Peer-to-Peer ネットワークにおける 木構造を用いた複製更新の伝搬について

中通　実† 　内田　渉†† 　原　隆浩†† 　前田　和彦†† 　西尾　章治郎††

† 大阪大学工学部情報システム工学科　〒 565–0871 大阪府吹田市山田丘 2-1
†† 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　〒 565–0871 大阪府吹田市山田丘 2-1
E-mail: †nakadori@ise.eng.osaka-u.ac.jp, ††{wataru,hara,k.maeda,nishio}@ist.osaka-u.ac.jp

あらまし　近年，計算機の高性能化とネットワークインフラの発達により，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．本稿では，P2P ネットワークにおいてデータ更新時に，複製をもつ全ピアに直ちに更新情報を通知する環境を想定し，更新伝搬のための各ピアの負荷分散および遅延の減少を両立する方式を提案する．提案方式では，オリジナルデータをもつピアを根とし，複製をもつその他のピアを内部節点とする n 分木を自律分散的に作成し，その木に従った順序で更新を伝搬する．さらに本稿では，シミュレーション実験によって，提案方式の有効性を検証する．

キーワード　Peer-to-Peer ネットワーク，複製，データ更新，木構造

On Update Propagation Using a Tree Structure for Replicas in Peer-to-Peer Networks

Minoru NAKADORI†, Wataru UCHIDA††, Takahiro HARA††,

Kazuhiko MAEDA††, and Shojiro NISHIO††

† Dept. of Information Systems Eng., Faculty of Eng., Osaka Univ.
†† Dept. of Multimedia Eng., Graduate School of Information Science and Technology, Osaka Univ.
2–1 Yamadaoka, Suita, Osaka 565–0871, Japan
E-mail: †nakadori@ise.eng.osaka-u.ac.jp, ††{wataru,hara,k.maeda,nishio}@ist.osaka-u.ac.jp

Abstract　Recently, there has been increasing interest in research of data sharing with peer-to-peer networks. In this paper, assuming an environment in which the update information is immediately notified to all peers holding the replicas when the update occurs, we propose a new update propagation strategy that achieves not only load balancing but also delay reduction. The proposed strategy creates an n-ary tree whose root is the owner of the original data and the other nodes are peers holding its replicas, and propagates the update information according to this tree. Moreover, we verify the effectiveness of the proposed strategy by simulation experiments.

Key words　peer-to-peer network, replica, data update, tree-structure

1. はじめに

近年，計算機の高性能化とネットワークインフラの発達により，Peer-to-Peer (P2P) モデルを用いたネットワークサービスが注目されている．Web サービスなどの既存のネットワークサービスが用いるモデルとして，最も主流となっているクライアント・サーバモデルとは異なり，P2P モデルを用いたネットワークサービスでは各端末 (ピア) はサーバ，クライアントといった明確な区別はなく，両方の役割を担い，互いにサービスを提供する．

P2P モデルを用いたネットワークサービスでは，サービスの提供者が分散しているため，各ピアが要求するサービスの提供者を検索するための機構が必要となる．P2P モデルを用いたネットワークサービスの形態は，サービス提供者の検索方法によって，クライアント・サーバシステムを融合させたハイブリッド P2P 型と，完全な分散環境であるピュア P2P 型に分類される．これらの違いは，次の通りである．

- ハイブリッド P2P 型

ハイブリッド P2P 型のネットワークサービスでは，サービス提供者の検索サービスをクライアント・サーバモデルを用いて

提供する．検索サービスを行うサーバは，ネットワーク上の全てのピアの識別子やそれらのピアが提供可能なサービスを，インデックス情報として一括して管理する．そのため，各ピアがサービスを検索する場合，サーバへ問合せを行えばそのサービスを提供可能なピアを容易に発見できるという利点がある．しかし，特定の機能を一箇所のサーバに置くため，サーバが単一障害点になり，サーバが停止した場合は，サービス全体が停止してしまう．さらに，完全な分散環境ではないため，クライアント数に対するスケーラビリティを得ることができず，クライアント・サーバ型の問題点を完全に解決することができない．

- ピュア P2P 型

ピュア P2P 型のネットワークサービスでは，ハイブリッド P2P 型のようにインデックス情報を管理するサーバは存在せず，各ピアが自律分散的に動作し，サービス検索を行う．各サービスを提供可能なピアに関する情報は，ネットワーク全体の一部分をそれぞれのピアが管理する．また，各ピアはサービス要求 (クエリ) を伝搬するために，物理ネットワークとは独立した論理的なネットワークを構成する．サービスの検索は，クエリを論理ネットワークで隣接する他ピアへ送信し，要求サービスを提供可能なピアを発見するまでそれを繰り返すことによって実現される．従って，検索ネットワークの構造によっては，検索時間が長くなってしまう上，検索の成功を保証することができない．しかし，サービスを行うシステム全体が完全な分散システムとなるため，クライアント数に対するスケーラビリティ，高い可用性を実現することができる．

本研究では，数百万規模の端末が参加するような大規模なデータ共有サービスを想定する．そのため，ピュア P2P 型のデータ共有を想定する．

P2P ネットワークを用いたデータ共有では，検索効率や可用性の向上，負荷分散のために，データの複製を作成し，ネットワーク上の複数のピアに配置するのが一般的である [2]．ここで，分散ファイルシステムサービス [5] を提供する場合など，共有されるデータに更新が発生するような環境では，ピアが更新前の古い複製にアクセスする可能性がある．更新前の複製が無効であり，更新されたデータが重要とされる環境では，無効な複製へのアクセスは，アプリケーションにおけるロールバック処理を引き起こすため，ネットワーク上の全ての複製を常に最新の状態に保つ必要がある．そこで，更新が発生した際には，複製をもつ全てのピアにその更新情報を即座に通知する必要がある．

複製をもつピアに更新情報を通知するための単純な方法として，オリジナルデータをもつピア (以下では，オリジナルノードと表記する) が，そのデータの複製をもつ全てのピアのネットワーク上での識別子 (例えば IP アドレスなど) を管理しておき，更新が発生するたびに複製をもつ全ピアに更新情報を直接伝搬する方法 (以下では，放射伝搬法と表記する) が考えられる．この方法を用いた場合，更新が発生した時，その更新情報を即座に複製をもつ全てのピアに通知することができる．しかし，更新伝搬時の負荷がオリジナルノードに集中し，複製をもつピア数の増加に従って，線型的に増加するため，本研究で想定する大規模なサービスには対応することができない．

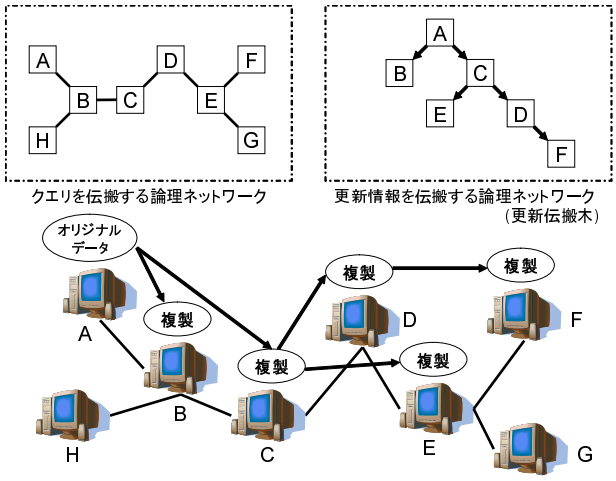

図 1　木構造を用いた更新の伝搬．
Fig. 1　Update propagation using the tree structure.

この問題を解決する更新伝搬方法として，複製をもつ各ピアが，同じ複製をもつ他の 1 つのピアへ，直線的に更新を伝搬する方法 (以下では，直線伝搬法と表記する) が考えられる．この方法により，更新伝搬時における各ピアの負荷を分散することができる．しかしこの方法を用いた場合，P2P ネットワークに参加するピア数が増加し，複製をもつピア数の増加に従って，更新を伝搬する経路は線型的に長くなる．従って，オリジナルノードから複製をもつ全てのピアへ更新の伝搬が完了するまでに，大きな遅延が生じてしまう．

そこで，本研究では，P2P ネットワークにおける複製更新の伝搬時における負荷分散と遅延減少を目的とし，それらを両立する方式を提案する．提案方式は，各データに対してオリジナルノードを根とし，複製をもつその他のピアを内部節点とする n 分木状の論理ネットワークを，クエリを送信する通常の P2P 論理ネットワークとは独立に作成する．更新発生時には，この木に従った順序で更新を伝搬する (図 1)．以下ではこの木を更新伝搬木と呼ぶ．また，P2P 型データ共有サービスでは，ネットワーク状況の変化に対応するため，頻繁に複製の作製・削除が行われる．そのような場合，更新伝搬木の構成を保つ必要がある．そこで本研究では，複製配置時と複製削除時における更新伝搬木の再構成方法についても提案する．

以下，2 章で関連研究について述べ，3 章で本研究の提案方式について説明する．4 章で提案方式の性能評価を行い，5 章で提案方式の拡張について述べる．最後に 6 章で本研究のまとめを行う．

2. 関 連 研 究

本章では，ピュア P2P 型ネットワークを用いたデータ共有に関する代表的な研究として，検索ネットワークのトポロジに関するプロトコルや，複製の配置方式，複製更新の伝搬法を紹介し，本研究との関連性について議論する．

2.1 ピュア P2P 型のサービス検索ネットワーク

ピュア P2P 型におけるクエリを伝搬するサービス検索ネット

ワークのトポロジは，構造 (Structured) 型と非構造 (Unstructured) 型の 2 種類に分類される．本節では，それぞれのトポロジによるサービス提供ピアの検索を想定した代表的な研究について紹介する．ここで，構造型，非構造型のいずれの検索方法を用いたシステムにおいても，可用性の向上や負荷分散のためにデータの複製を作製し，ネットワーク上の複数のピア上に配置することが一般的である．従って，検索ネットワークのトポロジに関係なく，データの複製を配置する任意の P2P ネットワークに対して，本研究の提案方式は有効である．

2.1.1 構造型検索トポロジを用いたシステム

構造型検索トポロジの研究の中で代表的なものとしては，Chord [9]，Content-Addressable Network (CAN) [7]，Pastry [8]，Tapestry [10] などがある．これらの方式は，キーが配置されるピアを DHT (Distributed Hash Table) と呼ばれる方法を用いて厳密に決定する．DHT を用いた検索では，ハッシュ関数を用いてサービスおよびピアの識別子を変換し，あるアドレス空間に配置する．その空間内の位置が近いピアに各サービスのキーを配置する．サービスおよびピアの座標は，それらの識別子から一意に決定される．各ピアは，あるサービスに対する検索要求が到着した場合，そのサービスの識別子を，ハッシュ関数を用いてアドレス空間内での位置に変換し，空間内での位置がより近い隣接ピアにクエリを転送する．クエリの転送を行う検索ネットワークのトポロジ，すなわち各ピアが隣接ピアとして管理する情報の構成やキーの配置を厳密に定めることによって，検索を効率化している．

ピア同士が情報を伝達するためのネットワークを自律的に作成するという目的が共通している点で，本研究における更新伝搬木と，これらの構造型検索トポロジは類似している．しかし，検索トポロジは検索対象の一点を発見するため，情報伝達するピア数を少なくする効率的な構成の作成を目的としているのに対し，更新伝搬木は，全ピアに情報を伝達することを前提とした上で，遅延や負荷を軽減する構成を作ることを目的としている点が異なっている．

2.1.2 非構造型検索トポロジを用いたシステム

非構造型トポロジを用いたデータ共有サービスには Gnutella [4] や Freenet [3] などがある．非構造型トポロジを用いたシステムには，DHT のように，明確な方針に従って決定された検索トポロジは存在せず，クエリ伝搬のためのネットワーク構造と，キーの配置は独立している．そのため，構造型のトポロジを用いたネットワークのようにクエリを伝搬すべき隣接ピアが定まらず，検索は，フラッディングやその派生型などの無作為な検索方法が用いられる．

フラッディングでは，あるピアが必要なサービスを検索する場合，TTL (Time To Live) と呼ばれる値を定めたクエリを全ての隣接ピアにブロードキャストする．クエリを受け取ったピアがそのサービスのキーを保持している場合は，クエリが送られてきたピアへデータを保持していることを通知する．キーを保持していない場合は，クエリを発行したピアからの論理ネットワーク上のホップ数が TTL 値を超えないかぎり，さらにそのピアの隣接ピアへクエリをブロードキャストする．この方法は Gnutella など，複数の P2P データ共有サービスで使用されている．

Adamic らは，インターネット上で構成される P2P 論理ネットワークの構成は，べき法則 (Power-Law) に従った，べき法則ランダムグラフ (Power-Law Random Graph:PLRG) で表されることを示している [1]．i 番目のピアの隣接ピアの数を d_i とすると，d_i の分布が $d_i = i^{\alpha}$ で表されるとき，このネットワークはべき法則に従うという．

Lv らは，非構造な P2P ネットワークを用いたデータ共有サービスを想定し，クエリの伝搬方法として，フラッディングの派生型やデータの複製管理，実在ネットワークのトポロジについて議論している [6]．その中で代表的な複製の配置方法として，Freenet でも使用されている，パス複製法が提案されている．パス複製法では，検索成功時に，クエリを発行したピアからクエリに応答したピアの，P2P ネットワークの経路上にある全てのピアに複製を配置する．

2.2 複製更新の伝搬に関する研究

P2P 型ネットワークサービスに関する研究分野では，データの検索方法や複製の配置に関する研究が活発に行われているが，データに更新が発生するような環境における更新伝搬に関する研究は，これまでにほとんど行われていない．

Datta らは，文献 [2] において，P2P ネットワーク上の複製に対して，更新情報を伝搬する方法を提案している．この方法では，複製をもつ各ピアが任意に更新を行い，限定フラッディングという方法を用いて更新を伝搬する．限定フラッディングでは，各ピアは更新伝搬時に，隣接ピアの中から一定の割合のピアに更新を伝搬するという方法を繰り返し行う．さらに，一度更新を伝搬したピアは，部分リストと呼ばれるリストに加えられ，更新通知と一緒に隣接ピアに伝えられる．通知を受け取った各ピアは，リストにないピアの中から選択したピアに更新を伝搬する．しかしこの方法では，全てのピアに更新が伝搬されるという保障はされず，データの一貫性を保障することが困難である．さらに，異なる経路を通って無駄な通知が多数発生する．なお，文献 [2] では任意のピアが更新を行い，同時に更新が発生しない環境を想定しているのに対し，本研究では，更新はオリジナルノードのみで行うものと想定している．

3. 更新伝搬木に基づく複製更新伝搬法

本章では，木構造に基づいて複製の更新を伝搬する方式を提案する．

提案方式では，オリジナルノードを根とし，複製をもつ各ピアを内部節点とする木構造型論理ネットワーク (更新伝搬木) を用いて更新を伝搬する．これにより，負荷の分散と遅延の減少を両立する．更新伝搬木において，遅延の減少を考慮した場合，更新伝搬木は偏りのない完全 n 分木となることが望ましい．しかし，完全 n 分木を作成するためには，木構造をサーバなどが一元的に管理するか，複製をもつピア同士が頻繁に木の再構成に関するメッセージを交換する必要があり，ネットワークやピアの負荷が増大する．そのため，提案方式では，更新伝搬木に参加する各ピアは，更新伝搬木における，親ノードおよび子

ノードの情報のみを管理しておき，更新伝搬木上での参加位置を自律的に決定する．また，P2P ネットワークでは，複製の削除や，ピアの退出，故障によって更新伝搬木が分断される状況が発生する．そこで，そのような場合に，木を再構成する必要がある．

そこで本研究では，自律分散的に，完全 n 分木に近い更新伝搬木を作成するための方法として，新規に複製を作成する際の更新伝搬木への参加方法，および更新伝搬木が切断された場合の再構成の方法の一つとして，複製の削除時の方法を提案する．

3.1 更新伝搬木への新しいピアの参加

新しくデータを複製したピアは，そのデータの更新伝搬木へ参加する必要がある．ここで，新しく複製を配置したピアを新規ピアと呼ぶ．新規ピアの更新伝搬木への参加位置を決定するピアを，責任ノードと呼ぶ．最初に責任ノードとなるのは，オリジナルデータまたはその複製を所持し，新たに配置する複製の複製元となるピアとする．例えば，パス複製法を用いた場合，クエリに応答したピアが最初に責任ノードとなる．更新伝搬木への参加手順は次の通りである．

（1） 責任ノードは，自分自身が更新伝搬木中での根 (オリジナルノード) 以外のピアであれば，更新伝搬木内の自身の親にあたるピアに子の数 x を問い合わせる．$x < n$ の場合は，新規ピアをその親の子とし，手順を終了する．$x = n$，または自身が更新伝搬木の根のとき，手順 (2) へ進む．

（2） 自身の子の数 y を確認し，$y < n$ のときは，新規ピアを自身の子とし，手順を終了する．$y = n$ のとき，手順 (3) に進む．

（3） 自身の子の中からランダムに一つの子を選択し，その子を新規ピアの責任ノードとする．

新たに責任ノードとなったピアは，(2) 以降の手順に従って，新規ピアの更新伝搬木における位置を決定する．手順 (1) において，親の子の数が n より小さい場合に，新規ピアを親の子とするのは，更新伝搬木をできる限り完全 n 分木に近づけるためである．

$n = 2$ とし，2 分木を作成する場合の参加手順の動作例を，図 2 を用いて示す．図 2 は，最初の責任ノードの親の数，および責任ノード自身の子の数が n の場合を表す．図 2(a) のような構成において，ピア C の複製をもとにして，ピア H に複製が作られるものとする．このとき，ピア H の参加に関する最初の責任ノード C は，親ノード A，自身の子の数 x, y がともに n であるので，ピア H の参加に関する責任ノードを，ピア D, F のうちからランダムに選ぶ．ここではピア D を選んだものとする．ピア D は自身の子の数が n 未満であるので，ピア H をピア D の子とする (図 2(b))．

3.2 複製削除時における木の再構成

一般に，ピアのデータ記憶領域には限りがあるため，新しくデータの複製を作成する際に，他の複製を削除しなければならない場合がある．複製を削除する時には，削除したデータに関する更新伝搬木からそのピアは脱退する．その際，更新伝搬木が切断されてしまうと，脱退するピアの子孫は，以後に起こる更新に対して，更新情報を受け取ることができなくなる．そこで，ピアが脱退することにより切断された木を再構成する必要がある．木の再構成の手順は次の通りである．

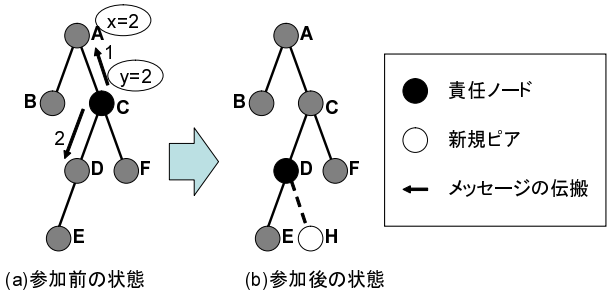

図 2 $x = n$ かつ $y = n$ の場合の参加例.
Fig. 2 Example of participation when $x = n$ and $y = n$.

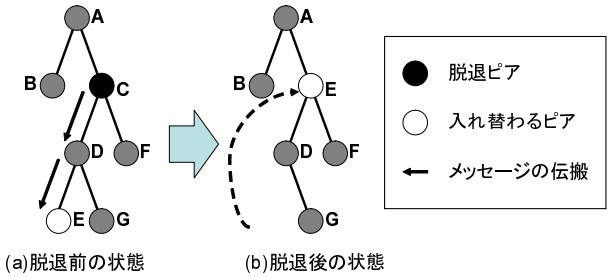

図 3 脱退ピアが内部節点の場合の脱退例.
Fig. 3 Example of exit when the peer is an internal node.

（1） 脱退するピア (脱退ピア) が更新伝搬木の葉のピアの場合，木から脱退することを親に通知する．親ノードは自身の子に関する情報から脱退ピアを削除し，それ以降は脱退ピアに更新を通知しない．脱退ピアは，親ピアの情報を削除し，手順を終了する．

（2） 脱退ピアが子をもつ場合，その中からランダムに一つの子を選択し，そのピアへ脱退希望を通知する．その通知を受けたピアは，葉ノードに達するまで同様の手順で子を選び，その通知を伝搬させる．通知が葉ノードに達したら，その葉ノードの伝搬更新木の位置を，脱退ピアがいた位置と入れ替える．その葉ノードの親ノードは，その葉ノードを自身の子に関する情報から削除し，手順 (3) へ進む．

（3） 入れ替えられたピアは，新たな位置における親と子にピアが入れ替わったことを通知する．その親と子のピアは，脱退ピアの情報を，入れ替えられたピアの情報に書き換える．その後，脱退ピアは親や子の情報を全て削除し，手順を終了する．

$n = 2$ とし，2 分木を作成する場合における脱退ピアが内部節点であったときの，木からの脱退手順の動作例を図 3 に示す．図 3(a) に示す更新伝搬木からピア C が脱退する場合，脱退希望をピア D，F のうち，一つのピアに伝搬する．ここではピア D を選択したものとする． 同様にピア D はピア E，G のうち，一つのピアを選び，脱退希望を伝搬する．ここではピア E を選んだものとする．ピア E は葉ノードであるので，脱退ピア C とピア E の木の位置を入れ替え，脱退を完了する (図 3(b))．

4. 性能評価

本章では，提案方式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．評価では，非構造な検索ネットワークトポロジを用いる，P2P ネットワークでのデータ共有を想定した．

4.1 想定環境

シミュレーション評価における想定環境について説明する．複製数に対する性能の変化を測定するため，P2P ネットワークに参加するピアの数は $100 \cdot k$ $(k = 1, 2, ..., 10)$ とし，それらは 2.1.2 項で述べた，PLRG 型の検索ネットワークを構成するものとした．ピア数が $100 \cdot k$ の場合，i 番目のピアの隣接ピア数を d_i とすると，d_i は以下のように与えた．

$$d_i = \left\lfloor 30 \cdot \left\{ 1 + 99 \frac{i-1}{100k-1} \right\}^{-0.4} \right\rfloor \quad (1)$$

このように d_i を設定することによって，総ピア数が変化しても，同様の密度の検索ネットワークとなるため，ピア数の変化に対する，性能の変化を正当に評価することができる．

また，全ピアのうち，ピア 1 から 100 までの 100 個のピアがそれぞれ異なるオリジナルデータ 1 から 100 をもつものとした．

次に，クエリを発行する確率は全てのピアで一定とし，各タイムスロットで 0.1 の確率とした．クエリは，100 種類のデータの中から 1 つのデータに，等しい確率でランダムに発行するものとした．クエリの伝搬にはフラッディングを用い，TTL の制限は行わないものとした．各ピアはクエリに応答したピアのうち，論理ネットワーク上の距離が最も近いピアに対してアクセスするものとした．複製の配置方法は，パス複製法を用いた．

各データのサイズは全て等しく，複製を保有可能な数は全てのピアで 10 とした．各ピアは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からランダムに一つを選択して削除し，新たな複製を作成するものとした．また，オリジナルデータは削除しないものとした．

ピアのネットワークからの退出や故障は発生しないものとした．更新が行われる確率は全てのデータに対して等しく，各タイムスロットで 0.1 とした．

以上のようなシステム環境において，100,000 タイムスロットのシミュレーション実験によって，提案方式の性能評価を行った．比較対象としては，放射伝搬法と直線伝搬法を用いた．以下では，シミュレーション評価の結果を示し，考察を行う．

4.2 n=2 での評価

はじめに，$n = 2$ とし，2 分木を作成した場合の，ピア数の変化に対する平均負荷と平均遅延についての評価結果を示す．

4.2.1 ピア数に対する負荷の変化

更新伝搬時に，各ピアが次に更新を伝搬したピア数の平均値 (平均負荷) を図 4 に示す．図中の ‘Tree’ は提案方式，‘Line’ は直線伝搬法，‘Radial’ は放射伝搬法の結果を表す．直線伝搬法を用いた場合，各ピアは複製をもつ 1 つのピアに更新を伝搬するので，平均負荷は常に 1 となっている．放射伝搬法を用いた場合は，平均負荷がピア数に従って線型増加している．これは，ネットワークのピア数の増加に伴い，オリジナルノードが更新を伝搬しなければならないピアの数が線型増加するためである．一方，提案方式では，一度の転送で更新を伝搬するピアの数は，最高でも 2 であり，ピア数が増加しても平均負荷が 2 以下になることがわかる．

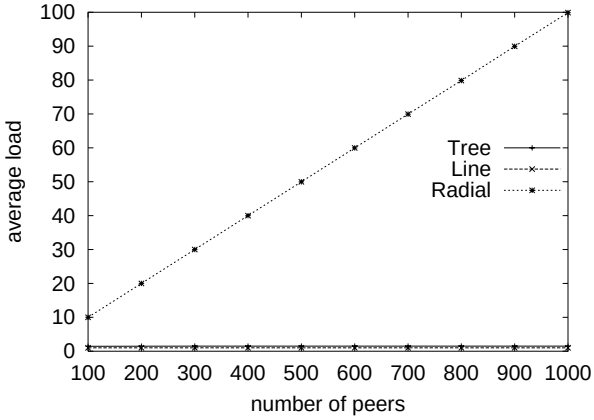

図 4　ピア数と平均負荷 ($n = 2$).
Fig. 4　Number of peers vs. average load ($n = 2$).

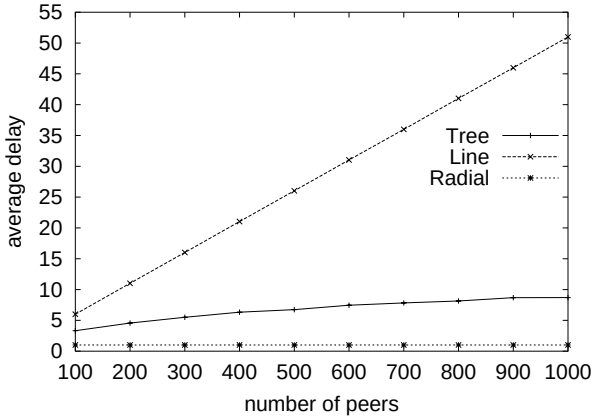

図 5　ピア数と平均遅延 ($n = 2$).
Fig. 5　Number of peers vs. average delay ($n = 2$).

4.2.2 ピア数に対する遅延の変化

オリジナルノードから各ピアへ更新が伝搬される際に要した論理ホップ数の平均 (平均遅延) を図 5 に示す．放射伝搬法を用いた場合，オリジナルノードが複製をもつ全ピアに一度に更新を伝搬するので，平均遅延は常に 1 である．直線伝搬法を用いた場合，ネットワークのピア数が増加するにつれて，更新情報を伝搬する回数が大きくなるので，平均遅延が線型増加している．一方，提案方式では，平均遅延をピア数に対して，log オーダで抑えられていることがわかる．このことから，提案方式によって，ほぼ偏りのない木構造型の論理ネットワークを構成できていることがわかる．

図 4 および図 5 から，提案方式は，オリジナルノードが全ピアに更新を伝搬させる放射伝搬法と比べて負荷を分散し，複製をもつピアに 1 つずつ更新情報を伝搬する直線伝搬法に比べて遅延を大幅に減少することがわかる．この結果から，提案方式は，負荷の分散と遅延の減少を両立できていることを確認できる．

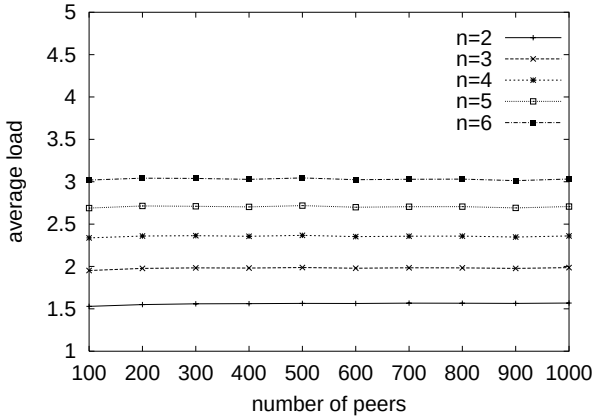

図 6　n と平均負荷.
Fig. 6　n vs. average load.

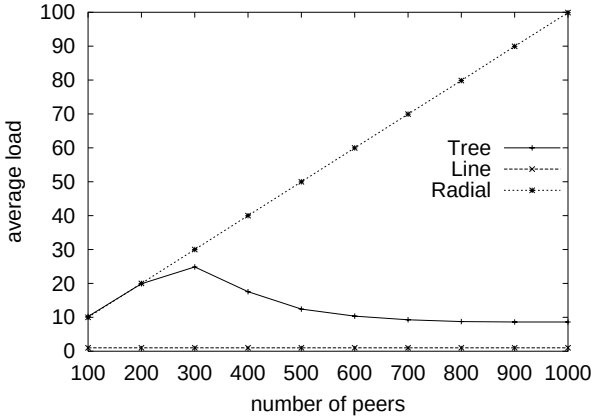

図 8　ピア数と平均負荷 ($n = 50$).
Fig. 8　Number of peers vs. average load ($n = 50$).

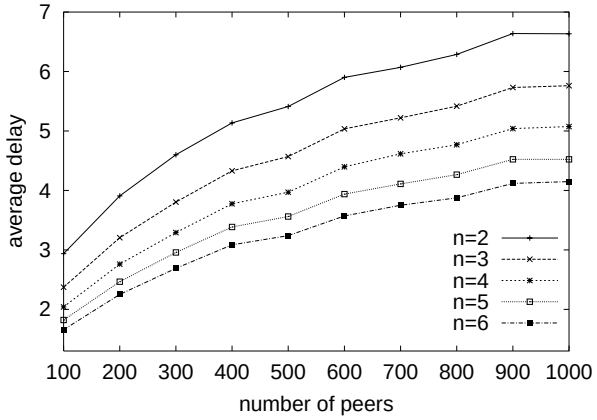

図 7　n と平均遅延.
Fig. 7　n vs. average delay.

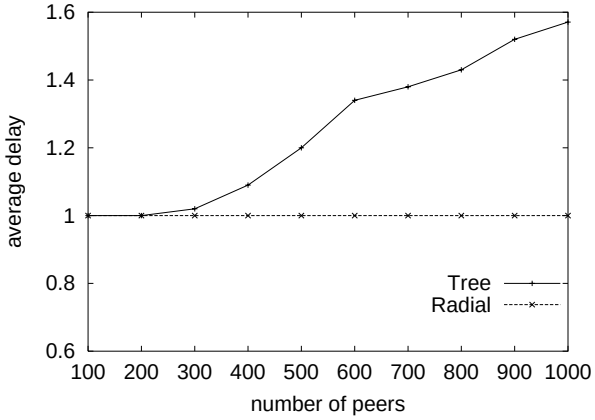

図 9　ピア数と平均遅延 ($n = 50$).
Fig. 9　Number of peers vs. average delay ($n = 50$).

4.3　n の影響

本節では，n を変化させた場合の，提案方式の平均負荷と平均遅延に対する影響を調べる．

まず，提案方式における n を 2 から 6 まで変化させた場合の，平均負荷を図 6 に示す．提案方式は，n=∞の場合が放射伝搬法と等価となり，n=1 の場合が直線伝搬法と等価となる．n が増加するほど，平均の子の数の増加するので，一度に更新を伝搬するピア数は増加する．つまり，平均負荷が増加することを確認できる．同様に，提案方式における n を 2 から 6 まで変化させた場合の，平均遅延を図 7 に示す．n が増加するほど，更新伝搬木の高さの平均は小さくなるため，平均遅延が減少することを確認できる．この結果から，n が増加した場合も偏りのない木を構成できていることが確認できる．

図 6 および図 7 の結果から，n の値を変化させることによって平均負荷と平均遅延のバランスを調整できることがわかる．n の値は，各ピアが許容可能な負荷によって，システムもしくはピア毎に設定することが有効である．

次に，n の値を大きくした場合 ($n = 50$) において，4.2 節と同様の評価を行った．この場合の提案方式の平均負荷を図 8 に示す．ピア数が 200 までは，放射伝搬法とほぼ同じ値をとっている．これは，提案方式では新規ピアの木への参加位置を決定する際に，責任ノードの親の子に空きがあれば親の子とするため，各データに関して，複製数が 50 を超えない場合は木の高さが常に 1 になり，放射伝搬法の場合と等しくなるためである．

しかし，ピア数が 300 の場合では放射伝搬法に比べて負荷が減少している．これは，各データの複製数が 50 を超え，木の高さが 2 になり，負荷の非常に少ないピアが存在する状況が発生するためである．さらに，ピア数が 400 以上になると，ピア数の増加に対して負荷が減少している．これは，ピア数が増えるほど複製数が 50 を超えることが多く，木の高さが 2 になり，負荷の少ない内部節点が増加するためである．

提案方式における，$n = 50$ とした場合の平均遅延を図 9 に示す．図 9 では，提案方式の平均遅延の変化を明確に示すため，直線伝搬法での結果は省略した．複製の数が 50 を超えると，高さが 2 のピアがでてくるため遅延は増加するが，ピア数が 1000 程度の場合の複製数では木の高さはそれ以上大きくならず，平均遅延は log オーダを示さない．

このことから，複製数が少ない場合に，n の値を大きくすると，木構造は構成されず負荷分散と遅延減少を両立できないことがわかる．従って，複製の数に合わせて n の値を決定する必要がある．

5. 提案方式の拡張

実環境において提案方式を運用する場合，現状では様々な問題が発生する．本章では，それらの問題に対応するための，提案方式の拡張について考察する．

5.1 ピアの退出，故障時の更新伝搬木の再構成

本研究の性能評価では，ピアの退出や突然の故障については考慮していない．しかし，実際の P2P ネットワークを用いたデータ共有サービスでは，ピアの参加，退出は頻繁に発生する上，故障など，ピアが正当な手続きを踏まずに退出する場合 (以下ではこれらをまとめて，ピアが故障した場合と称する) も多い．そのような場合における，更新伝搬木の再構成についても検討する必要がある．

ピアが P2P ネットワークから正式な手続きを踏んで退出する場合，退出するピアは退出時に自身が所持している全複製に対して，複製削除時と同様に更新伝搬木から脱退する手順を行うことで，更新伝搬木を正常に保つことができる．

ピアが故障した場合，正当な手続きを踏む退出時と異なり，故障ピアは更新伝搬木からの脱退の手順を行うことが不可能である．このような場合において，更新伝搬木を再構成するには次のような方法が考えられる．

現状の提案方式において，各ピアが保持する，同じ複製をもつ他ピアの情報は，更新伝搬木内の自身の親ノードと子ノードのみである．そこで，この範囲を拡大することを考える．具体的には，親ノードのさらに親ノード，また子ノードのさらに子ノードの情報も保持するようにする．これにより，更新伝搬時に故障ピアの存在を知った故障ピアの親ノードにあたるピアは，故障ピアの代わりに，故障ピアの子ノードに対して故障ピアの脱退手順を行うことができる．このようにして故障時に木の再構成を行い，再構成後に更新を伝搬すればよい．

5.2 P2P ネットワークへの復帰

退出や故障したピアが P2P ネットワークに復帰する場合には，以前所持していた複製は更新前の古いデータである可能性が高い．そこで，ネットワークへの復帰時に，即座に更新情報を受け取り，複製を更新する必要がある．

このような場合，ピアの退出時や故障時には，所持していた各複製に関して，それの更新伝搬木中の他ピアの情報を削除せずに保持しておくことが有効である．復帰時には，その情報の中から復帰時にも木に参加しているピアがある場合は，そのピアから更新情報を受け取り，複製配置時と同様の手順で新たに新規ピアとして木に参加する．

ただし，退出または故障してから復帰するまでの時間により，他ピアが，複製を削除してしまっていた場合など，既に木に参加しているピアがいないという状況が考えられる．この場合，正しく更新が受け取れなかった複製に関しては削除するか，もしくはその複製に対するクエリを発行し，それを保持するピアを検索するなどの方法が考えられる．

5.3 更新伝搬木に偏りが発生した場合への対応

本研究の提案方式では，更新伝搬木への参加は，責任ノードが親ノードの子の数を確認することで，自律分散的に完全 n 分木を作成しようと試みる．しかし，ピアの更新伝搬木への参加および脱退時に，ランダムに子を選択するという特徴から，更新伝搬木がバランスのとれた状態を保てない状況が考えられる．例えば，2 分木において内部節点の片方のノードが多い場合，この木はバランスの悪い木である．このような状況においては，完全 2 分木と比較して大きな遅延が生じてしまい，意図した遅延と負荷を保つことができない．その上，木の片方にノード数が偏った場合，さらにその部分にアクセスが集中し，複製がそれらのピアから作製されるため，偏りがさらに大きくなる．そこで，状況に応じて木の再構成を行い偏りを解消することで，更新伝搬木をより完全 n 分木に近づけ，遅延を軽減する必要がある．

5.1 節で述べたように現状の提案方式では，各ピアが保持する，同じ複製をもつ他ピアの情報は，更新伝搬木内の自身の親ノードと子ノードのみである．そこで，この範囲を拡大し，新規ピアが更新伝搬木に参加する際に，現状よりもさらに木の上の位置に参加するようにすることで，木の偏りを解消することができる．

6. おわりに

本研究では，P2P ネットワーク内で共有されるデータに更新が発生し，更新情報を複製をもつ全ピアに即座に通知する必要がある環境を想定して，データの更新伝搬時における各ピアの負荷分散と遅延減少を両立する方式を提案した．提案方式では，オリジナルノードを根とし，複製をもつその他のピアを内部節点とした n 分木の更新伝搬木を構成する．更新発生時には，この更新伝搬木に従って更新を伝搬する．さらに，複製を作成することでピアが更新伝搬木に参加する場合や，複製を削除するなどピアが更新伝搬木から脱退する場合に，更新伝搬木に参加しているピアが自律分散的に木を再構成する方法を提案した．

また，本研究では，シミュレーション実験により，提案方式の性能評価を行った．その結果から提案方式は，ピア数の増加に対して平均負荷を定数オーダに抑えると同時に，平均遅延を log オーダに抑えることを確認した．

実際の P2P ネットワークを用いたデータ共有サービスでは，ピアの退出や故障が頻繁に発生する．そのような場合は，そのピアは自身がもつ複製に関する更新伝搬木から脱退し，木を再構成する必要がある．また，退出や故障したピアが P2P ネットワークに復帰する場合には，以前所持していた複製が古いデータである可能性が高い．本研究では，このような場合に対応するための提案方式の拡張について考察した．

今後は，各データに対するアクセスの発生頻度や各データの更新頻度，各ピアが保持可能な複製数などを変化させ，実際のデータ共有サービスを考慮した様々な環境における性能評価を行う必要がある．また，ピアの P2P ネットワークからの退出および故障，またネットワークに復帰した場合，さらに，更新伝搬木に偏りが生じた場合などにおける提案方式の拡張について，具体的な手順を検討する必要がある．

謝　　辞

本研究は，文部科学省 21 世紀 COE プログラム「ネットワーク共生環境を築く情報技術の創出」，特定領域研究 (15017262) および科学技術振興調整費「モバイル環境向 P2P 型情報共有基盤の確立」の研究助成によるものである．ここに記して謝意を表す.

文　　献

[1] L. A. Adamic, R. M. Lukose, A. R. Puniyani, and B. A. Huberman, “Search in Power-Law Networks,” Physical Review E., vol. 64, no. 4, 046135, 2001.

[2] A. Datta, M. Hauswirth, and K. Aberer, “Updates in Highly Unreliable, Replicated Peer-to-Peer Systems,” Proc. ICDCS’03, pp. 76-85, 2003.

[3] FreeNet, <URL: http://freenet.sourceforge.net>.

[4] Gnutella, <URL: http://www.gnutella.com>.

[5] J. Kubiatowicz, D. Bindel, Y. Chen, S. Czerwinski, P. Eaton, D. Geels, R. Gunnadi, S. Rhea, H. Weatherspoon, W. Weimer, C. Wells, and B. Zhao, “OceanStore: An Architecture for Global-Scale Persistent Storage,” Proc. ASPLOS2000, pp.190-201, 2000.

[6] Q. Lv, P. Cao, E. Cohen, K. Li, and S. Shenker, “Search and Replication in Unstructured Peer-to-Peer Networks,” Proc. ICS’02, pp. 84-95, 2002.

[7] S. Ratnasamy, P. Francis, M. Handley, R. Karp, and S. Shenker, “A Scalable Content-Addressable Network,” Proc. SIGCOMM’01, pp. 161-171, 2001.

[8] A. Rowstron and P. Druschel, “Pastry: Scalable, Distributed Object Location and Routing for Large-scale Peer-to-Peer Systems,” Proc. Middleware 2001, pp. 329-350, 2001.

[9] I. Stoica, R. Morris, D. Karger, M. F. Kaashoek, and H. Balakrishnan, “Chord: A Scalable Peer-to-Peer Lookup Service for Internet Applications,” Proc. SIGCOMM’01, pp. 149-160, 2001.

[10] B. Zhao, J. Kubiatowicz, and A. Joseph, “Tapestry: An Infrastructure for Wide-area Fault-tolerant Location and Routing,” U. C. Berkeley Technical Report UCB//CSD-01-1141, 2000.