の結果はUPT-FT法と等しくなる。そのため、チェーン伝播法に関しては、論理ネットワーク維持負荷と平均遅延の結果のみを記載する。

4.3.1 更新情報伝播負荷

更新情報伝播負荷を図8に示す。図8において、横軸は更新発生率を表し、縦軸は更新情報伝播負荷を表す。結果より、すべての手法において、更新発生率が増加するにつれて、更新情報伝播負荷が増大していることが分かる。ここで、UPT-FT法において、更新発生率

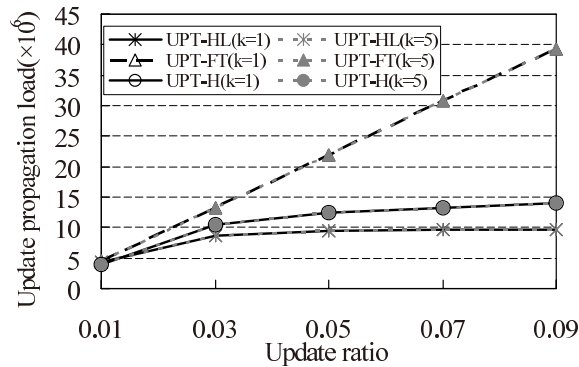


図 8 更新情報伝播負荷

Fig. 8 Update propagation load.

とともに更新情報伝播負荷が線形に増加しているのに対し、UPT-HL 法と UPT-H 法では、負荷の増加量が小さく抑えられている。UPT-FT 法では複製を所持するすべてのピアに更新データを伝播させるため、更新発生率に比例して負荷が大きくなる。一方、更新発生率が大きくなるにつれて、UPT-HL 法では、H ピアが L ピアに移動する可能性が大きくなり、UPT-H 法では、複製が削除される可能性が高くなる。そのため、更新データを伝播すべきピアの数が減少し、負荷が抑えられる。また、UPT-HL 法と UPT-H 法では、UPT-HL 法における負荷の方がより小さく抑えられている。これは、UPT-HL 法では L ピアでありながら最新のデータを要求することが可能である一方で、UPT-H 法では、最新のデータを要求したピアは必ず H 木に参加しなければならないため、その分更新データを受け取る回数が多くなるからである。

また、UPT-FT 法、UPT-HL 法、および UPT-H 法において、 $k = 5$ に設定した場合、更新情報伝播負荷は、 $k = 1$ の場合からまったく変化していない。これは、 k の値を変化させた場合、ピアが更新伝播木に参加する位置は変化するものの、複製を所持する（更新伝播木に参加する）ピア、および（UPT-HL 法においては）H ピアと L ピアの割合は変化しないためである。

4.3.2 論理ネットワーク維持負荷

論理ネットワーク維持負荷を図 9 に示す。図 9 において、横軸は更新発生率を表し、縦軸は論理ネットワーク維持負荷を表す。結果より、UPT-HL 法では、更新発生率が増加する

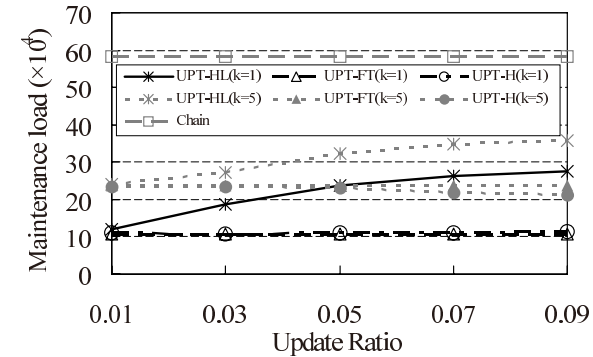


図 9 論理ネットワーク維持負荷

Fig. 9 Maintenance load of overlay network.

につれて木構造維持負荷が大きくなるが、UPT-FT 法では、更新発生率とは関係なく木構造維持負荷が小さい値で一定に保たれていると同時に、UPT-H 法でも、負荷の増加量がきわめて小さく抑えられていることが分かる。これは、UPT-HL 法ではピアの参加、脱退に加えて、更新発生時に H ピアと L ピアの入れ替えを行うため、他の手法に比べて各ピア間で交換されるメッセージ数が多くなるためである。一方、UPT-FT 法では、ピアの入れ替えは行わず、UPT-H 法においても、H 木からの脱退時のメッセージ数が増加するだけ（木構造の性質上、半数のピアが葉ノードとなり、少ないメッセージ数で脱退が可能である）であり、各ピアで交換されるメッセージ数が少なくなる。チェーン伝播法では、参加、脱退のたびに左右 m 個ずつのピアにメッセージを送信しなければならないため、木構造を用いた他の 3 つの手法に比べてチェーン構造維持負荷が大きくなる。

UPT-HL 法では、更新発生率が高くなるにつれて木構造維持に必要なメッセージ数が増加してしまう。しかし、図 8 の結果から分かるように、木構造維持負荷の増加量に比べ、更新情報伝播負荷の減少量の方が大きく、全体の負荷は減少する。このように、無効化情報や論理ネットワーク維持のためのメッセージのサイズと比較して、更新データのサイズが大きい場合は、ネットワーク全体の負荷が減少することが確かめられる。ここで、図 8 は、更新データのサイズが無効化情報、および論理ネットワーク維持のためのメッセージの 100 倍である場合の結果であり、更新データのサイズが小さくなると、更新情報伝播負荷がすべての手法において減少する。たとえば、更新データのサイズを 50 とした場合、更新情報伝播負荷はおよそ半分になる。一方、図 9 の結果は更新データのサイズに依存しないため、

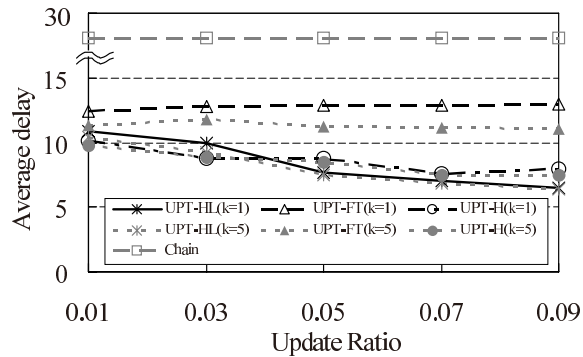


図 10 平均遅延
Fig. 10 Average delay.

更新データのサイズが小さい場合は、ネットワーク全体の負荷が逆転し、UPT-HL 法の負荷が他の手法より大きくなる可能性がある。したがって、想定するアプリケーションによって、適した更新伝播手法を適用する必要があるものと考えられる。

また、UPT-FT 法、UPT-HL 法、および UPT-H 法において $k = 5$ に設定した場合、論理ネットワーク維持負荷が $k = 1$ の場合より増加している。これは、 k の値が大きい場合、更新伝播木に参加するピアがより多くの先祖ノードの情報を管理する必要があり、交換されるメッセージ数や、それぞれのサイズが大きくなるためである。

4.3.3 更新伝播時の遅延

平均更新伝播遅延を図 10 に示す。図 10 において、横軸は更新発生率を表し、縦軸は平均遅延を表す。結果より、UPT-FT 法に比べ、UPT-HL 法と UPT-H 法における遅延が小さくなっていることが分かる。これは、UPT-FT 法では複製を所持するピアすべてに更新を伝播しなければならないのに対し、UPT-HL 法および UPT-H 法では、更新データを送るべきピアが H ピアのみ限定されるためである。また、これら 2 つの手法では、更新発生率が高くなるにつれて、データアクセス回数よりも更新発生回数の方が大きくなるピアが増加し、更新データを伝播させるべきピア (H ピア) の数がより減少する。そのため、更新発生率が高くなるにつれて、遅延が小さくなる。チェーン伝播法では、左右 m 個ずつのピアに直線的に更新データを伝播させるため、遅延が大きくなる。 m の値を大きくすることで遅延は減少するが、その場合は前節で述べた論理ネットワーク維持負荷が増加してしまう。

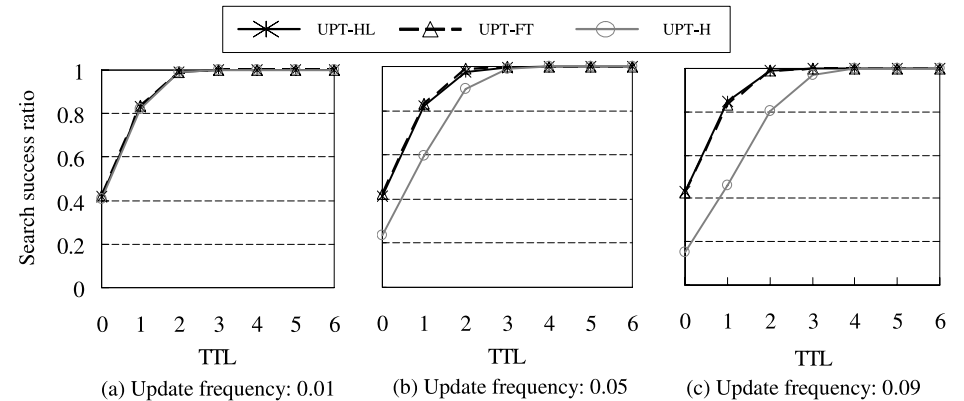


図 11 検索成功率
Fig. 11 Search success ratio.

一方、 $k = 5$ に設定した場合、 $k = 1$ の場合に比べて、UPT-FT 法では多少の遅延減少が見られるものの、UPT-H 法、および UPT-HL 法における平均遅延の減少はわずかであることが分かる。3.5 節で述べたとおり、 k の値を大きくすることによって更新伝播木の高さが抑制できるが、これは、更新伝播木に参加するピア数が多い場合に特に有効である。そのため、UPT-FT 法と比べて、H ピアの数が少なくなる UPT-H 法および UPT-HL 法では、 k の値を大きくしても、平均遅延が減少しにくくなっている。

4.3.4 検索成功率

更新発生率が 0.01, 0.05, および 0.09 の場合の検索成功率を図 11 に示す。図 11 において、横軸は TTL を表し、縦軸は検索成功率を表す。なお、TTL = 0 のとき、データを要求したピア自身が、そのデータのオリジナルもしくは複製を所持していた場合に、検索成功とする。また、UPT-HL 法において、L ピアは最新のデータを所持する H ピアへのリンクを所持しているため、L ピアを発見した場合でも検索は成功であるとした。ここで、4.3.1 項で述べたように、 k の値を変化させても、複製を所持しているピアは変化しないため、検索成功率は k の値に依存しない。そのため、本項では $k = 1$ の場合の結果のみを示す。

まず、図 11 (a) より、更新発生率が小さい環境において、3 つの手法で検索成功率に大きな差が見られないことが分かる。これは、更新があまり発生しないため、更新発生回数がアクセス回数を上回るピアが少なくなるためである。これにより、UPT-HL 法における L ピアに移動するピアや、UPT-H 法における複製を削除するピアがほとんど存在せず、3 つの

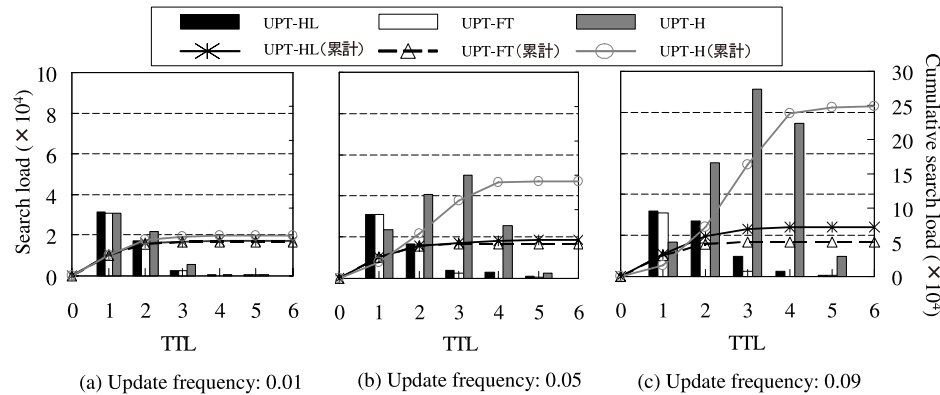


図 12 検索負荷
Fig. 12 Search load.

手法における更新伝播木の形状がほぼ等しくなるため、検索成功率に大きな差が生じない。一方で、図 11 (b) および (c) より、更新発生率が大きくなるにつれて、UPT-H 法の検索成功率が低下している。これは、UPT-H 法では、更新発生率が大きい環境では、複製を持つピアの数が少なくなり、データの発見が困難になるためである。

4.3.5 検索負荷

更新発生率が 0.01, 0.05, および 0.09 の場合の検索負荷を図 12 に示す。図 12 において、横軸は TTL を表し、縦軸は検索負荷を表す。また、棒グラフは、各 TTL で検索が成功した場合にその検索で発行されたメッセージの総数を表し、折れ線グラフは、その累計を表す。本項においても、4.3.4 項と同様の理由により、 $k = 1$ の場合の結果のみ示す。

図 12 より、UPT-FT 法に比べて、UPT-HL 法の検索負荷がわずかに増加していることが分かる。これは、UPT-HL 法においては、L ピアを発見した場合、その L ピアの責任 H ノードにデータ要求を送る必要があるためである。

一方、UPT-H 法では、他の手法と比較して検索負荷が増大しており、更新発生率の増加とともに、その差が顕著となっている。これは、図 11 の結果から分かるように、UPT-H 法に関しては、他の手法と比べてデータを発見するまでに必要なホップ数が大きくなり、エキスパンディングリングにおける検索負荷が指数関数的に増加するためである。

4.4 H ピアの負荷

UPT-HL 法において、更新発生率を変化させた場合の以下の値を評価した。

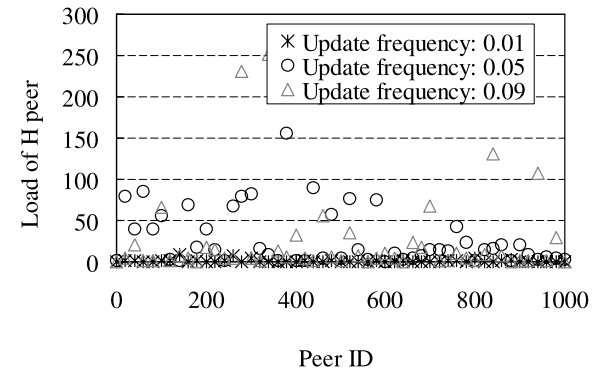


図 13 H ピアの負荷
Fig. 13 Load of H peer.

● H ピアの負荷

各 H ピアに付属する L 木上の L ピア (無効化情報を受信済み) から、最新のデータを要求される回数 (シミュレーション時間全体)。なお、本手法では、各ピアは適宜 H ピアと L ピアを移動するため、調査対象のピアが H ピアであるときに要求を受けた回数の合計を表す。

シミュレーション実験の結果を、図 13 に示す。ここで、負荷を調べるピアは、1,000 個の中から無作為に 50 個のピアを選択した。図 13 において、横軸はピア番号を表し、縦軸は H ピアの負荷を表す。結果より、更新発生率が高い場合、1 つの H ピアが受け取るデータ要求回数が多くなる一方で、更新発生率が低い場合、受け取るデータ要求回数が少なくなっていることが分かる。これは、更新発生率が高い場合、L ピアの割合が多くなり、1 つの H ピアに多くの L ピアが付属するためである。各 H ピアは、自身に付属する L 木上の L ピアからの最新のデータ要求に回答するため、付属する L ピアの数が多いほど H ピアの負荷が大きくなる。更新発生率が 0.09 の場合、きわめて負荷の高い H ピアが存在するほか、更新発生率が 0.05 の場合でも、H ピアの負荷は全体的に大きくなっている。一方、更新発生率が 0.01 の場合、1 つの H ピアに負荷が集中することがなくなっていることが分かる。

ただし、本論文の実験環境を含め、一般的な環境では、H ピアの負荷 (L ピアからのデータ要求の負荷) よりも、更新情報伝播負荷の方が支配的となる。L ピアの数に比べて非常に多い場合など、特殊な場合を除いて、H ピアの負荷は現実的には大きな問題とならない。

4.5 様々な環境における提案手法の有効性に関する考察

本論文におけるシミュレーション実験では、データアクセス頻度が最も高い、データ番号が1のデータを対象とし、すべてのピアが同一のデータアクセス特性に従ってアクセスするデータを決定する環境において、UPT-HL法の有効性を証明した。本節では、データアクセス頻度が低いデータを対象とした場合、および、各ピアが異なるデータアクセス特性に従う環境における、UPT-HL法の有効性に関する考察を行う。

4.5.1 データアクセス頻度の小さいデータを対象とした場合

3.7.1項で述べているとおり、UPT-HL法では、更新発生回数とデータアクセス回数の大小関係によって、Hピア、Lピアの移動判定を行う。したがって、データアクセス頻度が低い場合は、更新発生率が高い場合と同等の結果が得られる。4.3節の結果から分かるとおり、UPT-HL法は、更新発生頻度が高い環境においても有効である。すなわち、UPT-HL法は、データアクセス頻度が小さいデータを対象とした場合でも有効であるといえる。

4.5.2 ピア間でデータアクセス頻度が異なる場合

ピア間でデータアクセス頻度が異なる場合は、対象となるデータの複製を保持するピアが、そのデータに頻繁にアクセスするもののみ限定されやすくなり、比較対象も含めたすべての手法において、更新データを伝播させるべきピア数が減少する。また、UPT-HL法では、HピアとLピアの分類がより明確に行われ、更新伝播にかかる負荷や遅延を効果的に抑制できる。さらに、HピアとLピアの移動も発生しにくくなり、論理ネットワーク維持負荷も小さくなるものと考えられる。

一方、本章のシミュレーション実験では、すべてのピアが同一のデータアクセス頻度に従ってデータにアクセスする環境を想定しており、HピアとLピアの明確な分類が行われにくく、両ピア間の移動が繰り返される不安定な状態になりやすい。このように、UPT-HL法に不利な環境においても、UPT-HL法の有効性が示されていることから、ピア間でデータアクセス頻度が異なる場合においても、UPT-HL法は有効であるといえる。

5. おわりに

本論文では、P2Pモデルを用いたデータ共有サービスにおいて、ピアが共有するデータ(複製)に更新が発生する環境を想定し、複製の更新にかかる遅延や負荷を抑制する複製更新伝播法(UPT-HL法)を提案した。UPT-HL法では、筆者らがこれまでに提案した複製更新伝播法を拡張し、データアクセス頻度に応じて異なる更新伝播木を形成する。更新発生時には、データアクセス頻度の高いピアには更新データを伝播する一方で、データアクセス

頻度の低いピアには無効化情報のみを伝播することで、更新伝播時の負荷や遅延の減少を実現する。

UPT-HL法の性能を評価するために、シミュレーション実験により、データアクセス頻度を考慮せず複製を所持するすべてのピアに更新データを伝播させる手法、および、Lピアへの移動を行わず複製を削除する手法との比較を行った。その結果からUPT-HL法は、更新伝播時の負荷や遅延を抑えながらも、検索時に必要なメッセージ数も抑えられることを確認した。なお、本研究では、式(2)のZipf係数 α の値を変化させることにより、異なるアクセス特性の環境においても同様の実験を行った。その結果、本論文で説明した $\alpha = 0.5$ の場合と同様の結果が得られ、異なるアクセス特性の環境でのUPT-HL法の有効性が確かめられた。

本研究では、データの更新発生間隔が大きく変化しない環境を想定している。今後、データの更新発生間隔などを動的に変化させた場合の適切なHピアとLピアの移動基準の検討を行う予定である。さらに、更新伝播用の論理ネットワークの維持負荷を軽減させるために、Lピアを木構造以外の方法で維持する方法を検討する予定である。また、本論文では、議論の簡化のためにピアの不当な退出を考慮しておらず、不当な退出発生時のピアの動作を規定していなかった。Hピアに関しては、保持する先祖ノードの数を増やすなど、UPT-HL法と同様の操作により対応できる。一方、Lピアに関しては、現状では、分断されたLピアが自身の責任Hノードに問い合わせることで、木の修復が可能であるものと考えられるが、そのほかにも、Lピアにも多くの先祖ノードの情報を所持させる、管理する責任Hノードの数を増やすなど、様々な対応が考えられる。今後、UPT-HL法においてもピアの不当な退出を考慮し、各ピアの動作の規定、および性能評価を行う予定である。

謝辞 本研究の一部は、科学技術振興調整費「先端融合領域イノベーション創出拠点の形成：ゆらぎプロジェクト」、特定領域研究(18049050)の研究助成によるものである。ここに記して謝意を表す。

参 考 文 献

- 1) Admic, L.A., Lukose, R.M., Puniyani, A.R. and Huberman, B.A.: Search in Power-Law Networks, *Physical Review E*, Vol.64, No.4, 046135 (2001).
- 2) Chen, X., Ren, S., Wang, H. and Zhang, X.: SCOPE: Scalable Consistency Maintenance in Structured P2P Systems, *Proc. INFOCOM'05*, pp.1502-1513 (2005).
- 3) Clarke, I., Sandberg, O., Wiley, B. and Hong, T.W.: Freenet: A Distributed Anonymous Information Storage and Retrieval System, *International Workshop on Design*

Issues in Anonymity and Unobservability (2001).

- 4) Cohen, E. and Shenker, S.: Replication Strategies in Unstructured Peer-to-peer Networks, *Proc. SIGCOMM'02*, pp.177–190 (2002).
- 5) Datta, A., Hauswirth, M. and Aberer, K.: Updates in Highly Unreliable, Replicated Peer-to-peer Systems, *Proc. ICDCS'03*, pp.76–85 (2003).
- 6) Ferreira, R.A., Ramanathan, M.K., Awan, A., Grama, A. and Jagannathan, S.: Search with Probabilistic Guarantees in Unstructured Peer-to-peer Networks, *Proc. P2P'05*, pp.165–172 (2005).
- 7) Jannotti, J., Gifford, D.K., Johnson, K.L., Kaashoek, M.F. and O'Toole, J.W.: Overcast: Reliable Multicasting with an Overlay Network, *Proc. OSDI'00*, pp.197–212 (2000).
- 8) Leontiadis, E., Dimakopoulos, V.V. and Pitoura, E.: Cache Updates in a Peer-to-peer Network of Mobile Agents, *Proc. P2P'04*, pp.10–17 (2004).
- 9) Lv, Q., Cao, P., Cohen, E., Li, K. and Shenker, S.: Search and Replication in Unstructured Peer-to-peer Networks, *Proc. ICS'02*, pp.84–95 (2002).
- 10) On, G., Schmitt, J. and Steinmetz, R.: The Effectiveness of Realistic Replication Strategies on Quality of Availability for Peer-to-peer Systems, *Proc. P2P'03*, pp.57–64 (2003).
- 11) Qiao, Y. and Bustamante, F.E.: Elders Know Best – Handling Churn in Less Structured P2P Systems, *Proc. P2P'05*, pp.77–86 (2005).
- 12) Roussopoulos, M. and Baker, M.: CUP: Controlled Update Propagation in Peer-to-peer Networks, *Proc. USENIX'03* (2002).
- 13) Tan, X. and Datta, S.: Building Multicast Trees for Multimedia Streaming in Heterogeneous P2P Networks, *Proc. ICMCS'05*, pp.141–146 (2005).
- 14) Tran, D.A., Hua, K.A. and Do, T.T.: Zigzag: An Efficient Peer-to-Peer Scheme for Media Streaming, *Proc. INFOCOM'03*, pp.1283–1292 (2003).
- 15) Wang, Z., Das, S.K., Kumar, M. and Shen, H.: An Efficient Update Propagation Algorithm for P2P Systems, *Computer Communications*, pp.1106–1115 (2007).
- 16) Watanabe, T., Kanzaki, A., Hara, T. and Nishio, S.: An Update Propagation Strategy for Delay Reduction and Node Failure Tolerance in Peer-to-Peer Networks, *Proc. FINA'07*, pp.103–108 (2007).
- 17) Yin, L. and Cao, G.: DUP: Dynamic-tree Based Update Propagation in Peer-to-peer Networks, *Proc. ICDE'05*, pp.258–259 (2005).

(平成 19 年 9 月 1 日受付)

(平成 20 年 2 月 5 日採録)



渡辺 俊貴 (学生会員)

2006 年大阪大学工学部電子情報エネルギー工学科卒業。2007 年同大学院情報科学研究科博士前期課程修了。現在、同大学院情報科学研究科博士後期課程在学中。P2P ネットワークにおける複製管理に興味を持つ。電子情報通信学会、日本データベース学会の学生会員。



神崎 映光 (正会員)

2002 年大阪大学工学部情報システム工学科卒業。2004 年同大学院情報科学研究科博士前期課程修了。2005 年同大学院情報科学研究科博士後期課程中退後、同大学院情報科学研究科マルチメディア工学専攻特任助手を経て、2007 年より同助教となり、現在に至る。博士(情報科学)。移動体ネットワーク、通信プロトコル、分散処理に興味を持つ。IEEE、電子情報通信学会、日本データベース学会の各会員。



原 隆浩 (正会員)

1995 年大阪大学工学部情報システム工学科卒業。1997 年同大学院工学研究科博士前期課程修了。同年同大学院工学研究科博士後期課程中退後、同大学院工学研究科情報システム工学専攻助手、2002 年同大学院情報科学研究科マルチメディア工学専攻助手、2004 年より同大学院情報科学研究科マルチメディア工学専攻准教授となり、現在に至る。工学博士。1996 年本学会山下記念研究賞受賞。2000 年電気通信普及財団テレコムシステム技術賞受賞。2003 年本学会研究開発奨励賞受賞。データベースシステム、分散処理に興味を持つ。IEEE、ACM、電子情報通信学会、日本データベース学会の各会員。



西尾章治郎（フェロー）

1975年京都大学工学部数理工学科卒業．1980年同大学院工学研究科博士後期課程修了．工学博士．京都大学工学部助手，大阪大学基礎工学部および情報処理教育センター助教授，大阪大学大学院工学研究科情報システム工学専攻教授を経て，2002年より大阪大学大学院情報科学研究科マルチメディア工学専攻教授となり，現在に至る．2000年より大阪大学サイバーメディアセンター長，2003年より大阪大学大学院情報科学研究科長，その後2007年より大阪大学理事・副学長に就任．この間，カナダ・ウォータールー大学，ビクトリア大学客員．データベース，マルチメディアシステムの研究に従事．現在，Data & Knowledge Engineering等の論文誌編集委員．本会理事を歴任．電子情報通信学会フェローを含め，ACM，IEEE等8学会の各会員．
