P2Pネットワークにおける
更新データ伝播に関する研究

2010年1月

渡辺　俊貴

P2Pネットワークにおける
更新データ伝播に関する研究

提出先　　大阪大学大学院情報科学研究科
提出年月　2010年1月

渡辺　俊貴

関連発表論文

1. 学会論文誌発表論文

1. 渡辺俊貴, 原 隆浩, 木戸裕樹, 中通 実, 西尾章治郎: P2P ネットワークにおける木構造に基づく複製更新伝播法, 情報処理学会論文誌, Vol. 48, No. 2, pp. 527–538 (Feb. 2007).

2. 渡辺俊貴, 神崎映光, 原 隆浩, 西尾章治郎: 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, 情報処理学会論文誌, Vol. 49, No. 6, pp. 1819–1832 (June 2008).

3. 渡辺俊貴, 神崎映光, 原 隆浩, 西尾章治郎: P2P ネットワークにおけるデータの更新量を考慮した更新伝播手法の提案, 日本データベース学会論文誌, Vol. 7, No. 3, pp. 13–18 (Jan. 2009).

4. 小林由依, 渡辺俊貴, 神崎映光, 義久智樹, 原 隆浩, 西尾章治郎: P2P ネットワークにおける動的クラスタを用いた検索手法, 情報処理学会論文誌, Vol. 51, No. 3 (Mar. 2009, 採録決定).

2. 国際会議等発表論文

1. Watanabe, T., Hara, T., Kido, Y., and Nishio, S.: An Update Propagation Strategy for Delay Reduction and Node Failure Tolerance in Peer-to-Peer Networks, in *Proceedings of International Symposium on Frontiers in Networking with Applications (FINA 2007)*, pp. 103–108 (May 2007).

2. Watanabe, T., Kanzaki, A., Hara, T., and Nishio, S.: An Update Propagation Strategy Considering Data Access Frequency in Peer-to-Peer Networks, in *Proceedings of International Conference on Database Systems for Advanced Applications (DASFAA 2008)*, pp. 661–669 (Mar. 2008).

3. Miyazaki, T., Watanabe, T., Kanzaki, A., Hara, T., and Nishio, S.: Keyword Search Considering User's Preference in P2P Networks, in *Proceedings of 3rd International Conference on Ubiquitous Information Management and Communication (ICUIMC 2009)*, pp. 462–470 (Jan. 2009).

4. Watanabe, T., Kanzaki, A., Hara, T., and Nishio, S.: Update Propagation Strategies Considering Degree of Data Update in Peer-to-Peer Networks, in *Proceedings of International Conference on Database Systems for Advanced Applications (DASFAA 2009)*, pp. 328–333 (Apr. 2009).

5. Watanabe, T., Kanzaki, A., Hara, T., and Nishio, S.: A Selective Update Propagation Based on Degree of Data Update in Peer-to-Peer Networks, in *Proceedings of International Conference on Network-Based System (NBiS 2009)*, pp. 52–59 (Aug. 2009).

6. Watanabe, T., Zhao, Y., Kanzaki, A., Hara, T., and Nishio, S.: A Replica Relocation Method for Improving Search Efficiency in P2P Networks, in *Proceedings of International Conference on Advances in P2P Systems (AP2PS 2009)*, pp. 13–18 (Oct. 2009).

7. Kobayashi, Y., Watanabe, T., Kanzaki, A., Yoshihisa, T., Hara, T., and Nishio, S.: A Dynamic Cluster Construction Method Based on Query Characteristics in Peer-to-Peer Networks, in *Proceedings of International Conference on Advances in P2P Systems (AP2PS 2009)*, pp. 168–173 (Oct. 2009).

3. 研究会等発表論文（査読付）

1. 渡辺俊貴, 木戸裕樹, 原 隆浩, 西尾章治郎: P2P ネットワークにおける障害耐性向上と遅延減少のための木構造に基づく複製更新伝播について, 情報処理学会マルチメディア, 分散, 協調とモバイルシンポジウム (DICOMO 2006) 論文集, Vol. 2006, No. 6, pp. 313–316 (July 2006).

2. 渡辺俊貴, 神崎映光, 原 隆浩, 西尾章治郎: P2P ネットワークにおけるデータの使用

頻度を考慮した木構造に基づく複製更新伝播法, 電子情報通信学会データ工学ワークショップ (DEWS 2007) 論文集 (Feb./Mar. 2007).

3. 趙 勇, 渡辺俊貴, 神崎映光, 原 隆浩, 西尾章治郎: P2P ネットワークにおけるクエリ統計情報を利用した自律分散型複製配置方式, 情報処理学会マルチメディア, 分散, 協調とモバイルシンポジウム (DICOMO 2007) 論文集, Vol. 2007, No. 1, pp. 384–391 (July 2007).

4. 宮崎知美, 渡辺俊貴, 神崎映光, 原 隆浩, 西尾章治郎: P2P ネットワークにおけるユーザの嗜好を考慮した検索方法, 情報処理学会マルチメディア, 分散, 協調とモバイルシンポジウム (DICOMO 2007) 論文集, Vol. 2007, No. 1, pp. 392–399 (July 2007).

5. 宮崎知美, 渡辺俊貴, 神崎映光, 原 隆浩, 西尾章治郎: P2P ネットワークにおけるユーザの嗜好特性を考慮した検索方法に関する一考察, 情報処理学会マルチメディア, 分散, 協調とモバイルシンポジウム (DICOMO 2008) 論文集, Vol. 2008, No. 1, pp. 668–675 (July 2008).

4. その他の研究会等発表論文

1. 渡辺俊貴, 神崎映光, 原 隆浩, 西尾章治郎: 非構造型 P2P ネットワークにおける情報爆発を考慮した更新伝播に関する一考察, 情報処理学会 第 70 回全国大会論文集, Vol. 5, pp. 9–10 (Mar. 2008).

2. 渡辺俊貴, 神崎映光, 原 隆浩, 西尾章治郎: P2P ネットワークにおけるデータの更新量を考慮した更新伝播木管理手法, 情報処理学会研究報告 (データベースシステム研究会 2008-DBS-146), Vol. 2008, No. 88, pp. 85–90 (Sept. 2008).

3. 小林由依, 渡辺俊貴, 神崎映光, 義久智樹, 原 隆浩, 西尾章治郎: P2P ネットワークにおける動的なカテゴリ生成を考慮した検索手法, データ工学と情報マネジメントに関するフォーラム (DEIM 2009) 論文集 (May 2009).

4. 渡辺俊貴, 趙 勇, 神崎映光, 原 隆浩, 西尾章治郎: P2P ネットワークにおけるデータの分布を考慮した複製再配置, 電子情報通信学会 (データ工学研究会 DE2009-9), Vol. 109, No. 153, pp. 49–54 (July 2009).

以上

内容梗概

近年の計算機の高性能化やネットワークインフラの発達により，Peer-to-Peer (P2P) ネットワークへの関心が高まっている．P2P ネットワークでは，クライアント，サーバのような明確な役割分担はなく，各端末（ピア）が互いにサービスを提供しあうことにより，単一障害点がなくなり，高いネットワーク耐性やスケーラビリティを実現できる．

P2P ネットワークを利用したサービスでは，検索効率やデータの可用性向上，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と考えられている．一方，データ共有や分散 Web コンテンツ，ニュース速報や掲示板などのサービスでは，複数のピアで共有され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，アプリケーションもしくはユーザが，更新前の古い複製にアクセスする可能性がある．そのため，古い複製へのアクセスが許されない，もしくは，最新のデータとの許容誤差が決められているアプリケーションでは，複製を所持するピアに対して適切に更新データを伝播させる必要がある．しかし，これまでに考えられてきた手法では，更新データを複製所持ピアに確実に伝播させる保証がなかったり，更新伝播時の負荷や遅延，更新データの必要性やデータの更新内容については考慮されていない．

そこで本論文では，更新データを必要としているピアに確実に伝播させつつ，更新伝播時の負荷の軽減と分散，遅延の減少を目的とする効率的な更新伝播手法について議論する．具体的には，同一のデータの複製を所持しているピアで一つの木構造型論理ネットワーク（更新伝播木）を構築し，この木に沿って更新データを伝播させる手法を提案する．次に，この手法を拡張し，各複製所持ピアがどの程度，その複製を使用しているかを考慮し，データの使用頻度に応じて伝播させる更新情報を変更する手法を提案する．最後に，データが更新された場合に，その更新の大きさ（更新量）に応じてその更新データを伝播させるピアを制限する手法を提案する．

本論文は5章から構成され，その内容は次の通りである．まず，第1章で序論として研究の背景および動機について述べる．第2章において，木構造型の論理ネットワーク（更新伝播木）を用いて更新データを伝播させる手法について述べる．この手法では，同じデータの複製を所持しているピアで一つの更新伝播木を構築し，その木に沿って更新データを伝播させる．これにより，更新伝播時の負荷を分散させつつ，更新伝播時の遅延を減少させる．また，ピアの参加や脱退が発生した際には，木構造上の各ピアが，自身が管理してい

る周辺ピアの情報のみを用いて局所的に木を修復することで，木構造の維持に必要なメッセージ数を抑制する．さらに，提案手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し，その有効性について検証する．

第3章では，第2章の提案手法を拡張し，データアクセス頻度を考慮した更新伝播手法について述べる．この手法では，複製所持ピアが所持している複製が使用される頻度（データアクセス頻度）を調べ，データアクセス頻度が高いピアには更新データを伝播させる．一方，データアクセス頻度が低いピアには，所持している複製が古くなった旨を通知するためのサイズの小さなメッセージのみを伝播させることにより，更新伝播時のトラヒックを削減する．また，提案手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し，その有効性について検証する．

第4章では，データの更新量を考慮した更新伝播手法について述べる．この手法では，各複製所持ピアが，どの程度の大きさの更新が発生した場合に更新データを受信するかという条件を設定し，この条件に応じて複数の木構造を構築する．データが更新された場合には，その更新データを必要としていないピアへの更新データの伝播を抑制することにより，更新伝播時のトラヒックを削減する．また，提案手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し，その有効性について検証する．

第5章では，本論文の成果を要約したのち，今後の研究課題について述べ，本論文のまとめとする．

目 次

第1章

序章

1.1 研究の背景

近年の計算機の高性能化やネットワークインフラの発達により，Peer-to-Peer (P2P) ネットワークへの関心が高まっている [3, 29, 32, 33, 37, 49, 59]. Web サービスなどの既存のネットワークサービスで最も主流となっているクライアント・サーバモデルでは，単一のサーバが全てのサービスを提供するため，クライアント数の増加に伴い，サーバの処理負荷が増大する．また，サーバに障害が発生した場合，サービス全体が停止する．一方，P2P ネットワークでは，クライアント，サーバのような明確な役割分担はなく，各端末（ピア）が互いにサービスを提供しあうことにより，単一障害点がなくなり，高いネットワーク耐性やスケーラビリティを実現できる．

P2P モデルを用いたネットワークサービスとしては，データ共有[1] [2] やインスタントメッセージング[3] [4]，グループウェア[5] などがある. これらの中で最も代表的な利用例として，データ共有サービスが挙げられる．例えば，不特定多数のユーザが参加し，各ユーザが所持するコンテンツ（文書，画像，動画など）を共有するサービスや，企業や特定のコミュニティが開発しているソフトウェアなどの大規模ファイルの共有サービスが存在する．また，地震情報や交通情報など，多くのユーザがリアルタイムに情報を要求するサービスに

[1] BitTorrent, *http://www.bittorrent.com.*

[2] FolderShare, *https://www.foldershare.com.*

[3] MSN メッセンジャー, *http://messenger.live.jp.*

[4] Skype, *http://www.skype.com.*

[5] Office Groove, *http://office.microsoft.com/ja-jp/groove/FX100487641041.aspx.*

おいても，サーバに負荷をかけることなく情報を得ることができるP2Pモデルが利用される．さらに，近年，企業内クラウドコンピューティング環境の構築に関する取り組みも進められており，企業内の端末を利用した企業内データベースの構築や共有，余っているPCなどの計算機資源を利用した業務用データの共有などにもP2Pモデルが有効である．これらのデータ共有サービスでは，それぞれのピアがキャッシュ領域にデータを保持し，ピア同士が互いにデータを提供し，データ共有を行う．そのため，P2Pネットワークにおいて，データ共有サービスを対象とした，もしくはデータ共有サービスに適応できるシステムの研究 [56, 78] が行われている．

1.2 P2Pネットワーク

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するため，各ピアが要求するサービス（データ）の提供者を検索する機構が必要となる．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機構によって，クライアント・サーバモデル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．

ハイブリッドP2P型のネットワークサービスでは，サービスの提供が可能なピアを検索するためにクライアント・サーバモデルを用いる．ハイブリッドP2P型のネットワークサービスとしては，KaZaA [28], OpenNap [46], BitTorrent [48] などがある．このようなP2Pネットワークサービスでは，検索サービスを提供するサーバが，ネットワーク上の全ピアの識別子（例：IPアドレス）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に管理する．ピアがサービスを検索する場合，サーバに問合せを行い，サービスの提供が可能なピアを発見する．サービス提供ピアの情報を得たあとは，ピア同士の直接通信によってサービスの提供を受ける．ハイブリッドP2P型のネットワークの構成例を図1.1(a)に示す．この図において，円が各ピアを表し，ピア間の直線が論理ネットワーク上のリンクを表す．このようにハイブリッドP2P型のネットワークサービスでは，サーバに問合せを行うだけで，容易にサービスの提供が可能なピアを発見できる．しかし，限られたサーバのみに検索の機能が集中しているため，サーバが故障もしくは停止した場合，システム全体が停止する．また，完全な分散環境ではないため，ネットワークに参加するピア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・サーバモデルの問題点を完全に解決できない．

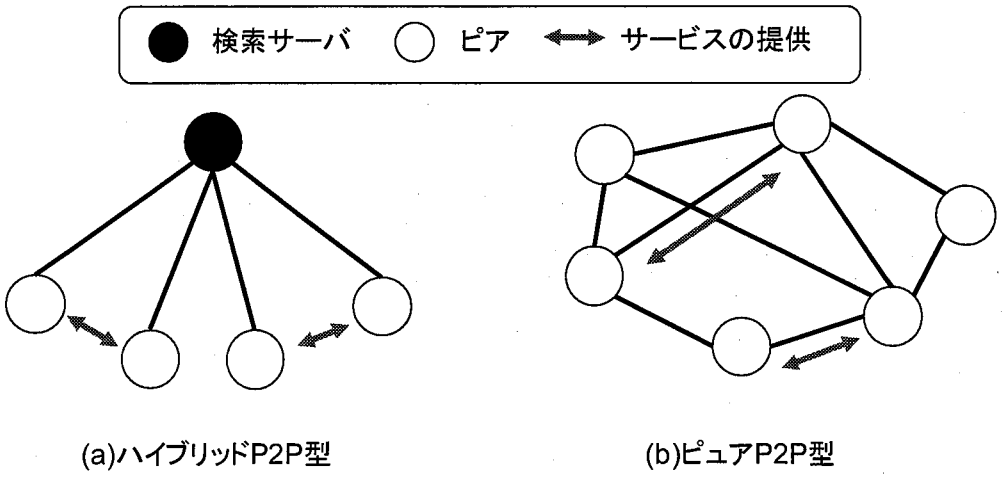


図 1.1: ハイブリッド P2P 型とピュア P2P 型

一方，ピュア P2P 型のネットワークサービスでは，サービスの提供が可能なピアの検索を完全な分散環境で行う．ピュア P2P 型のネットワークサービスとしては，FreeNet [14], Gnutella [22], JXTA [27] などがある．このような P2P ネットワークサービスでは，サービスの提供者を自律分散的に検索するため，ネットワークに参加するピア数に対するスケーラビリティを得ることができる．ピュア P2P 型のネットワークの構成例を図 1.1(b) に示す．本研究では，P2P ネットワークサービスの利用者が急速に増加していることを考慮して，不特定多数のユーザ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 型のネットワークサービスを想定する．また，ピュア P2P 型のネットワークサービスでは，サービス提供者の発見確率や検索時間がネットワークの構成（トポロジ）に大きく依存する．ピュア P2P 型ネットワークサービスにおけ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) 型に分類される．

構造型トポロジを用いた P2P システムは，ネットワークのトポロジやデータを検索するために必要なキー（データとデータを提供するピアの識別子の組）の配置を厳密に決定するシステムである．構造型トポロジを用いた P2P システムとしては，Kademlia [38], Content Addressable Network (CAN) [51], Pastry [54], Chord [58], Tapestry [82] などがある．このようなシステムでは，DHT (Distributed Hash Table) を用いて，ピアの識別子にハッシュ関数を適用してハッシュ値を算出し，あるアドレス空間に配置する．また，各ピアが所持するデータのキー情報も同様に，データ名やデータサイズなどを基にハッシュ値を算出し，アド

レス空間内の位置が近いピアにそのキー情報を配置する．データを検索する場合は，データの識別子を基にハッシュ値を求め，アドレス空間内での位置がより近いピアにクエリを転送する．このようにして，データの検索を効率化できる．また，構造型トポロジを用いたP2Pシステム上で，範囲検索などの柔軟な検索を行うための研究も行われている [2,8,18,23,57]．しかし，構造型トポロジを用いたP2Pシステムでは，各ピアが所持するデータやどのピアを隣接ピアとするかに制約があり，柔軟さに制限がある．

一方，非構造型トポロジを用いたP2Pシステムでは，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が存在しないため，データの検索には，フラッディングやその派生型 [11,39,43] 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，TTL (Time To Live) と呼ばれる値を設定したクエリを全ての隣接ピアに送信する．クエリを受け取ったピアは，そのデータを所持していなければ，クエリ発行元ピアからの論理ネットワーク上のホップ数がTTLの値を超えない限り，自身の隣接ピアへとクエリを伝播させていく．データを所持している場合は，クエリが送られてきたピアに，データを所持していることを通知する．非構造型トポロジを用いたP2Pシステムでは，無作為な検索によるトラヒックや検索にかかる遅延が大きいが，トポロジに制約がないため，柔軟に任意の論理ネットワークを構築できる．そのため，キーワードの部分一致検索だけでなく，データの類似性やユーザの嗜好を考慮したデータの検索方法に関する研究 [20,40,41,42] も盛んに行われている．

1.3 研究内容

実環境におけるP2Pネットワークでは，ピアの参加，脱退が頻繁に発生する．P2Pネットワークでは，各ピアが互いにデータ提供者の役割を担うため，ピアが脱退するとそのピアの所持するデータにアクセスできなくなる．特に非構造型トポロジを用いたP2Pシステムでは，どのピアがどのデータを所持しているかといった明確な情報を得ることができない．また，1.2節で述べたように，非構造型トポロジを用いたP2Pシステムにおけるデータ検索で利用されるフラッディングでは，クエリ発行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，発行されるクエリ数が指数関数的に増加する．したがって，少ないホップ数で目的のデータを発見することにより，検索時のメッセージ数を削減できる．以上の理由より，構造型，非構造型のいずれのトポロジを用いたP2Pシステムにおいても，データの可用性や検索効率の向上，データを所持しているピアの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ことが有効である [15,16,36,45,77]. 一方，各ピアが所持できるデータ量には限界があるため，ネットワーク内の全てのデータの複製を所持することはできない．そこで，データの複製の配置手法に関して様々な研究が行われている [9,24,30,75,76,81].

また，P2P ネットワークサービスに関する研究分野では，複数のピアで共有されるデータに更新が発生する環境において，効率的に更新を伝播させるための研究がいくつか行われている [17,35,44,50,55,62,80]. 1.1 節で挙げたデータ共有サービスにおいても，各ユーザが所持しているコンテンツが更新されたり，ソフトウェアのアップデートが行われるなど，複数のピアで共有されているデータに更新が発生する場合が多い．データに更新が発生する環境では，更新データの伝播が遅れたり伝播できなかった場合，アプリケーションもしくはユーザが更新前の古い複製にアクセスする可能性がある．そのため，古い複製へのアクセスが許されなかったり，最新のデータとの許容誤差が決められているアプリケーションでは，複製を所持するピアに対して，適切に更新データを伝播させる機構が必要となる．しかし，これまでに提案された手法では，複製を所持しているピアに確実に更新データを伝播させる保証がなかったり，更新伝播時の各ピアの負荷や遅延，データの使用頻度や更新の内容についてはあまり考慮されていなかった．

そこで本研究では，P2P ネットワークにおける効率的な更新データ伝播手法の実現を目的とする．具体的には，複製を所持するピアで構築される木構造に沿って更新データを伝播させる UPT (Update Propagation Tree) 法，各ピアのデータアクセス頻度を考慮して更新データを伝播させる UPT-HL (UPT with H peers & L peers) 法，データの更新内容に応じて更新データを伝播させるピアを制限する UPDD-S (Update Propagation strategy considering Degree of Data update with Same-condition trees) 法および UPDD-SO (UPDD with Same-condition trees and Ordered-condition trees) 法を提案する．これらの手法により，更新データを必要としているピアに確実にその更新データを伝播させると同時に，更新伝播時のトラヒックや遅延を軽減する．

1.4 本論文の構成

本論文は 5 章から構成され，その内容は次の通りである．まず，第 2 章では，同じデータを所持するピアで一つの木構造を構築し，その木に沿って更新データを伝播させる，木構造を用いた複製更新伝播法（UPT 法）を提案する．この手法では，更新伝播時の負荷分

散と遅延減少を実現しつつ，同時に，耐障害性を向上させている．さらに，提案手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し，その有効性について検証する．

第3章では，第2章で提案したUPT法を拡張し，各複製所持ピアのデータの使用頻度（データアクセス頻度）に着目して更新データを伝播させるUPT-HL法を提案する．この手法では，各複製所持ピアのデータアクセス頻度に応じて伝播させる更新情報を変更することにより，更新伝播時のトラヒックを軽減する．さらに，提案手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し，その有効性について検証する．

第4章では，各ピアが，ある一定量の大きさの更新が発生した場合に更新データを要求する環境を想定し，データの更新量（データの更新の大きさ，度合）に応じて更新データを伝播させるタイミングを変更するUPDD-S法およびUPDD-SO法を提案する．この手法では，更新が発生した場合に，その時点で更新データを必要としていないピアへの更新データの伝播を抑制することにより，更新伝播時のトラヒックを削減する．さらに，提案手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し，その有効性について検証する．

第5章では，本論文の成果を要約したのち，今後の研究課題について述べ，本論文のまとめとする．

なお，第2章は，文献 [63, 64, 66] で公表した結果に基づき論述する．第3章は，文献 [65, 67, 69] で公表した結果に基づき論述し，第4章は，文献 [68, 70, 71, 72, 73, 74] で公表した結果に基づき論述する．

第2章

木構造に基づく複製更新伝播法

2.1 まえがき

P2P ネットワークにおける更新伝播手法は，これまでにいくつか提案されている．しかし，これまで提案されている手法は，複製を所持しているピアに確実に更新データを伝播させる保証がなかったり，更新伝播時の各ピアの負荷や遅延に関してほとんど考慮されていない．また，ピアの参加や脱退，ネットワーク障害にともなうネットワークトポロジの動的な変化への対応が不十分であった．

本章では，複製を所持しているピアに確実に更新データを伝播させつつ，更新伝播時の負荷分散と遅延減少の両立を目的として，木構造を用いた更新伝播法（UPT: Update Propagation Tree 法）を提案する．UPT 法では，データのオリジナルを所持するピア（以降，オリジナルノードと表記する）を根ノードとし，そのデータの複製を所持するピアを内部節点とする n 分木の論理ネットワーク（更新伝播木）を構築し，この更新伝播木に沿って更新データを伝播させる．この手法により，更新伝播時の負荷分散と遅延減少を両立できる．さらに，更新伝播木上の各ピアは自身の周辺ノードの情報を管理しておき，複製の作成によるピアの参加時や，複製の削除やネットワーク障害などによるピアの脱退時にも，局所的な情報のみで自律分散的に更新伝播木を修復する．

以下では，2.2 節で想定環境を説明し，2.3 節で関連研究について紹介する．2.4 節では，本章で提案する木構造に基づく複製更新伝播法（UPT 法）について説明する．2.5 節でシミュレーション実験の結果を示し，最後に 2.6 節で本章のまとめを行う．

2.2 想定環境

本章における提案手法は，構造型P2Pネットワーク，非構造型P2Pネットワークのどちらに対しても適応可能であるが，1.3節で説明したように，複製を配置することおよび効率的に更新データを伝播させることは，トポロジに明確な指針が存在しない非構造型P2Pネットワークにおいてより有効である．したがって，本章では，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所持するデータにアクセスする環境を想定する．各ピアは，自身のデータ記憶領域に他のピアが所持しているデータの複製を作成する．ここで，本論文における提案手法をはじめ，非構造型P2Pネットワークを対象としたデータ共有サービスでは，データを要求したピアからその要求に応答したピアまでのパス（検索パス）上に存在するピアが複製を作成することが多い．そこで，本論文では，以下の検索手順に基づいて，検索パス上に存在するピアを決定するものとする．ピアがデータを要求する場合，ネットワーク内に検索クエリをフラッディングすることにより，データを所持するピアが存在するかを問い合わせる．要求データを所持するピアを複数発見した場合，それらのうち論理ネットワーク上のホップ数が最も小さいピアにアクセスするものとする．

また，各データのオリジナルを所持するオリジナルノードのみがそのデータを更新するものとし，第2章および第3章では，データのオリジナルが更新された時点で，更新前のデータの複製は無効になるものとする．一方，第4章では，最新のデータの複製を所持していない場合でも，ユーザ自身が許容している誤差の範囲内であれば，その複製はそのユーザにとっては有益なものであるとする．

2.3 関連研究

1.3節で述べたとおり，P2Pネットワークサービスに関する研究分野では，データの複製配置に関する研究は盛んに行われているが，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場合が想定され，適切に更新データの伝播がされないと，更新前の古いデータにアクセスする可能性が高くなる．そのため，複製を所持するピアに対して，適切に更新データを伝播させる機構が必要となる．本節では，従来研究における更新伝播法について説明する．

2.3.1 複製（インデックス）の配置を考慮した更新伝播

文献 [55] で提案されている CUP (Controlled Update Propagation) 法では，検索パス上にある全てのピアに複製（インデックス）を配置し，更新が発生した場合には，検索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 (Dynamic-tree based Update Propagation) 法 [80] では，検索パス上の全てのピアにはインデックスを配置せず，検索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，更新データを経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，更新データを伝播すべきピアの管理を容易にしている．しかし，更新データを伝播させる経路を柔軟に変更できないため，検索パスが長い場合には更新伝播にかかる遅延が大きくなってしまう．また，一部の複製所持ピアにデータ要求が集中した場合，そのピアが更新データを伝播させる回数が多くなり，負荷が高まってしまう．さらに，優先的に更新データを伝播させるピアを設定したり，更新データを伝播させるピアと伝播させないピアとに分けることができないため，更新伝播時のトラヒックの削減も困難である．

2.3.2 構造型 P2P ネットワークにおける更新伝播

構造型 P2P ネットワークにおいても，複製を所持するピアに対して更新データを伝播させる研究が行われている．

Freenet [14] では，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キーが割り当てられているピアへとクエリを伝播させることにより，要求データを所持しているピアを効率的に発見する．その後，発見したピアに対して更新データの伝播を行う．しかし，あるデータの複製を所持するピアを全て発見できるという保証はなく，複製を所持するピアが更新データを受け取れない場合が発生する．

また，文献 [10] で提案されている手法では，アドレス空間を分割し，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，この木に沿って更新データを伝播させることで，要求データを所持しているピアを効率的に発見する．しかし，アドレス空間上での各ピアやデータの位置を参考に更新データを伝播させているため，ネットワーク構造や各ピアが所持するデータを柔軟に変更できない．

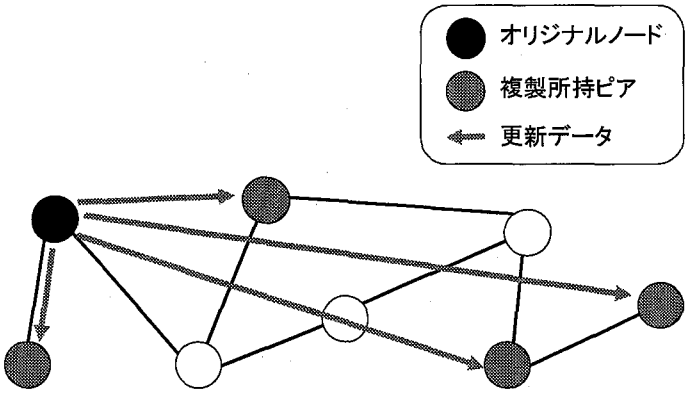


図 2.1: 放射伝播法

2.3.3 非構造型 P2P ネットワークにおける更新伝播

非構造型 P2P ネットワークにおける更新伝播手法は，以下に示す四種に大別できる．

単純な更新伝播法

複製を所持するピアに更新データを伝播させるための単純な手法として，オリジナルノードが，そのデータの複製を所持する全てのピアのネットワーク上での識別子（例：IP アドレス）を管理しておき，更新が発生するたびに複製を所持する全てのピアに直接更新データを伝播させる手法が考えられる．以降，この手法を放射伝播法と表記する．放射伝播法による更新データの伝播の様子を図 2.1 に示す．この手法を用いた場合，複製を所持する全てのピアに即座に更新データを送信できるが，更新伝播時の負荷がオリジナルノードに集中する．さらに，複製を所持するピア数の増加にともないオリジナルノードの負荷が線形的に増加するため，大規模なネットワークには適していない．

この問題を解決する別の更新伝播法として，複製を所持する各ピアが，同じ複製を所持する他の一つのピアへ，直線的に更新データを伝播させる手法が考えられる．以降，この手法を直線伝播法と表記する．直線伝播法による更新データの伝播の様子を図 2.2 に示す．この手法により，更新伝播時におけるオリジナルノードの負荷を他のピアに分散できるが，複製を所持するピア数の増加にともない更新データを伝播させる経路が線形的に長くなる．したがって，オリジナルノードから複製を所持する全てのピアへ更新データの伝播が完了するまでに，大きな遅延が生じてしまう．

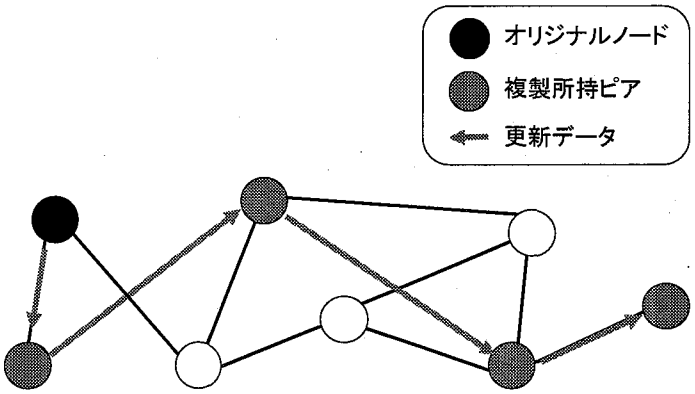


図 2.2: 直線伝播法

確率的な更新伝播法

Datta らは，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 [17]．この手法では，複製を所持する各ピアは，同じデータの複製を所持するピアの情報をランダムに保持し，それらのピアを自身の隣接ピアとする．データを更新したピアは，自身の隣接ピアに対してある確率（更新伝播率）で更新データを送信する．一度更新データを受信したピアは，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，部分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送信する．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信する確率が低くなる．しかしこの手法では，全てのピアに確実に更新データが伝播される保証がない．さらに，一つのピアに対して複数の経路を通って更新データが伝播されることがあり，余分なトラヒックが発生する可能性がある．

チェイン伝播法

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構築することにより，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する手法（チェイン伝播法）を提案している [62]．この手法では，図 2.3 のように，

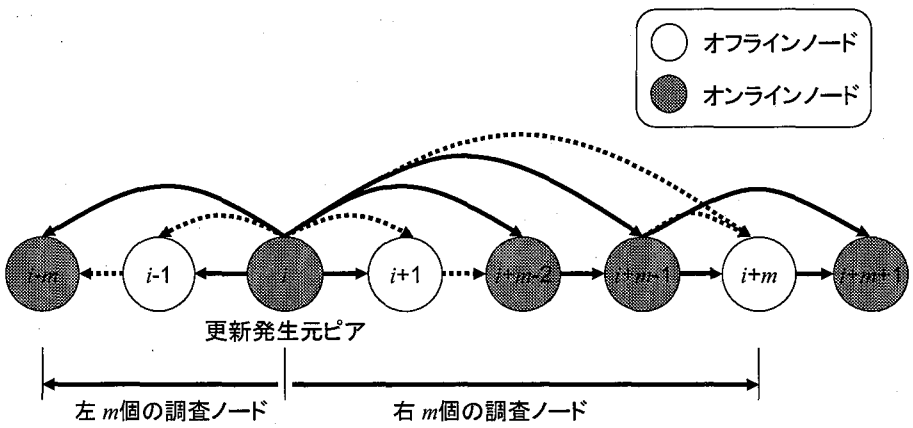


図 2.3: チェイン伝播法

複製を所持する各ピアは，左右 m 個ずつのピアの情報（例：IP アドレス）を自身の調査ノードとして保持する．チェイン上のピアで更新が発生した場合，データを更新したピアは，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する．その後，更新伝播責任ノードは，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送信し，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，チェイン上の全てのオンラインノードに更新データを伝播させることができる．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，ピア数が多くなると，全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う．また，更新データを多くのピアに送信するピアと全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う．

複製所持ピアのリストを利用した更新伝播法

文献 [79] では，非構造型 P2P ネットワークにおける問合せの効率化やトラヒックの削減を目的として，効率的な複製配置手法について議論しており，その中で更新データの伝播手法についても議論している．この手法では，オリジナルノードは，そのデータの複製を所持する全てのピアからなるリストを管理する．データに更新が発生した場合，オリジナルノードはそのリストを分割し，分割された各リストの中から一つのピアを選択して，更新データおよび分割したリストを送信する．更新データおよびリストを受け取ったピアは，

そのリスト内のピアへ，更新データとさらに分割したピアのリストを送信する．この操作を繰り返すことにより，更新データを伝播させる．

しかし，この手法では，オリジナルノードは，複製を所持する全てのピアの情報を管理しなければならない．したがって，複製の作成や削除が頻繁に起こるような環境では，リストの管理や更新のための負荷が大きくなってしまう．また，各ピアは，他ピアのネットワークへの参加状態を常に把握しているわけではないので，更新伝播時にピアの存在を確認したり，更新データを送信するピアを再決定するために遅延が発生する．

2.3.4 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盛んに行われている [6,25,26,60,61]．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，各ピアに付属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，よりバランスの取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，本論文の提案手法では，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取れた木を構築，維持している．

2.4 UPT法

本節では，本章で提案する複製更新伝播法であるUPT法について説明する．UPT法では，検索に用いるP2Pネットワークとは別に，データのオリジナルを所持するオリジナルノードを根ノードとし，そのデータの複製を所持するピアを内部節点とする n 分木の論理ネットワーク（更新伝播木）を構築する．検索によってピアがあるデータへアクセスした場合，検索を行ったピアは，1.3節で説明したような何らかの複製配置方式によって複製を作成し，作成したデータの更新伝播木へ参加する．一方，ピアが所持できるデータ量には限界があるため，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．複製を削除したピアは，削除したデータの更新伝播木から脱退する．また，

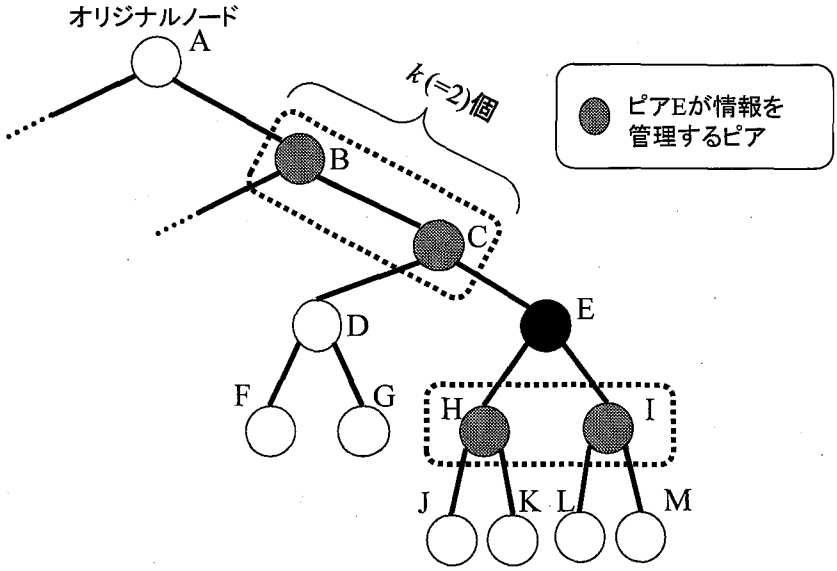


図 2.4: 更新伝播木上のピアが管理する情報 ($k = 2$)

データが更新された場合には，この更新伝播木に沿って更新データを伝播させる．このように，UPT法では，複製を所持しているピアは，必ずそのデータの更新伝播木に参加しているため，確実に更新データを受信することができる．また，各ピアは，更新伝播時に，自身の子ノードである最大でもn個のピアに更新データを送信すればよく，各ピアの更新伝播時の負荷が小さく抑えられる．さらに，全ての複製所持ピアに更新データを伝播させる際の遅延も対数オーダに抑えられる．以降，本章では，ある一つのデータに対する更新伝播木を想定し，議論を進める．

UPT法では，各ピアが，更新伝播木における親の方向に対してk個分の先祖ノード，および，子ノードの情報（例：IPアドレス）を管理しておき，更新伝播木上での参加位置の決定，および複製削除時などの更新伝播木の再構築を自律分散的に行う．UPT法において，$k = 2$の場合の更新伝播木を図2.4に示す．この図においてピアEは，自身の2個分の先祖ノードとしてピアBとピアCの情報を，自身の子ノードとしてピアHとピアIの情報を管理している．UPT法では，更新伝播木上の各ピアの子ノードの最大数nの値を変更することにより，更新伝播時の各ピアの負荷や遅延が変化することが考えられる．また，更新伝播木上の各ピアが管理する先祖ノードの数kの値を変更することにより，更新伝播木を修復する際のメッセージ数や更新伝播時の遅延が変化することが考えられる．

以下では，複製の作成によるピアの更新伝播木への参加方法，複製の削除によるピアの更新伝播木からの脱退，および異常脱退発生時の更新伝播木の修復方法を説明する．

2.4.1 ピアの参加

新たに複製を作成したピアは，そのデータの更新伝播木に参加する必要がある．ピアが参加する際には，各ピアが管理している少ない情報のみを用いて，更新伝播木の偏りが大きくならないようにピアを参加させる必要がある．UPT法では，各ピアが更新伝播木上のk個上位の先祖ノードの情報を管理しているため，その先祖ノードの子ノードに空きがあるかを確認することにより，新たなピアを更新伝播木のより上部に参加させる．これにより，更新伝播木の偏りを小さく抑えることができる．ここで，新たに更新伝播木に参加するピアを新規参加ピア，新規参加ピアの参加位置の決定を行うピアを責任ピアと呼ぶ．ここでは，検索時にデータを要求したピアからのクエリに応答したピアが責任ピアとなるものとする．更新伝播木への参加手順は以下の通りである．

1. 責任ピアは，自身が更新伝播木の根ノード（オリジナルノード）以外であれば，更新伝播木上の自身からk個上位の先祖ノード（先祖ノードがk個未満の場合は根ノード）に子ノードの数xを問い合わせる．このとき，$x < n$の場合は，新規参加ピアをその先祖ノードの子ノードとし，その先祖ノードは自身の子ノードに関する情報として，新規参加ピアを追加する．さらに，新規参加ピアは接続したノードから$k-1$個分（先祖ノードの数が$k-1$個未満の場合は根ノードまで）の先祖ノードの情報を受け取り，その接続したノードおよび$k-1$個の先祖ノードの情報を，自身の先祖ノードの情報として記録する．$x = n$の場合は，同様の処理を$k-i$ $(i = 1,2,\cdots,k-1)$個上位の先祖ノードに対して，新規参加ピアの参加位置が決まるまで（iを一つずつ増やしながら）繰り返し行う．上記の処理により，責任ピアが新規参加ピアの参加位置を決定できなかった場合，手順2.に進む．

2. 責任ピアは，自身の子ノードの数yを確認し，$y < n$のときは，新規参加ピアを自身の子ノードとし，$k-1$個分（先祖ノードの数が$k-1$個未満の場合は根ノードまで）の先祖ノードの情報を新規参加ピアに送る．また，責任ピアは，自身の子ノードに関する情報に，新規参加ピアを追加し，参加処理を終了する．$y = n$のときは，手順3.に進む．

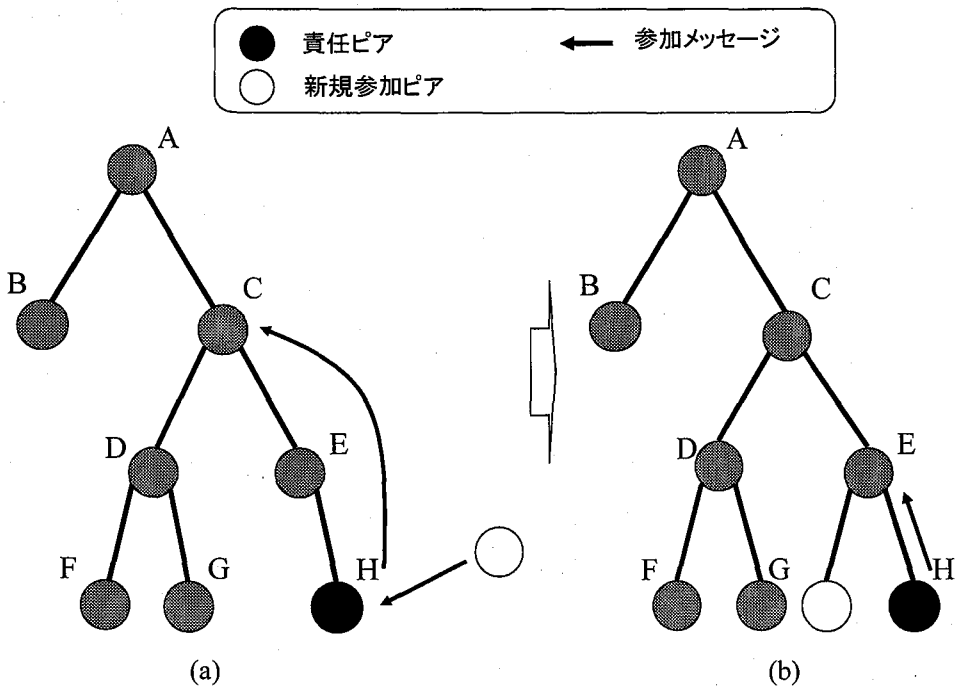


図 2.5: ピアの参加例

3. 責任ピアは，自身の子ノードの中からランダムに一つの子ノードを選択し，その子ノードを新たに新規参加ピアの責任ピアとする．新たに責任ピアとなったピアは，手順 2. 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．

$n = 2$（2 分木），$k = 2$ の場合における，新規参加ピアの参加手順の動作例を，図 2.5 に示す．この図では，新規参加ピアからのクエリに応答したピア H が責任ピアとなっている．まず，ピア H は，自身がオリジナルノードでないので，図 2.5(a) のように，自身から $2(= k)$ 個上位の先祖ノードであるピア C に子ノードの数 x を問い合わせる．ここで，$x = 2(= n)$ であるため，ピア C には新規参加ピアを接続しない．続いて，ピア H は，図 2.5(b) のように，自身から $1(= k - 1)$ 個上位の先祖ノードであるピア E に子ノードの数 x を問い合わせる．ここで，$x = 1(< n)$ であるため，新規参加ピアをピア E の子ノードとし，参加処理を終了する．

2.4.2 ピアの脱退

一般的に，ピアのデータ記憶領域には限界があり，新たにデータの複製を作成する際に，他のデータの複製を削除しなければならない場合がある．UPT法では，データの複製を削除したピアは，そのデータの更新伝播木から脱退する．ここで，更新伝播木から脱退するピアを脱退ピアと呼ぶ．脱退ピアが更新伝播木から脱退した場合，更新伝播木が分断されてしまい，脱退ピアの子孫ノードは，以後の更新データを受け取ることができなくなる可能性がある．このような分断を避けるため，UPT法では，ピアの脱退に対し，更新伝播木の修復を行う．ピアが脱退する際には，更新伝播木の修復時のメッセージ数を抑えつつ，可能な限り更新伝播木の形状を維持する必要がある．UPT法では，脱退ピアが脱退メッセージを葉ノードへと伝播させ，その葉ノードと脱退ピアの位置を入れ替えることにより，更新伝播木の形状を大幅に変更することなく，少ないメッセージ数で更新伝播木の分断を修復できる．更新伝播木の修復手順は以下の通りである．

1. 脱退ピアが葉ノードの場合，脱退ピアは自身の親ノードに更新伝播木から脱退することを通知する．通知を受け取った親ノードは，自身のもつ子ノードに関する情報から脱退ピアの情報を削除する．脱退ピアは，親ノードを含むk個分（先祖ノードの数が$k-1$個の場合は根ノードまで）の先祖ノードの情報を削除し，脱退処理を終了する．

2. 脱退ピアが子ノードをもつ場合，脱退ピアは自身の子ノードの中から一つをランダムに選択し，脱退メッセージを送信する．脱退メッセージを受け取った子ノードは，自身が葉ノードでない限り，同様の手順で子ノードを選択し脱退メッセージを伝播させる．ここで，脱退メッセージを受け取った葉ノードを入替えピアと呼ぶ．入替えピアは，自身の位置を脱退ピアの位置と入れ替える．入替えピアの親ノードは，その入替えピアの情報を，自身のもつ子ノードに関する情報から削除し，手順3.に進む．

3. 脱退ピアと位置を入れ替えた入替えピアは，新たな位置における親ノードと，深さk分の子孫ノード（子孫ノードの数がk個未満の場合は葉ノード）にまで，ピアが入れ替わったことを通知する．通知を受け取った親ノードと子孫ノードは，自身のもつ子ノードおよび先祖ノードに関する情報において，脱退ピアを，入替えピアの情報に変更する．その後，脱退ピアは，先祖ノードや子ノードに関する情報を全て削除し，脱退処理を終了する．

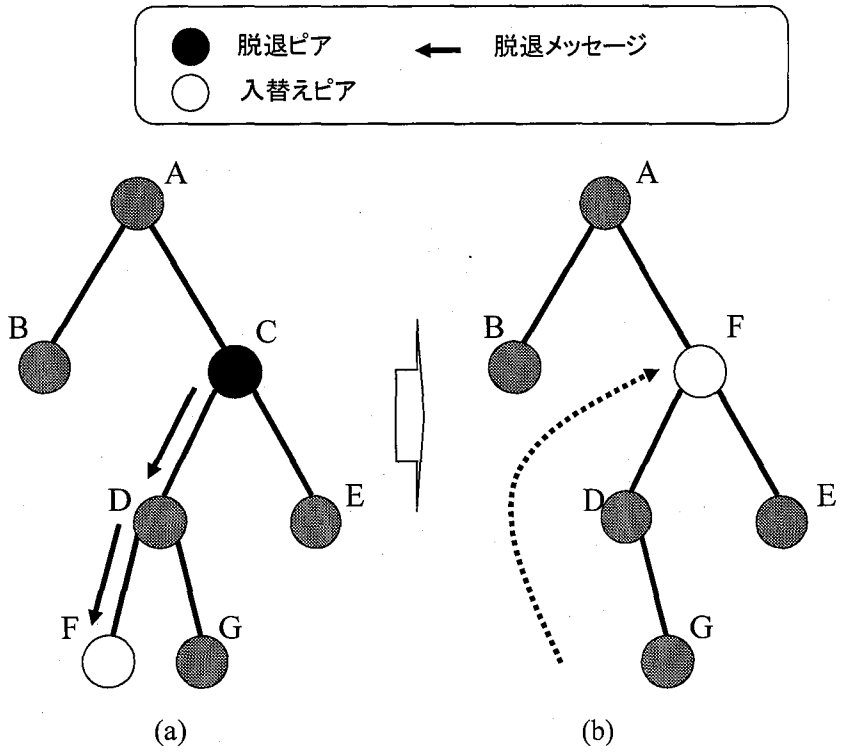


図 2.6: ピアの脱退例

$n = 2$（2 分木），$k = 2$ の場合における，ピアの脱退手順の動作例を，図 2.6 に示す．この図では，ピア C が脱退するものとする．まず，ピア C は子ノードをもつため，図 2.6(a) のように，ランダムに選択したピア D に脱退メッセージを送信する．脱退メッセージを受け取ったピア D も子ノードをもつため，ランダムに選択したピア F に脱退メッセージを送信する．ピア F は葉ノードであるため，自身が入替えピアとなる．その後，図 2.6(b) のように，ピア F はピア C と位置を入れ替え，脱退処理を終了する．

2.4.3 ピアの異常脱退

2.4.2 項で説明した脱退手順では，ネットワーク障害や機器の故障，アプリケーションの強制終了などによる，周辺ピアへの通知のない脱退（以下，このような脱退を「異常脱退」と称する）を考慮していない．一方，実環境における P2P ネットワークでは，ピアの異常脱退が発生する場合も多い．ピアの異常脱退が発生した場合，更新伝播木の修復を行うことができず，更新データを全てのピアに伝播させることができなくなる可能性がある．更

新伝播木が分断された場合に，分断されたピアの一つが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する方法も考えられるが，大きなトラヒックが発生する．そこでUPT法では，更新伝播木上の各ノードが管理している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局所的に更新伝播木を修復する．そのため，フラッディングに比べ，修復に要するトラヒックを大幅に抑制できる．UPT法では，最大$k-1$個までの連続した先祖ノードの異常脱退まで修復が可能となる．

UPT法では，ピアの異常脱退に対応するため，更新伝播木上の各ピアが親ノードに対して，ネットワーク上に存在しているかを確認するためのメッセージ（確認情報）を定期的に送信する．これにより，親ノードの異常脱退を検出できる．ここで，親ノードの異常脱退を検出したピアを修復責任ピアと呼ぶ．更新伝播木の修復の際には，修復時のメッセージ数を抑えつつ，可能な限り更新伝播木の形状を維持する必要がある．UPT法では，異常脱退を検出したピアが，脱退メッセージを葉ノードへと伝播させ，その葉ノードを利用して脱退箇所を修復する．このように，各ピアが管理している情報を利用し，異常脱退を検出したピア自身およびその子孫ノード以下で局所的に更新伝播木を修復できるため，更新伝播木の形状を大幅に変更することなく，修復時のメッセージ数も抑えられる．ピアの異常脱退発生時の更新伝播木の修復手順の詳細は，以下の通りである．

1. 更新伝播木上の各ピアは，親ノードに対して定期的に確認情報を送り，親ノードの生存を確かめる．確認情報に対する親ノードからの応答が得られなかった場合，親ノードの異常脱退を検出したものとし，そのピアが修復責任ピアとなる．修復責任ピアは，さらに一つ上位の先祖ノードに確認情報を送る操作を，生存する先祖ノードが見つかるまで繰り返すことにより，連続する先祖ノードの脱退数z，および，異常脱退した先祖ノード群の位置（脱退認識箇所）を調べる．ここで，初めて生存が確認できた先祖ノードを修復管理ピアと呼ぶ．また，連続する先祖ノードの脱退数zを未修復ノード数と呼ぶ．

2. 修復管理ピアから，脱退認識箇所の中で最も根に近い箇所が既に他のピアによって修復されていることが修復責任ピアに伝えられた場合，修復責任ピアは自身のz個上位の先祖ノードの情報を修復済みのピア情報に変更する．その後，未修復ノード数zの値を1減らし，脱退認識箇所からその修復済みの箇所を削除した後，その修復済みのピアを新たな修復管理ピアとする．脱退認識箇所の最上部が既に他のピアによっ

て修復されているという情報が修復責任ピアに伝えられる限りこの操作を繰り返し，$z=0$ となった場合，手順5. に進む．脱退認識箇所の最上部が未修復であれば，手順3. に進む．

3. 修復責任ピアが葉ノードであれば，修復責任ピア自身が脱退認識箇所の最上部に移動する．具体的には，修復責任ピアが修復管理ピアの子ノードとして参加し，修復管理ピアから $k-1$ 個分（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を受け取る．また，修復管理ピアは，自身の子ノードに関する情報を，修復責任ピアの情報に書き換え，修復処理を終了する．

4. 修復責任ピアが葉ノードでない場合，修復責任ピアは，自身の子ノードの中から一つをランダムに選択し，先祖ノードの脱退メッセージを送信する．この操作を脱退メッセージが葉ノードに達するまで繰り返す．脱退メッセージを受け取った葉ノードは，自身が修復管理ピアの子ノードとして参加し，修復管理ピアから $k-1$ 個分（先祖ノードの数が $k-1$ 個未満の場合は根ノードまで）の先祖ノードの情報を受け取る．また通知を受け取った修復管理ピアは，自身の子ノードに関する情報を，入替えピアの情報に書き換える．入替えピアの親ノードは，その入替えピアを自身の子ノードに関する情報から削除する．その後，未修復ノード数 z の値を1減らし，入替えピアを新たな修復管理ピアとする．z が0でない限り，手順2.〜4. の操作を繰り返し，$z=0$ となれば，手順5. に進む．

5. 修復責任ピアは，自身が所持している先祖ノードの情報，および自身の情報を，k 個下位の子孫ノード（子孫ノードの数が k 個未満の場合は葉ノード）にまで伝播させる．その情報を受け取ったピアは，自身の先祖ノードの情報を書き換え，修復処理を終了する．

以上の動作により，修復責任ピア，およびその子孫ノードを更新伝播木に再参加させ，更新伝播木の修復が完了する．

$n=2$（2分木），$k=3$ の場合における，ピアの異常脱退発生時の更新伝播木の修復例を，図2.7に示す．図において，ピアB, Dが異常脱退した場合，ピアC, G, Hが親ノードに対して確認情報を送信することにより，脱退を検出する．ここでは，図2.7(a) のように，ピアGが最初に脱退を検出したものとする．ピアGは，自身の親ノードであるピアDからの応答が得られなかった場合，一つ上位の先祖ノードであるピアBに確認情報を送る．こ

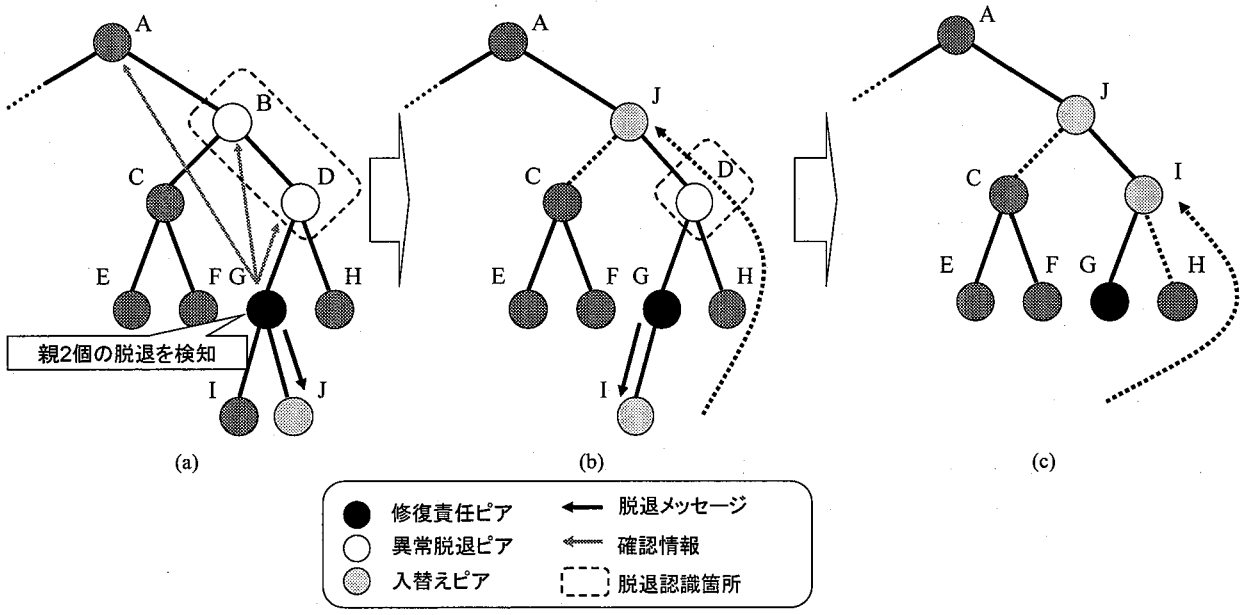


図 2.7: 異常脱退発生時の木の修復法

こでは，ピアBからも応答が得られないため，さらに上位の先祖ノードであるピアAに確認情報を送る．ここでピアAからの応答が得られると，未修復ノード数が2($z = 2$)，脱退認識箇所がピアBおよびピアDの位置であるとわかる．以上の動作により，ピアGが修復責任ピア，ピアAが修復管理ピアとなる．また，この時点で脱退認識箇所の最上部であるピアBの位置は他のピアによって修復されていない．まず，修復責任ピアであるピアGは，自身が葉ノードではないため，自身の子ノードの中からランダムに選択したピアJに脱退メッセージを送信する．脱退メッセージを受け取ったピアJは葉ノードであるため，自身が入替えピアとなり，脱退認識箇所の最上部であるピアBの位置とピアJの位置を入れ替え，zの値を1減らす（図2.7(b)）．ここで，$z = 1(> 0)$なので修復を続け，ピアGはピアIに脱退メッセージを送信する．ピアIは葉ノードなので，ピアIが入替えピアとなり，ピアDの位置とピアIの位置を入れ替え，zの値を1減らす（図2.7(c)）．この時点で$z = 0$となり，ピアGは，更新された先祖ノード（ピアIおよびピアJ）の情報を伝えるべき子ノードをもたないので，ピアGが担当する修復を完了する．

ここで，ピアCおよびピアHは，更新伝播木が修復されたという情報を得ていない．そのため，これらのピアは，ピアBおよびピアDの脱退に対する修復処理を開始する．例えば，ピアHは，元々の親ノードであるピアDおよびその先祖ノードであるピアBの脱退を

検出し，ピアAを修復管理ピアとする．ここでピアHは，ピアAから，すでにピアBの位置がピアJによって修復されていることが伝えられる．その情報を受け取ったピアHは，自身が管理するピアBの情報をピアJの情報に書き換え，zの値を1減らし，新たにピアJを修復管理ピアとする．次にピアHは，ピアJから，ピアDの位置がすでにピアIによって修復されていることを伝えられる．ピアHは，自身が管理するピアDの情報をピアIの情報に書き換え，zの値を1減らす．ここで，$z = 0$となり，ピアHは，更新された先祖ノード（ピアIおよびピアJ）の情報を伝えるべき子ノードをもたないので，ピアHが担当する修復を完了する．同様に，ピアCもピアBの位置がピアJによって修復されたことを認識し，自身が担当する修復を完了する．以上の動作により，ピアBおよびDの異常脱退に対する更新伝播木の修復が完了する．

ここで，一つの修復責任ピアが修復操作を行っている間に，他のピアからも脱退メッセージなどの修復に必要なメッセージが送られてきた場合，修復管理ピアは，そのメッセージの送信元ピアに，すでに他のピアによる修復が行われていることを通知する．この通知を受け取ったピアは，修復処理を停止し，現在の修復が完了するまで待機する．修復管理ピアは，修復責任ピアにより修復処理が完了した後，待機させていたピアのうちの一つに対し，修復処理の再開を指示する．また，脱退メッセージを葉ノードへと伝播させている途中で，自身は葉ノードではないにも関わらず，子ノードが全て異常脱退してしまっているために，脱退メッセージを葉ノードまで伝播させることができない場合がある．その場合，脱退メッセージの伝播および更新伝播木の修復を中断し，自身の子孫ノードの修復を待ち，修復が完了した後で，再び脱退メッセージの伝播を再開させる．ただし，一定時間修復を待機していても子孫ノードが修復されない場合は，子孫ノードが全て異常脱退したものとして，修復処理を再開する．

2.5 性能評価

本節では，UPT法が，子ノードの最大数nや各ピアが管理する先祖ノードの数kの値が変化した場合でも，更新伝播時の遅延や各ピアの負荷，木構造維持に必要なメッセージ数の面で有効であるか，およびこれらの値はどのように変化するかを調査する．以下では，UPT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．

2.5.1 シミュレーション環境

シミュレーション実験では，非構造型P2Pネットワークにおけるデータ共有を想定した．インターネットにおける各ノードの隣接ノード数は，べき法則 (Power-Law) の性質に従うことが示されている [5, 19]．また，インターネット上に構築されたP2Pネットワークも，同様にべき法則の性質があることが報告されている [1, 52, 53]．そこで，P2Pネットワークに参加するピアの数は5,000とし，それらがべき法則に従うネットワークを構成するものとした．ただし，2.5.2項においてピア数に対する遅延や負荷の影響を評価する場合にのみ，ピア数を1,000から5,000までの間で変化させた．ここで，i番目のピアの隣接ピアの数をd_iとし，d_iを以下の式で与えた．

$$d_i = \lfloor 70 \cdot i^{-0.4} \rfloor \tag{2.1}$$

上式において，ピア数を5,000とした場合のd_iの分布を図2.8に示す．このように，一部のピアにリンクが集中する環境を実現した．

データの種類を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タイムスロット毎に0.1の確率であるデータを要求する．各ピアの各データに対するアクセス確率は，Zipf分布 [83] に従うものとする．Zipf分布とは，一部のデータにアクセスが集中するという一般的なデータ要求の分布を示したものであり，Webサービスなどのコンテンツに対するアクセス頻度もZipf分布に従うことが報告されている [4]．具体的には，j番目のデータの要求確率をq_jとし，以下の式で与えた．

$$q_j = \frac{j^{-\alpha}}{\sum_{k=1}^{100} k^{-\alpha}} \tag{2.2}$$

ここで，αはデータの要求頻度の差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，要求確率が大きいデータと小さいデータの差をある程度大きくするために，Zipf係数αは0.5とした．αを0.5としたときの各データの要求確率を図2.9に示す．クエリの伝播には，TTLを十分に大きな値としたフラッディングを用い，検索が必ず成功するものとした．各ピアは，クエリに応答したピアのうち，自身からの検索ネットワーク上のホップ数が最も小さいピアに対してアクセスするものとした．複製の配置手法には，パス複製法を用いた．パス複製法では，クエリを発行したピアからクエリに応答したピアまでの経路上にある全てのピアに複製を配置する．更新伝播時の負荷を可能な限り分散させるために，更新伝播木は2分木 (n=2) とした．

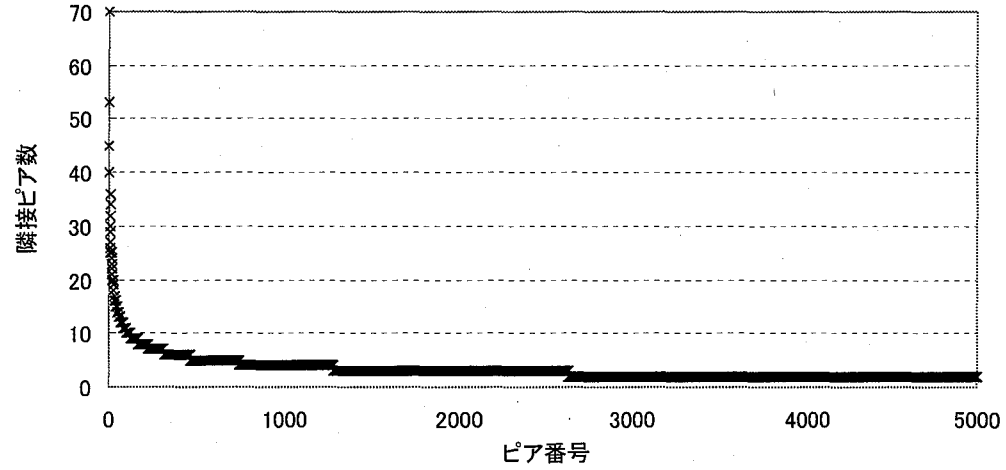


図 2.8: 隣接ピア数

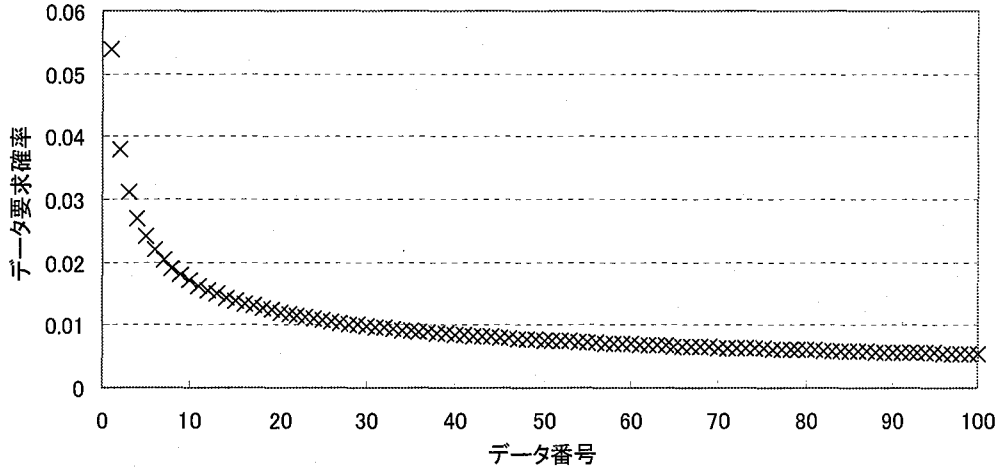


図 2.9: データ要求確率

各データのサイズは全て等しく，ネットワーク上の全てのデータの内の一部のみを各ピアが保有可能であるものとするため，複製を保有可能な数は全てのピアで10とした．ただし，データのオリジナルを所持するピアは，オリジナルの他に10個の複製を所持できるものとした．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

た複製の中で最も古い複製を削除し，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．

以上のような環境において，10,000タイムスロットのシミュレーション実験を行った．以下では，データ番号100のデータに注目して，このデータの更新伝播木に関する評価結果を説明する．ただし，他のデータを対象とした場合も，データ番号100のデータと同様の特徴をもつ結果が得られると考えられる．他のデータを対象とした場合の結果に関する考察は2.5.5項で行う．

2.5.2 他の複製更新伝播法との比較

まず，異常脱退が発生しない環境を想定し，2.3節で説明した直線伝播法，放射伝播法，文献[62]のチェイン伝播法($m = 3$)，および提案手法であるUPT法に関して，管理している先祖ノードの数が最も少ない場合($k = 1$)と十分に多い場合($k = 4$)について，以下の項目を評価した．

- 平均ホップ数

  一回の更新発生あたりに，オリジナルノードから各複製所持ピアへ更新データが伝播される際に要した論理ホップ数の平均．本論文では，更新伝播時の遅延をこのホップ数で評価する．

- 平均負荷

  一回の更新発生あたりに，更新データの送信を行った各ピアが更新データを送信した回数の平均値

ピア数に対する遅延の変化

平均ホップ数の結果を図2.10に示す．図2.10において，横軸はピア数を表し，縦軸は平均ホップ数を表す．放射伝播法を用いた場合，オリジナルノードが，複製を所持する全てのピアに更新データを送信するため，平均ホップ数は常に1となっている．直線伝播法を用いた場合，各ピアが一つのピアにしか更新データを送信しないため，ピア数の増加にしたがって平均ホップ数も線形的に増加している．チェイン伝播法の場合，ほとんどのピアがm個のピアに更新データを送信するため，直線伝播法と比較すると平均ホップ数は小さく抑えられる．しかし，ピア数の増加にしたがって，平均ホップ数が線形的に増加する．一

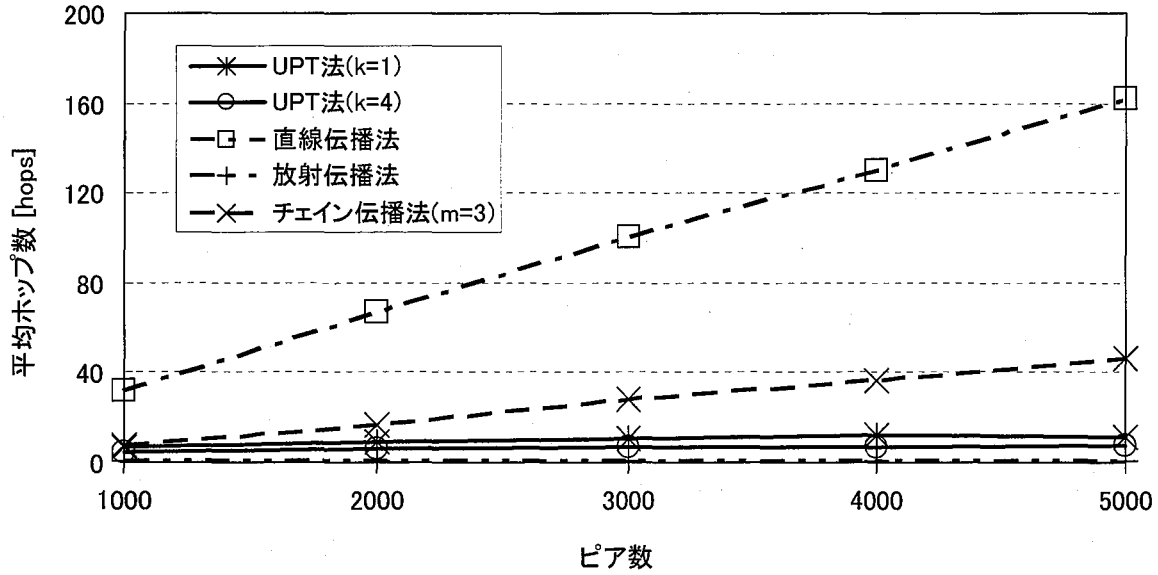


図 2.10: 平均ホップ数

方，UPT 法では，平均ホップ数が直線伝播法およびチェイン伝播法より常に小さく，ピア数の増加にともなう遅延の増加も対数オーダに抑えられている．したがって，ネットワーク上のピア数によらず UPT 法が有効であることがわかる．また，$k = 1$ よりも，$k = 4$ の場合の方が平均ホップ数が小さくなっている．これは，新規参加ピアが更新伝播木に参加する場合，k 個上位の先祖ノードの子ノードの数が n 未満のノードを発見するため，k の値が大きくなるほど，新規参加ピアが更新伝播木のより上部に参加でき，その結果，更新伝播木の偏りが小さくなり，より高さの低い完全 n 分木に近くなるためである．このことを検証するため，同様の環境において，UPT 法における k の値を変化させた場合の遅延への影響を調べた．k を変化させた場合の平均ホップ数，最大ホップ数をそれぞれ図 2.11，図 2.12 に示す．ここで，最大ホップ数とは，オリジナルノードから各複製所持ピアへ更新データが伝播される際に要した論理ホップ数の最大値である．これらの図では，平均ホップ数および最大ホップ数が最小となる理想的な形状である完全 2 分木における結果についても，検証のために示している．図 2.11 および図 2.12 より，k の値が大きくなるにつれてホップ数が小さくなっており，k の値が 1 の場合と比べて k の値が 5 の場合，平均ホップ数は約 30%，最大ホップ数は約 55% 減少している．以上の結果より，k の値が大きくなるほど更新伝播木の偏りが小さくなり，遅延が小さく抑えられることが確認できる．

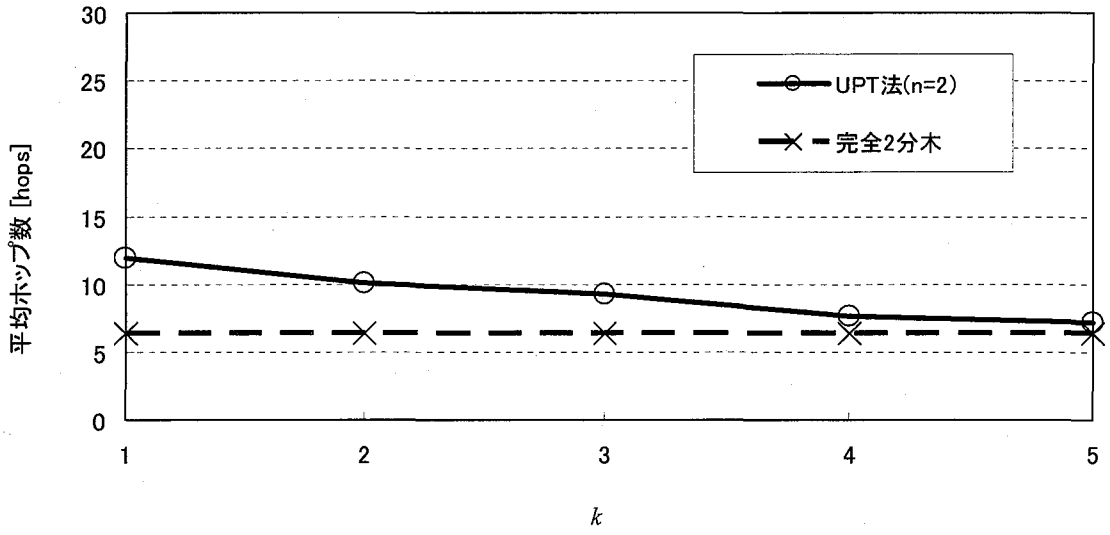


図 2.11: 管理する先祖ノードの数 k と平均ホップ数（UPT 法）

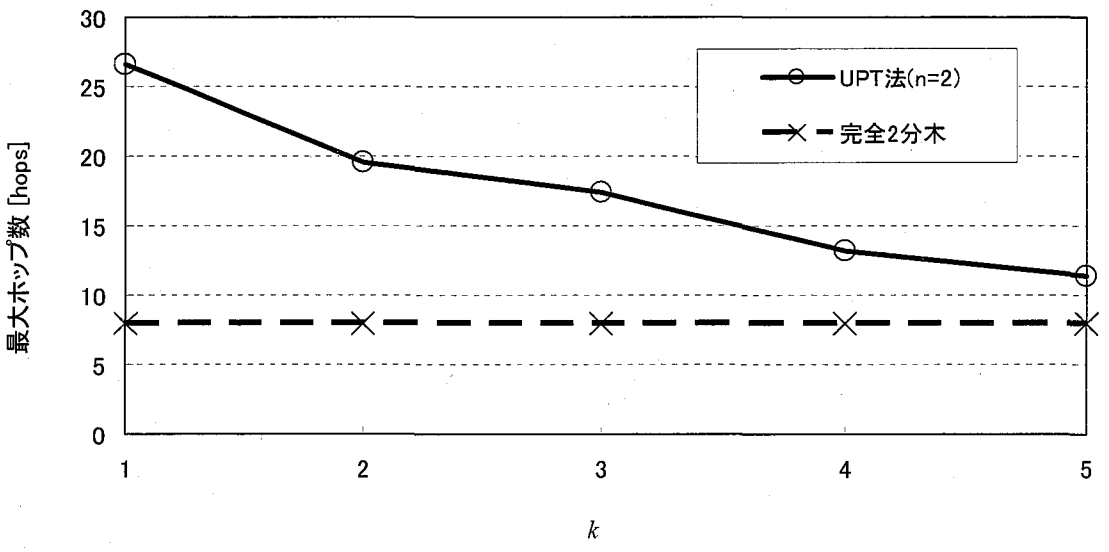


図 2.12: 管理する先祖ノードの数 k と最大ホップ数（UPT 法）

本シミュレーション実験では，論理ネットワーク上でのホップ数で更新伝播時の遅延を評価したが，物理ネットワーク上での遅延の結果も，論理ネットワーク上での結果と類似した特徴を示すと考えられる．しかし，論理ネットワーク上では１ホップであっても，物

理ネットワーク上ではいくつもの端末を経由しているなど，論理ネットワーク上でのホップ数が小さい場合に，必ずしも物理ネットワーク上での遅延が減少するとは限らない．したがって，物理ネットワークを考慮したさらなる検証が必要である．

ピア数に対する負荷の変化

平均負荷の結果を図2.13に示す．図2.13において，横軸はピア数を表し，縦軸は平均負荷を表す．ここで，放射伝播法では，オリジナルノードが，複製を持つ全てのピアに更新データを伝播させるため，ピア数の増加に比例して平均負荷が増加し，他の更新伝播法と比べて負荷が大幅に大きくなる．そのため，図2.13では放射伝播法による結果は省略している．直線伝播法を用いた場合，各ピアは複製を所持する一つのピアに更新データを送信するため，平均負荷は常に1となっている．チェイン伝播法の場合，オリジナルノードのみ左右両方向（最大6個）のピアに更新データを送信するが，それ以外のピアは，チェイン上の一方向のピアにのみ更新データを送信すればよく，平均負荷は最大でも3である．そのため，ピア数が増加しても，平均負荷は3程度に抑えられる．UPT法では，更新データを子ノードにのみ送信するため，一つのピアが更新データを送信すべきピア数は最大でも$n(=2)$である．そのため，ピア数が増加しても平均負荷は$n(=2)$以下になり，UPT法はネットワークに参加するピア数によらず有効であることがわかる．また，$k=1$と$k=4$ではほとんど差が現れず，kの値による平均負荷の差は小さいといえる．

以上の実験結果から，直線伝播法では平均ホップ数が，放射伝播法では平均負荷が極めて大きくなってしまう．チェイン伝播法では，これら二つの手法と比較して負荷分散と遅延減少を両立できているが，ピア数の増加にしたがって，平均ホップ数が線形的に大きくなってしまう．mの値を大きくすることにより，平均ホップ数の増加量を減らすことができるが，平均負荷が高まるため，負荷分散と遅延減少の両立は難しい．一方，UPT法では，平均負荷をn以下に保ちながら平均ホップ数を対数オーダに抑えることができ，負荷分散と遅延減少を両立できている．また，kの値を大きくすることにより更新伝播木を完全n分木に近づけることができ，遅延を小さく抑えられる．

2.5.3 異常脱退発生時の木構造維持コストの変化

次に，異常脱退が発生する環境において，UPT法におけるkの値を変化させた場合の，木構造維持に必要なメッセージ数を評価した．

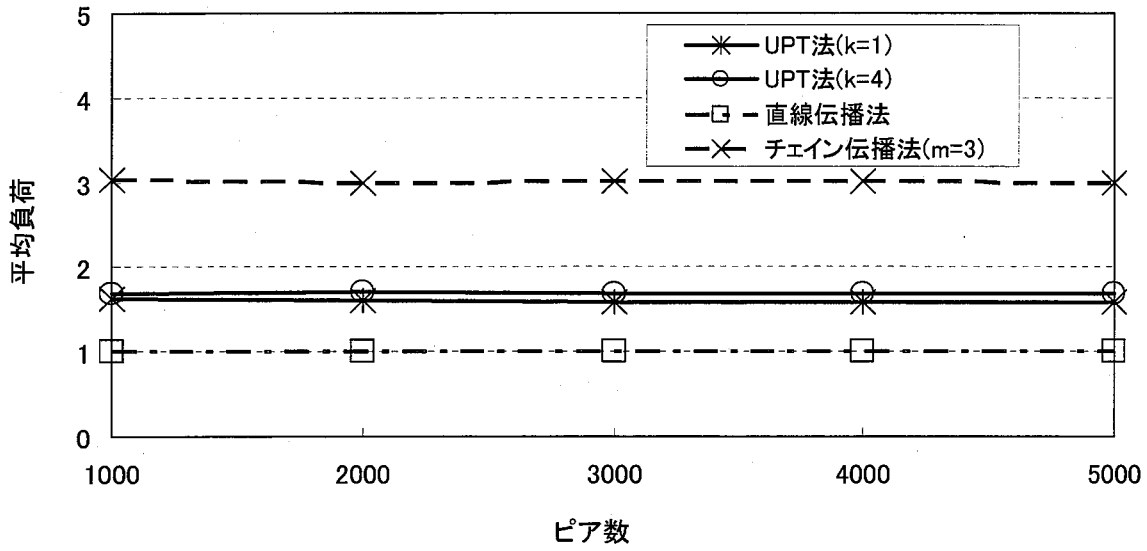


図 2.13: 平均負荷

ネットワーク上では，一定の確率（異常脱退発生確率）でピアの異常脱退が発生するものとした．更新伝播木に参加する各ピアは，確認情報を1タイムスロットごとに送信し，異常脱退を検出した場合は，即座に更新伝播木の修復を行うものとした．異常脱退によって更新伝播木の分断が修復できない（連続する k 個以上のノードが同時に脱退した）場合，脱退を検出したピアが検索ネットワークにクエリをフラッディングし，そのデータのオリジナルまたは複製を所持するピアを発見するものとした．このとき，発見したピアに対して参加要求を送り，分断した部分木を更新伝播木へ復帰させる．

そこで，異常脱退発生確率が，各ピアのデータ要求確率よりも小さい場合と大きい場合，および同程度の場合を網羅するために，異常脱退発生確率を1タイムスロットあたり0.001，0.01，0.1としたときの，以下の項目を評価した．

- 木構造修復メッセージ

  ピアの異常脱退が発生した際に，各ピアが管理するピアの情報を用いて局所的に更新伝播木を修復するために必要なメッセージ数

- 再参加メッセージ

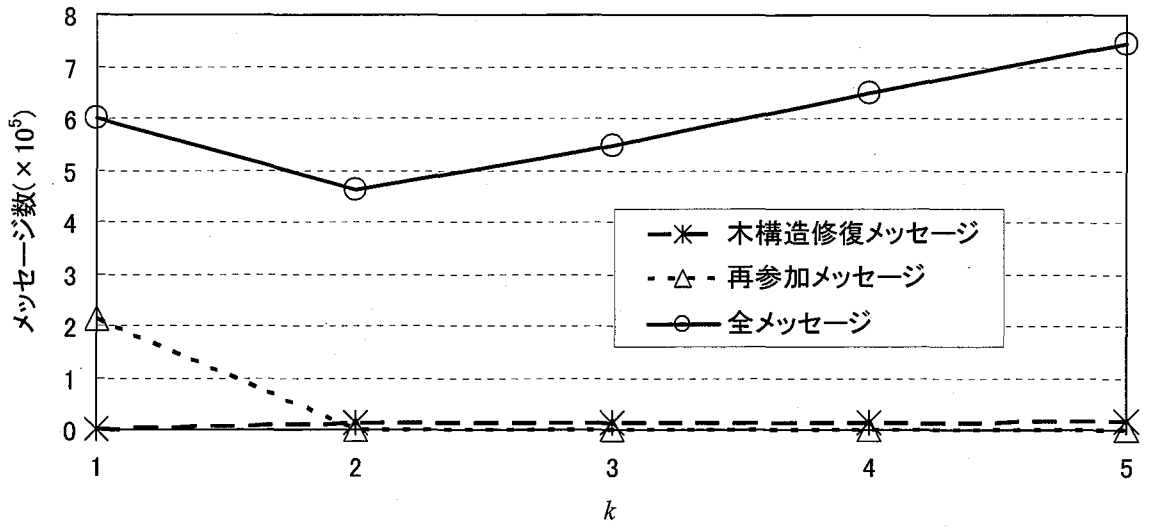


図 2.14: 管理する先祖ノードの数 k とメッセージ数（異常脱退確率：0.001）

各ピアが管理するピアの情報だけでは更新伝播木の修復が不可能な場合に，フラッディングによって更新伝播木に再参加するために必要なメッセージ数

- 全メッセージ

  木構造修復メッセージと再参加メッセージに加え，新規参加ピアの参加時や，複製削除等の正当な手続きを踏んだピアの脱退時にかかるメッセージ数を含めた，木構造維持に必要な全メッセージ数

実験結果を，それぞれ図2.14から図2.16に示す．これらの図において，横軸は k を表し，縦軸はメッセージ数を表す．結果より，異常脱退確率の大小にかかわらず，木構造修復メッセージ数は，k の値が増加するにつれて大きくなる．これは，更新伝播木の修復処理によって，ピアの更新伝播木上の位置が変更された場合に，より多くのピアの先祖ノードの情報を更新する必要があるためである．一方，再参加メッセージ数は，$k = 1$ の場合には，異常脱退による更新伝播木の分断が修復されないため，多くのピアがフラッディングによる更新伝播木への再参加の操作を行わなければならず，メッセージ数が極めて多くなっている．$k = 2$ 以上であれば，更新伝播木を修復できる可能性が高くなり，フラッディング操作を行う回数が少なくなるため，メッセージ数も小さく抑えられる．その中でも，ピアの異常

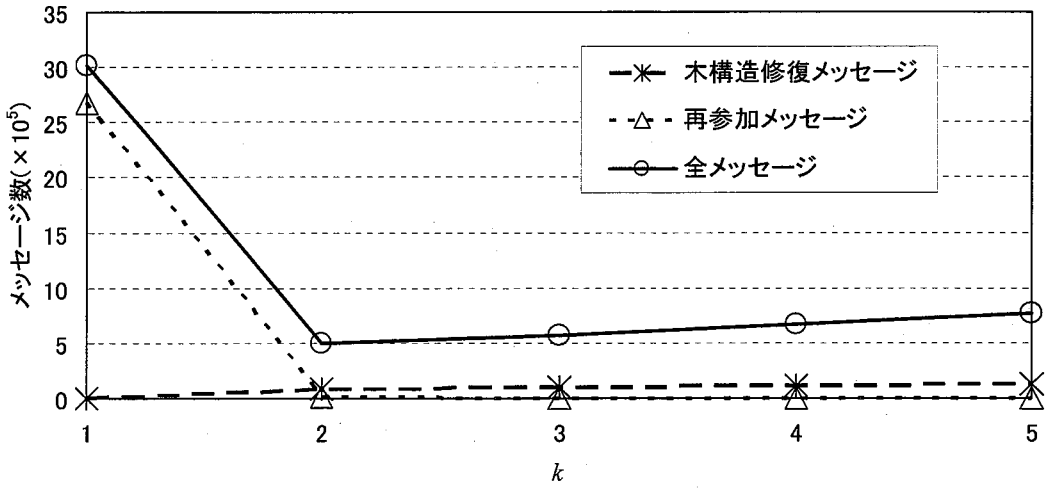


図 2.15: 管理する先祖ノードの数 k とメッセージ数（異常脱退確率：0.01）

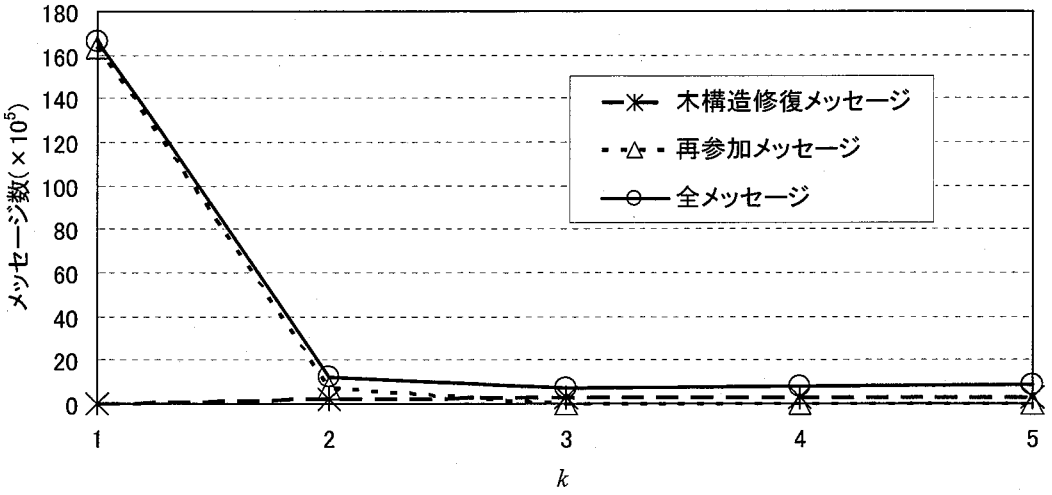


図 2.16: 管理する先祖ノードの数 k とメッセージ数（異常脱退確率：0.1）

脱退の発生確率が低い環境では更新伝播木の分断が発生しにくく，$k = 2$ でも高確率で更新伝播木の修復を行うことができる．逆に，異常脱退の発生確率が高い環境では，k の値を大きくする必要があり，図 2.16 では，$k = 3$ のときにメッセージ数が最小となっている．

全メッセージに関しては，図2.14のように異常脱退発生確率が低い環境では，kの値を大きくした場合，局所的に更新伝播木を修復することによるメッセージ数の減少量よりも，先祖ノード情報の更新によるメッセージ数の増加量が大きくなる．したがって，kが4以上であれば，$k=1$のときよりも多くのメッセージ交換が必要になってしまう．逆に，図2.15や図2.16のように異常脱退発生確率が高い環境では，更新伝播木の分断が発生しやすいため，kの値を大きく取ることが有効である．したがって，図2.14および図2.15では$k=2$が，図2.16では$k=3$のときにメッセージ数が最小となっている．全体としては，ピアの異常脱退を即座に検出できる環境においては，$k=2$であれば十分であるといえる．

このように，UPT法では，k個以上の連続する先祖ノードの異常脱退が発生した場合にフラッディング操作が必要であり，多くのメッセージが発行される．それに対し，$k-1$個までの連続する先祖ノードの異常脱退であれば，管理している周辺ピアの情報のみを用いて，少ないメッセージ数で更新伝播木の修復を行うことができる．そのため，kの値を大きくすることで，更新伝播木に再参加させるために必要なメッセージ数を減少させることができる．一方，kの値が大きくなるにつれて，管理する周辺ピアの情報が増加するため，ピアの参加や脱退などによって更新伝播木の構成が変化したときに，各ピアの情報を更新するために送受信されるメッセージ数が増加する．

2.5.4 子ノードの最大数nの影響

子ノードの最大数nを変化させた場合の，UPT法の平均ホップ数と最大ホップ数，平均負荷，木構造維持に必要な全メッセージ数に対する影響を調べた．異常脱退発生確率を0.01に統一した場合の結果をそれぞれ図2.17から図2.20に示す．これらの図において，横軸はkを表し，縦軸はそれぞれ平均ホップ数，最大ホップ数，平均負荷およびメッセージ数を表す．

図2.17および図2.18より，nの大小にかかわらず，kの値が増加するにつれて平均ホップ数，最大ホップ数ともに小さくなることがわかる．また，nの値が大きいほど，ホップ数が小さく抑えられることがわかる．これは，nの値が大きいほど，各ピアが多くの子ノードをもつことができ，更新伝播木の高さを低く抑えられるためである．ここで，nの値の増加に伴い，ホップ数の減少量は小さくなっている．例えば，nの値を2から3に増加させた場合，平均ホップ数および最大ホップ数は約20% ～25% 減少している一方で，nの値を5から6に増加させても，平均ホップ数および最大ホップ数の減少量は10% 以下である．

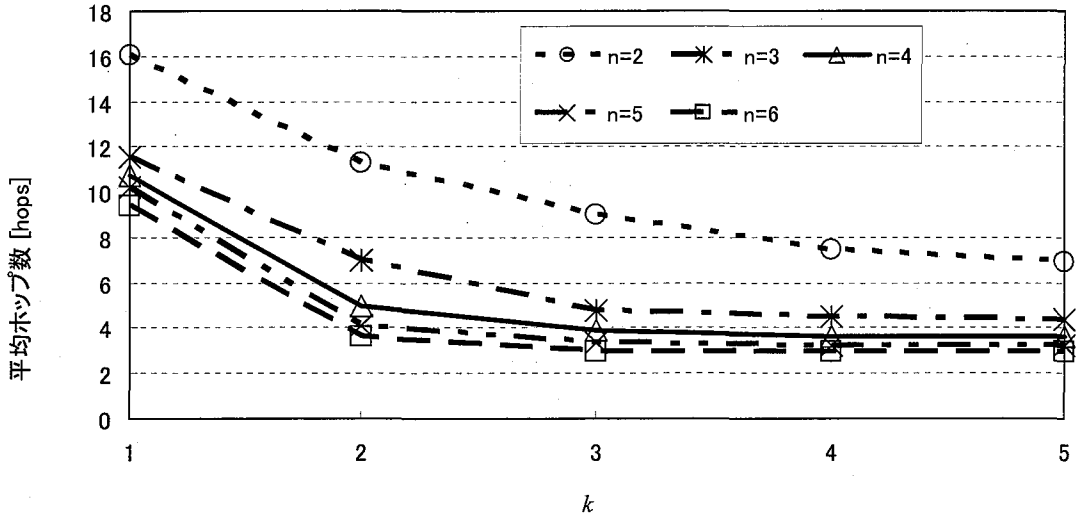


図 2.17: 子ノードの最大数 n と平均ホップ数

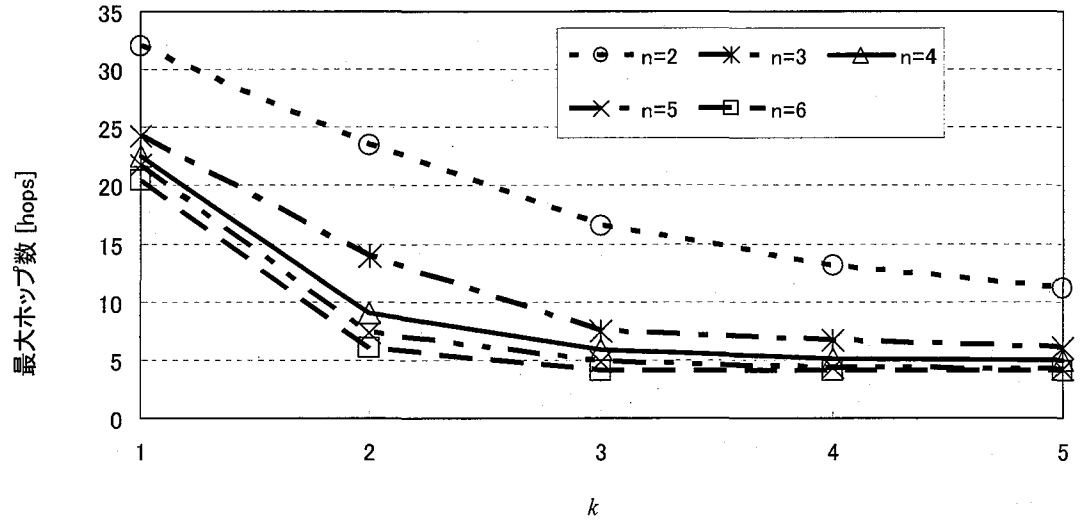


図 2.18: 子ノードの最大数 n と最大ホップ数

これは，更新伝播木に参加するピア数に起因する．更新伝播木に参加するピア数がそれほど多くない場合，更新伝播木上の各ピアの子ノードに空きが多くなり，n の値を増加させても更新伝播木の形状はほとんど変化しなくなる．

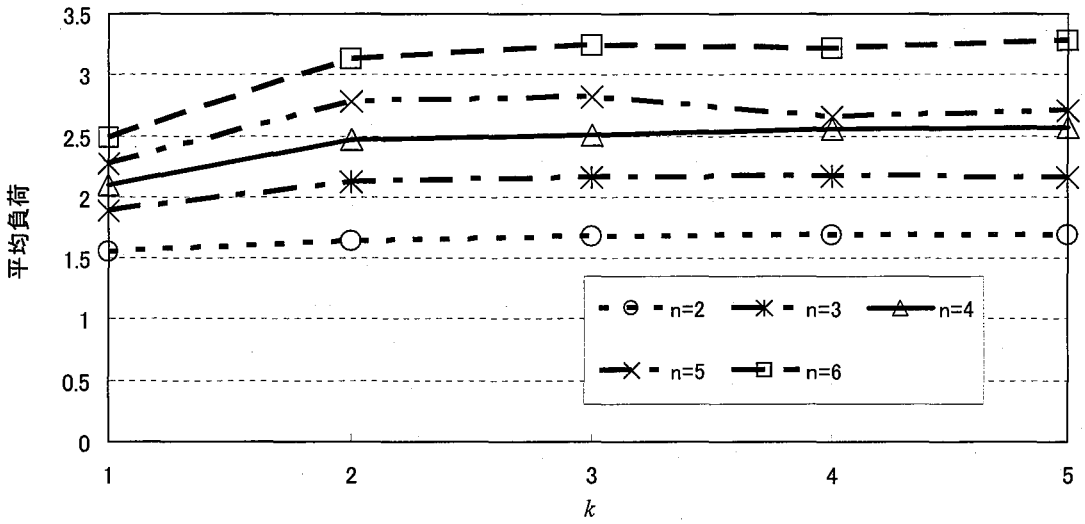


図 2.19: 子ノードの最大数 n と平均負荷

一方，図 2.19 より，n の値が増加するにつれて，更新伝播時の平均負荷が増加することがわかる．これは，n の増加によって，各ピアの子ノードの数が増加するためである．

図 2.20 より，n の大小にかかわらず，$k=1$ のときにメッセージ数が最も大きく，$k=2$ 以上の場合はメッセージ数を小さく抑えられることがわかる．また，n の値が大きいほど，木構造維持に必要なメッセージ数が小さくなっている．これは，n の値が大きくなるほど，更新伝播木の高さが小さくなることに起因する．UPT 法では，ピアの参加や脱退などによりピアの位置情報が更新された場合，新しく情報を変更されたピアから k 個の下位ノード（葉ノードまでの下位ノード数が k 個未満の場合は葉ノード）にまで，新たな位置情報を伝える必要がある．このとき，更新伝播木の高さが小さい場合は，あるピアに対して k 個下位までのノードが存在しない可能性が高いため，伝播させるメッセージ数が少なくなる．このような理由から，n の値が大きく，更新伝播木の高さが低いほど，伝播させるメッセージ数が減少する．ここで，n の値の増加に伴い，メッセージ数の減少量は小さくなっている．これは，遅延の考察で述べたことと同様に，更新伝播木に参加するピア数に起因する．更新伝播木に参加するピア数に対して n の値が大きい場合，更新伝播木上の各ピアの子ノードに空きが多くなり，n の値を増加させても更新伝播木の形状はほとんど変化しない．そのため，各ピアが管理している情報も変わらず，伝播させるメッセージ数も減少しなくなる．

以上の結果より，n の値を大きくすることにより，更新伝播時のホップ数が小さくなるだ

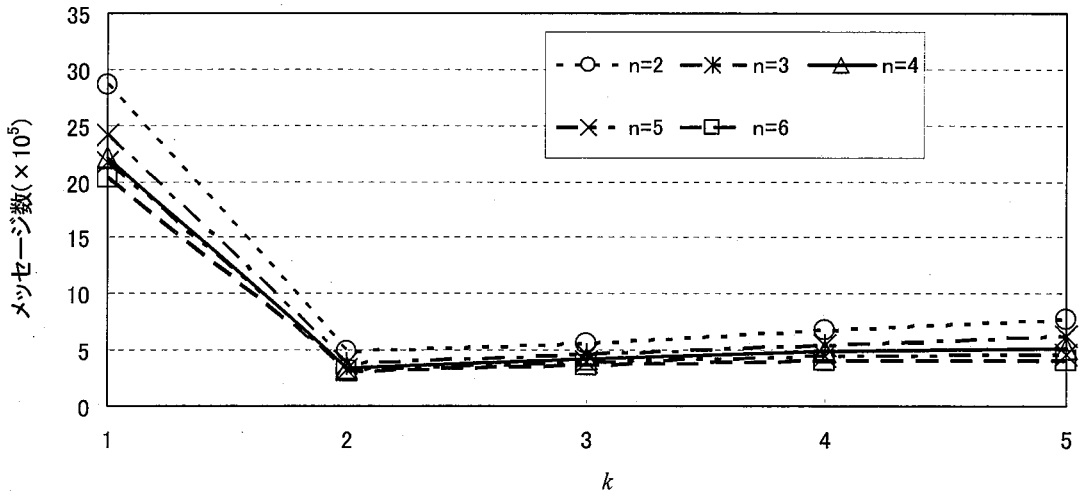


図 2.20: 子ノードの最大数 n とメッセージ数

けでなく，木構造維持に必要なメッセージ数も小さく抑えられる一方で，更新伝播時の平均負荷が大きくなることがわかる．一方，n の値を大きくしても，更新伝播時のホップ数や木構造維持に必要なメッセージ数の減少量には限りがある．したがって，各ピアのデータの送信速度や処理能力，データサイズやネットワーク上のピア数などのシステム環境を考慮して，適切な n の値を設定する必要がある．

2.5.5 異なるアクセス頻度のデータを対象とした場合の考察

本節の実験では，データ番号 100 の，アクセス頻度が比較的小さいデータに注目し，このデータに関する評価を行った．一方，アクセス頻度が高いデータに注目した場合，そのデータの複製が作成される機会が増加するため，そのデータの更新伝播木に参加するピア数が増加するものと考えられる．

ここで，個々の更新伝播木に参加するピア数が増加した場合，ネットワーク全体のピア数が増加した場合と同等の結果が得られる．図 2.10 の平均ホップ数および図 2.13 の平均負荷の結果から，UPT 法では他の手法と比べて，ネットワーク全体のピア数が増加した場合でも，負荷分散と遅延減少を両立できていることがわかる．また，UPT 法において，k や n の値を変えた場合でも，得られた結果の特徴に大きな変化はなく，更新伝播木に参加す

るピア数の増加はあまり関与していない.

以上の理由から，アクセス頻度が大きいデータを対象にした場合でもUPT法は有効である.

2.6 むすび

本章では，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，木構造に基づく複製更新伝播法であるUPT法を提案した．UPT法では，各ピアがn分木の論理ネットワークである更新伝播木を構築し，この更新伝播木に沿って更新データを伝播させることにより，更新伝播時の負荷分散と遅延減少を実現する．また，各ピアが更新伝播木上のk個上位までの先祖ノードの情報を管理することにより，ネットワーク障害などによって，ピアが正当な手続きを踏まずにP2Pネットワークから脱退した場合でも，自律分散的に更新伝播木を修復する.

UPT法の性能を評価するため，シミュレーション実験により，他の更新伝播法との比較を行った．その結果からUPT法では，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，平均ホップ数を対数オーダに抑えているなど，他の手法に比べて，更新伝播時の各ピアの負荷分散と遅延減少を両立できていることを確認した．また，kの値を大きくすることにより，更新伝播時の遅延が抑えられることを確認した．さらに，UPT法において，kの値を変化させることによるメッセージ数の変化を調べた．その結果，木構造維持に必要なメッセージ数を最小にするkの値を検証し，ピアの異常脱退の発生確率が高くなると，最適なkの値が増加することを確認した．最後に，nの値の変化による影響を調べた．その結果，nの値を大きくすることにより，更新伝播時の遅延や木構造維持に必要なメッセージ数が小さく抑えられるが，逆に更新伝播時の平均負荷が高くなることを確認した.

第3章

データアクセス頻度を考慮した複製更新伝播法

3.1 まえがき

第2章では，更新伝播時の負荷分散と遅延減少の両立を目的として木構造に基づく複製更新伝播法であるUPT法を提案した．

ここで，複製を所持するピアの中には，自分自身がその複製を頻繁に使用していたり，他のピアからのアクセスが集中するなど，自身が所持している複製へのアクセス頻度が高くなるピアが存在する．その一方で，自分自身や周辺のピアが複製にあまりアクセスしないなど，自身が所持している複製へのアクセス頻度がそれほど高くないピアも存在する．このように，所持している複製へのアクセス頻度が低いピアにおいて，データが更新されるたびに更新データを受信していても，更新と更新の間で一回もアクセスが無い場合，以前の更新データの受信は無駄となる．したがって，自身が所持している複製へのアクセス頻度が低いピアは，常には更新データを受け取らず，必要な時にのみ最新のデータを要求した方がよいことが考えられる．しかしUPT法は，複数のピアで共有されているデータが更新された場合には，そのデータの複製を所持する全てのピアに同等に更新データを伝播させている．したがって，あまりデータにアクセスしないピアにまで更新データが伝播され，余分なトラヒックが発生する．

そこで本章では，UPT法を拡張し，各ピアが所持している複製が使用される頻度（データアクセス頻度）に応じて，伝播させる更新情報を変更する手法として，UPT-HL法（Update

Propagation Trees with H peers & L peers 法）を提案する．UPT-HL 法では，データアクセス頻度が高いピアには更新データを伝播させる一方で，データアクセス頻度の低いピアには，データが古くなったことのみを通知するサイズの小さなメッセージ（無効化情報）を伝播させる．これにより，更新データを伝播させるピア数を減少させ，更新伝播時のトラヒックが削減されると同時に，更新伝播時の遅延も軽減できる．

以下では，3.2 節で，本章で提案する UPT-HL 法について説明する．3.3 節でシミュレーション実験の結果を示し，最後に 3.4 節で本章のまとめを行う．なお本章では，第 2 章と同様のシステム環境（2.2 節）を想定する．

3.2 UPT-HL 法

本節では，本章で提案する UPT-HL 法における更新伝播木の構成，更新伝播木上の各ピアが管理する情報，更新情報の伝播の様子および更新伝播木の修復方法を説明する．

3.2.1 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 3.1 に示す．UPT-HL 法では，複製を所持するピアを，データアクセス頻度が高く，常に最新のデータを必要とするピア（H ピア）と，データ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）に分類する．その後，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築する．このとき，一つのデータに対して一つの H 木を構築し，L ピアは各 H ピアに付属する形で複数の L 木を構築する．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．

3.2.2 ピアが管理する情報

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，自身に付属する L 木の根ノードの情報を管理する．ここで，オリジナル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，H 木における先祖ノードの数が k 個未満の場合は，先祖ノード情報としてオリジナル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

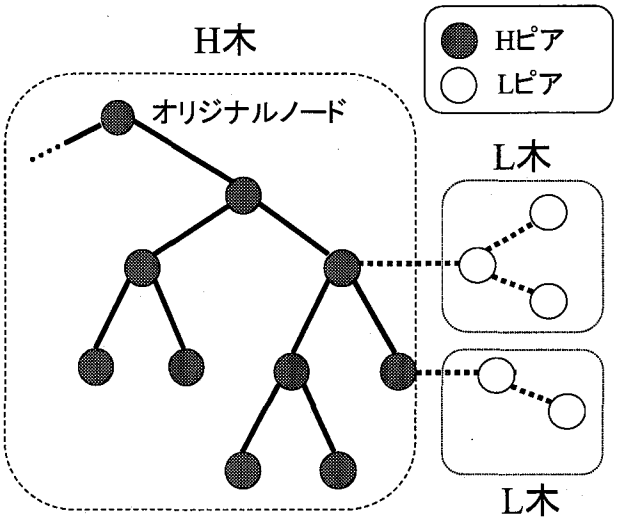


図 3.1: 更新伝播木の構成

の情報を管理する．以降，あるL木が付属しているHピアを，そのL木の責任Hピアと呼ぶ．

3.2.3 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3.2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，自身の所持するデータの複製を更新し，H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，自身に付属する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．ただし，複製を無効化したLピアは，自身の責任Hピアに関する情報は破棄せず，引き続きL木に参加し続ける．これにより，他のピアからのクエリを受け取った場合に，そのクエリを責任Hピアに転送することにより，検索時に発行されるクエリ数を削減できる．また，複製を無効化したLピアが最新のデータを必要とする場合も，自身の責任Hピアにデータを要求し，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，更新伝播時のトラヒックやそれにともなう遅延を抑えることができる．

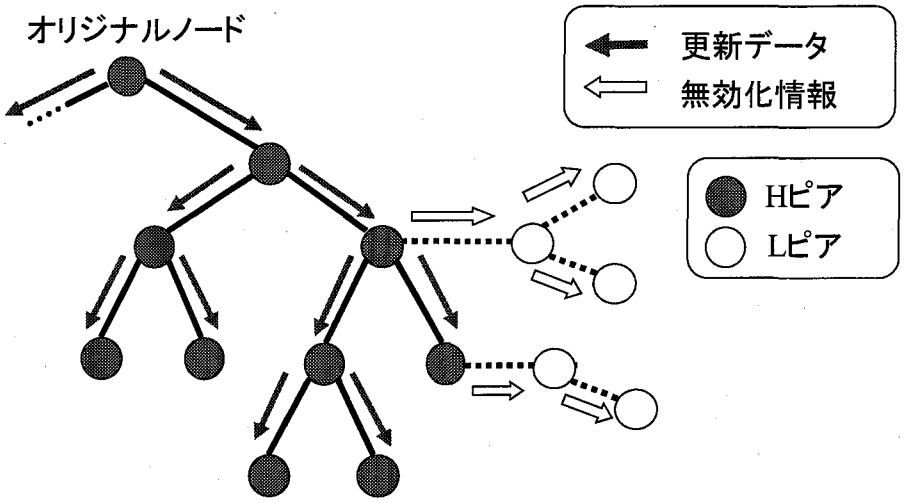


図 3.2: 更新情報の伝播

3.2.4 ピアの参加

UPT 法と同様，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新規参加ピアは，まずH ピアとしてH 木に参加する．新規参加ピアのH 木への参加手順は以下の通りである．

1. 新規参加ピアからのクエリ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．クエリ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 ピアの情報を返す．クエリを受け取ったピアが無効化情報を受信済みのL ピアであれば，そのL ピアは自身の責任H ピアの情報のみを返す．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ピアと呼ぶ．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 ピアが責任ピアとなる．

2. 責任ピアは，2.4.1 項で述べたように，UPT 法と同様の手順に従って，新規参加ピアをH 木に参加させる．

3.2.5 ピアの脱退

複製の置換えなどにより，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，更新伝播木の修復を行う．H木，L木の修復手順は，それぞれ以下の通りである．

Hピアの脱退

H木から脱退するHピアは，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，H木から脱退した後，そのL木を他のHピアに付け替える．

1. 脱退ピアがH木上の葉ノードの場合，脱退ピアは，自身の親ノードにH木からの脱退を通知する．脱退メッセージを受け取った親ノードは，自身のもつ子ノードに関する情報から脱退ピアを削除する．その後，手順4.に進む．

2. 脱退ピアが子ノードをもつ場合，脱退ピアは，自身の子ノードの中からランダムに一つを選択し，脱退メッセージを送信する．脱退メッセージ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，脱退メッセージを伝播させる．ここで，脱退メッセージを受け取った葉ノードを入替えピアと呼ぶ．入替えピアは，自身と脱退ピアの位置を入れ替える．入替えピアの親ノードは，自身のもつ子ノードに関する情報からその入替えピアを削除し，手順3.に進む．

3. 脱退ピアの位置に移動した入替えピアは，新たな位置における親ノードと，k個下位の子孫ノードにまで，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，自身のもつ子ノードおよび先祖ノードに関する情報に含まれる脱退ピアを，入替えピアの情報に変更する．その後，脱退ピアにL木が付属している場合は手順4.に進み，脱退ピアにL木が付属していない場合は脱退処理を終了する．

4. 脱退ピアは，入替えピア（脱退ピアがH木上の葉ノードの場合はその親ノード）に，自身に付属しているL木の参加メッセージを送信する．参加メッセージを受け取った入替えピアは，自身にL木が付属していない場合は，脱退ピアに付属していたL木を自身に直接付属させ，脱退処理を終了する．自身にL木が付属している場合，そのL木の根ノードに参加メッセージを送信し，手順5.に進む．

5. 参加メッセージを受け取ったLピアは，自身の子ノードの数に空きがある場合は，自身の子として脱退ピアに付属していたL木を接続する．子ノードの数に空きがない場合は，ランダムに一つの子ノードを選択し，参加メッセージを送信する．この操作を繰り返すことにより，脱退ピアに付属していたL木を再参加させ，脱退処理を終了する．

$n = 2$（2分木）の場合において，脱退ピアがH木の内部節点であり，脱退ピアにLピアが付属している場合の，脱退手順の動作例を図3.3に示す．まず，脱退ピアCは子ノードをもつため，図3.3(a)のように，ピアCはランダムに選択したピアEに脱退メッセージを送信する．脱退メッセージを受け取ったピアEは葉ノードであるため，ピアEが入替えピアとなり，ピアEとピアCの位置を入れ替える．ここで，脱退したピアCにはL木が付属しているため，図3.3(b)のように，ピアEにそのL木の参加メッセージを送信する．ピアEにはすでにL木が付属しているため，ピアEは，自身に付属するL木の根ノードに参加メッセージを送信する．参加メッセージを受け取ったL木の根ノードは，自身の子ノードの数には空きがあるため，図3.3(c)のように，脱退ピアに付属していたL木を自身の子として再参加させ，脱退処理を終了する．

Lピアの脱退

脱退ピアは，2.4.2項で説明したように，UPT法におけるピアの脱退と同様の手順に従ってL木から脱退する．

3.2.6 ピアの移動

3.2.1項で説明したように，UPT-HL法では，複製を所持する各ピアが，自身が所持する複製へのデータアクセス頻度を調査し，HピアとLピアを適宜移動する．UPT-HL法では，LピアがHピアに分散して付属することにより，少ないメッセージ数でHピア，Lピア間の移動を行うことができる．

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（複製）ごとに，以下に示す二つの値を保持する．

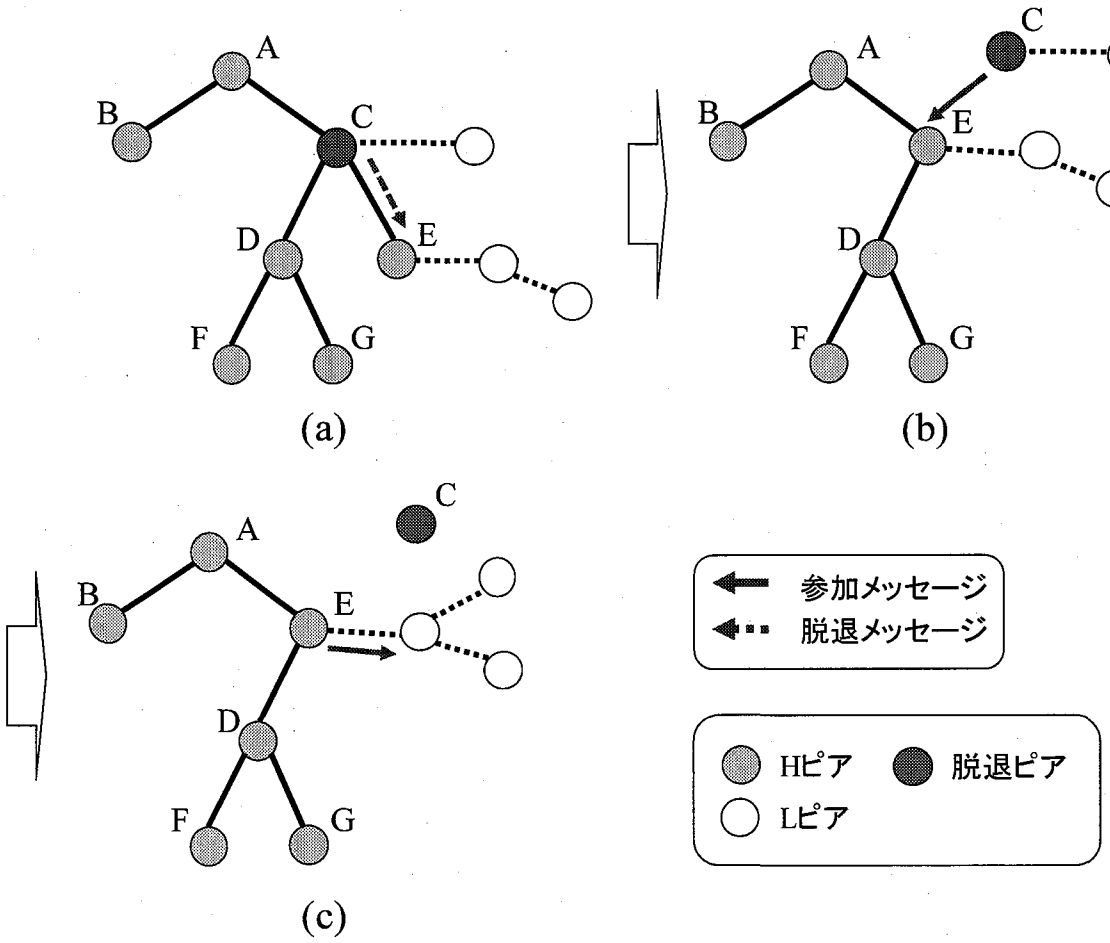


図 3.3: H ピアの脱退例

- データアクセス回数

  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

- 更新発生回数

  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）を受け取った回数の合計

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．更新情報を伝播させる際の基準として，無駄な更新データの伝播を最小限に抑えつつ，各ピアが最新のデータにアクセスできるこ

とが望ましい．ここで，一般的に，データアクセス回数の方が更新発生回数よりも大きいピアは，更新が1回発生する間に，自身が所持するデータが複数回アクセスされていることになる．つまり，更新発生時に常に更新データを受信していても，受信したデータは無駄にならない．また，常に最新のデータにアクセスできる回数が多くなる．逆に，更新発生回数の方がデータアクセス回数よりも大きいピアは，更新が1回発生する間に，自身が所持するデータがアクセスされる回数が1回未満である．つまり，更新発生時に常に更新データを受信しても，その更新データの送信が無駄になる可能性がある．また，これらのピアが所持しているデータがアクセスされる回数は少ないため，たとえそれらのデータが最新のデータでなくても，最新でないデータへのアクセス回数を小さく抑えられる．したがって，更新データの無駄な送信回数が少ないこと，および，複製にアクセスした場合にその複製が最新のデータである回数が多いことを両立するためには，データアクセス回数と更新発生回数の比のしきい値を1に設定し，ピアの移動を行うかどうかを決定することが最適である．具体的に，更新情報を受け取ったピアは，自身のデータアクセス回数と更新発生回数を比較し，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．ピアの移動の際には，移動時のメッセージ数を抑えつつ，可能な限り更新伝播木の形状を維持する必要がある．以降，移動することを決定したピアを移動ピアと表記する．UPT-HL法では，各HピアにL木が分散して付属しているため，移動ピアは，自身もしくは自身の近くに接続している別の種類のピアに移動することで，更新伝播木の形状を大幅に変更することなく，移動時のメッセージ数も抑えられる．

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，移動ピア自身もそのL木に参加する．以下に，HピアからLピアへの移動手順の詳細を説明する．

1. 移動ピアは，3.2.5項の手順1.から手順3.に従ってH木から脱退する．

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，3.2.5項の手順4.および手順5.に従って，自身に付属するL木を移動ピアに付け替える．その後，手順3.に進む．

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木の付替えを完了した移動ピアは，入替えピア（移動ピアがH木上の葉ノードの場合はその親ノード）に，Lピアとしての参加メッセージを送信する．参加メッセージを受け取った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木の根ノードとして参加させ，ピアの移動処理を終了する．自身にL木が付属している場合，手順4.に進む．

4. 参加メッセージを受け取った入替えピアは，自身に付属するL木の根ノードに参加メッセージを送信する．参加メッセージを受け取ったLピアは，自身の子ノードの数に空きがある場合は，移動ピアを自身の子ノードとして参加させる．子ノードの数に空きがない場合は，自身の子ノードの中からランダムに一つの子ノードを選択し，参加メッセージを送信する．この操作を繰り返すことにより，ピアの移動処理を完了する．

$n = 2$（2分木）の場合において，L木が付属しているHピアがLピアに移動する場合の，移動手順の動作例を図3.4に示す．まず，移動ピアBは子ノードをもつため，図3.4(a)のように，ピアBはランダムに選択したピアDに脱退メッセージを送信する．脱退メッセージを受け取ったピアDも子ノードをもつため，同様に，ランダムに選択したピアHに脱退メッセージを送信する．ピアHは葉ノードであるため，ピアHが入替えピアとなり，図3.4(b)のようにピアHとピアBの位置を入れ替える．次に，ピアBは，自身に付属するL木の参加メッセージをピアHに送信する．ピアHにはすでにL木が付属しているため，ピアHは自身に付属するL木の根ノードに参加メッセージを送信する．ここで，参加メッセージを受け取ったL木の根ノードの子の数には空きがあるため，図3.4(c)のように，ピアBに付属していたL木をピアHに付属しているL木の根ノードの子として参加させる．その後，脱退ピアB自身もピアHに参加メッセージを送信し，図3.4(d)のようにピアHに付属するL木上の一つのピアとして参加し，移動処理を終了する．

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，自身が属するL木から脱退した後，H木に参加する．以下に，LピアからHピアへの移動手順の詳細を説明する．

1. 移動ピアは，2.4.2項で説明したように，UPT法におけるピアの脱退と同様の手順に従ってL木から脱退する．

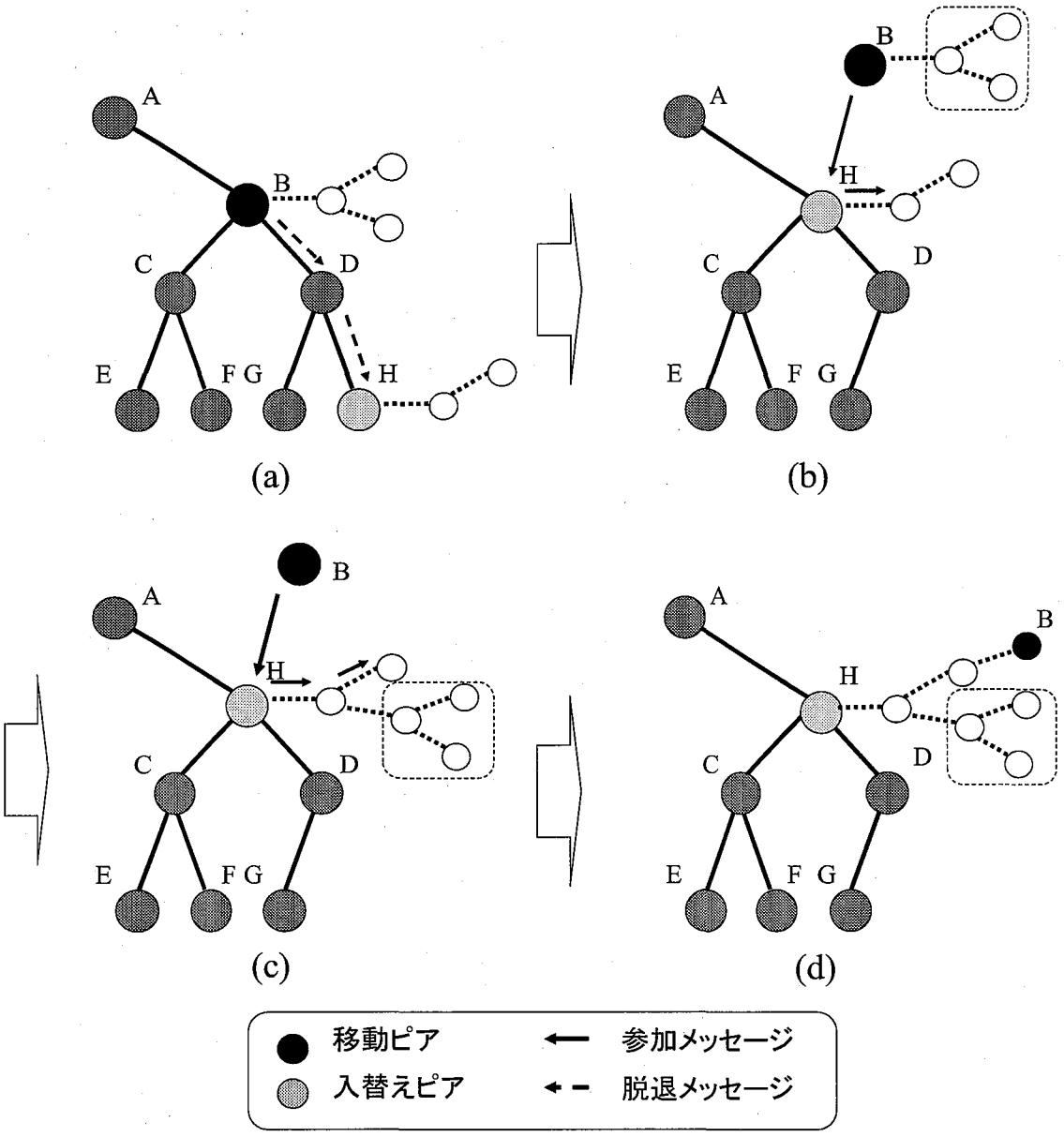


図 3.4: H ピアから L ピアへの移動

2. L 木から脱退した移動ピアは，自身の責任 H ピアに参加メッセージを送信する．その後，3.2.4 項の手順 2. と同様の手順に従って H 木に参加する．

$n = 2$（2 分木）の場合において，L 木の内部節点のピアが H ピアに移動する場合の，移動手順の動作例を図 3.5 に示す．まず，移動ピアは子ノードをもつため，2.4.2 項の手順に従い，図 3.5(a) のようにランダムに選択した子ノードの一つに脱退メッセージを送信し，脱

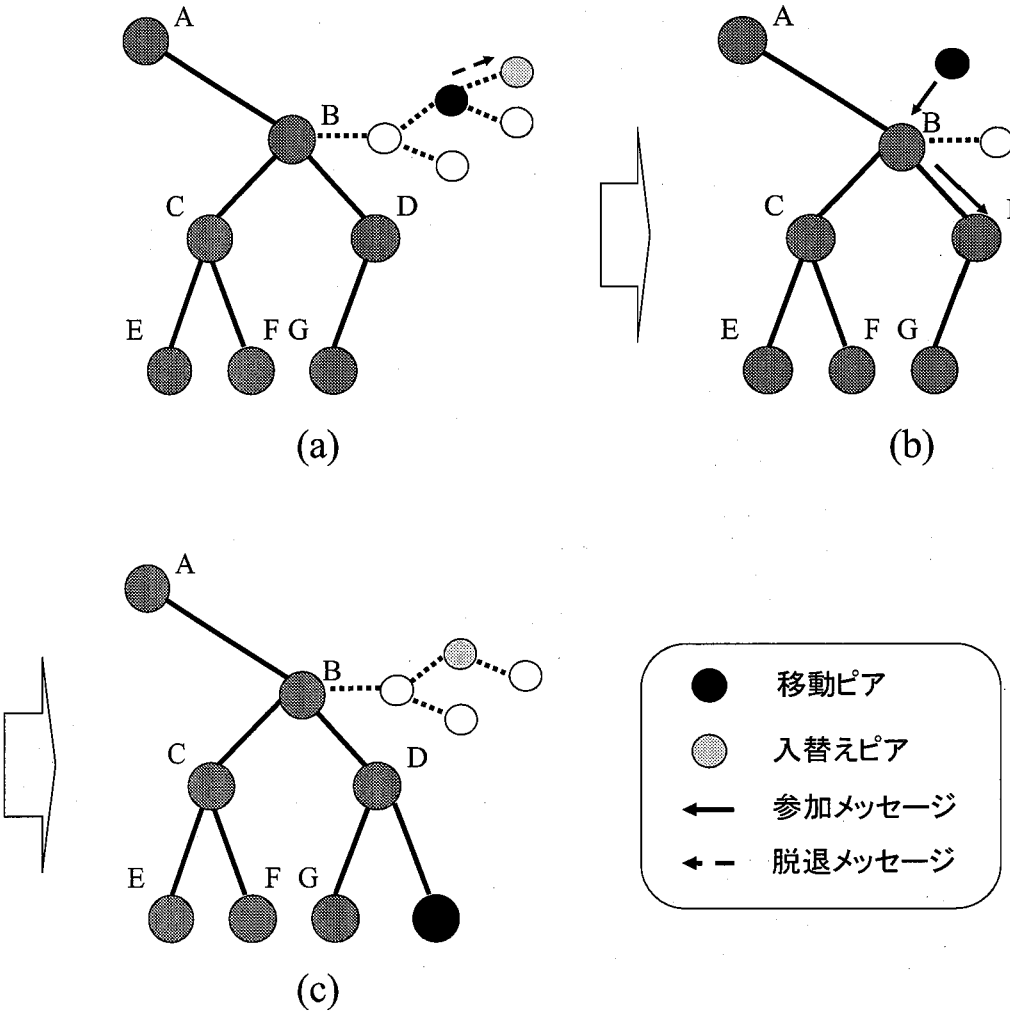


図 3.5: L ピアから H ピアへの移動

退メッセージを受け取った葉ノードと位置を入れ替える．その後，移動ピアは，図 3.5(b) のように，自身の責任 H ピアであるピア B に参加メッセージを送信する．ピア B の子ノードの数には空きがないため，ランダムに選択したピア D に参加メッセージを送信する．ピア D の子ノードの数には空きがあるため，図 3.5(c) のように，移動ピアは，ピア D の子ノードとして H 木に参加し，移動処理を終了する．

3.3 性能評価

本節では，UPT-HL法が，様々な更新発生頻度のデータに対しても，更新伝播時に発生するトラヒックや遅延，各ピアの負荷の面で有効であるか，およびこれらの値はどのように変化するかを調査する．以下では，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．

3.3.1 シミュレーション環境

シミュレーション実験では，第2章と同様，非構造型P2Pネットワークにおけるデータ共有を想定した．P2Pネットワークに参加するピア数を1,000とし，それらがべき法則に従ってネットワークを構築するものとした．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が集中する環境を実現した．

$$d_i = \lfloor 20 \cdot i^{-0.4} \rfloor. \tag{3.1}$$

データの種類を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タイムスロット毎に0.1の確率であるデータを要求する．データ要求の発行は，第2章の実験と同様，Zipf係数0.5のZipf分布に従うものとし，式(2.2)を用いて決定した．

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．

複製の配置手法には，オーナー複製法[36]を用いた．オーナー複製法では，データを発見した際，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全て等しく，複製を保有可能な数は全てのピアで10，データのオリジナルを所持するピアは，オリジナルの他に10個の複製を所持できるものとした．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でデータを更新するものとした．以降，この確率を更新発生率と表記する．

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．ただし，他のデータを対象とした場合も，データ番号1のデータと同様の特徴をもつ結果が得られると考えられる．他のデータを対象とした場合の結果に関する考察は3.3.4項で行う．

3.3.2 複製配置手法の比較

最初に，UPT法，UPT-H (UPT with H peers) 法，2.3.3項で説明したチェイン伝播法 ($m = 5$)，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，そのピアが所持する複製を削除して更新伝播木から完全に脱退する手法である．これらの実験では，データ要求確率よりも更新発生率の方が小さい場合と大きい場合を網羅するために，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法，UPT-H法，およびUPT-HL法において，$k = 1$の場合と$k = 5$の場合の性能を評価した．

- 更新情報伝播トラヒック

  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トラヒック，および，Lピアに無効化情報を伝播させることによるトラヒックのシミュレーション時間全体の合計．ただし，無効化情報のサイズに比べて更新データのサイズが比較的大きい場合を想定しているため，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トラヒックとした．

- 論理ネットワーク維持トラヒック

  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，チェイン構造）を修復するために交換されるメッセージによるトラヒック，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によるトラヒックの合計（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，伝播されるデータの総量をトラヒックとした．

- 平均ホップ数

  更新伝播時の，オリジナルノードからH木上の各ノード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.

- 検索トラヒック

  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によるトラヒック（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，伝播されるデータの総量をトラヒックとした.

ここで，チェイン伝播法は，複製を所持する全てのピアに更新データを伝播させるという点でUPT法と同じであり，更新情報伝播トラヒック，検索成功率および検索トラヒックの結果はUPT法と等しくなる．そのため，チェイン伝播法に関しては，論理ネットワーク維持トラヒックと平均ホップ数の結果のみを記載する.

更新情報伝播トラヒック

更新情報伝播トラヒックの結果を図3.6に示す．図3.6において，横軸は更新発生率を表し，縦軸は更新情報伝播トラヒックを表す．結果より，全ての手法において，更新発生率が増加するにつれて，更新情報伝播トラヒックが増大している．ここで，UPT法では更新発生率とともに更新情報伝播トラヒックが線形に増加しているのに対し，UPT-HL法とUPT-H法ではトラヒックの増加量が小さく抑えられており，UPT法と比較してUPT-H法では最大約65%，UPT-HL法では最大約75%，更新情報伝播トラヒックが減少している．UPT法では複製を所持する全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トラヒックが大きくなる．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，トラヒックが抑えられる．したがって，UPT-HL法およびUPT-H法は，更新発生率が低い場合にも有効であるが，更新発生率が高い場合により有効であることがわかる．また，UPT-HL法とUPT-H法を比較した場合，UPT-HL法におけるトラヒック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.

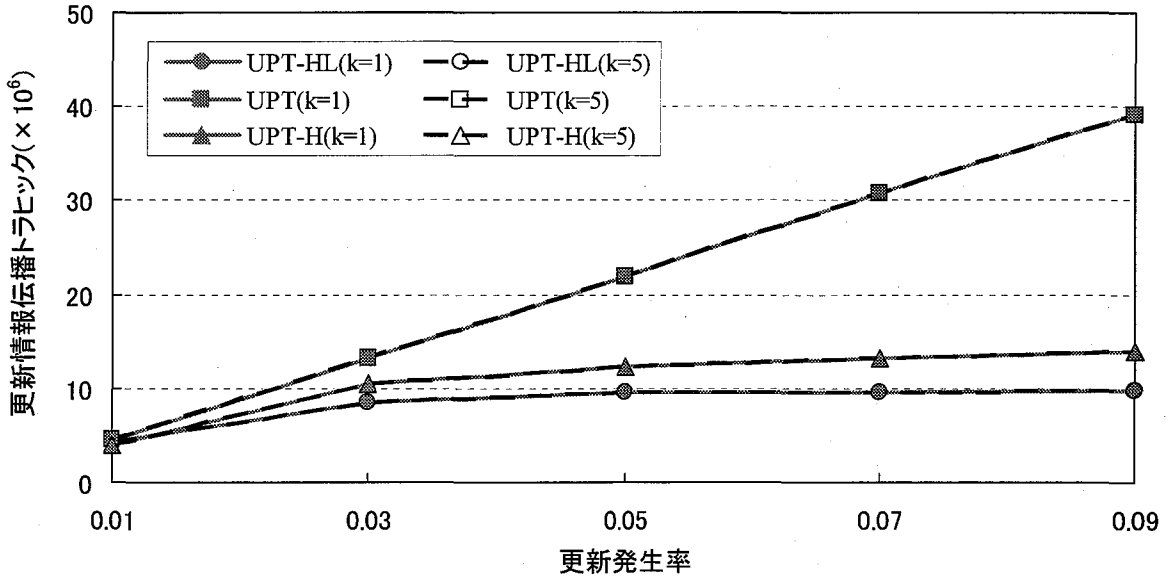


図 3.6: 更新情報伝播トラヒック

また，UPT 法，UPT-H 法，および UPT-HL 法において，$k = 5$ に設定した場合，更新情報伝播トラヒックは，$k = 1$ の場合からほとんど変化していない．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．

論理ネットワーク維持トラヒック

論理ネットワーク維持トラヒックの結果を図 3.7 に示す．図 3.7 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トラヒックを表す．結果より，UPT-HL 法では，更新発生率が増加するにつれて論理ネットワーク維持トラヒックが大きくなるが，UPT 法では，更新発生率とは関係なく論理ネットワーク維持トラヒックが小さい値で一定に保たれていると同時に，UPT-H 法でも，トラヒックの増加量が極めて小さく抑えられている．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 法では，ピアの入替えは行わず，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

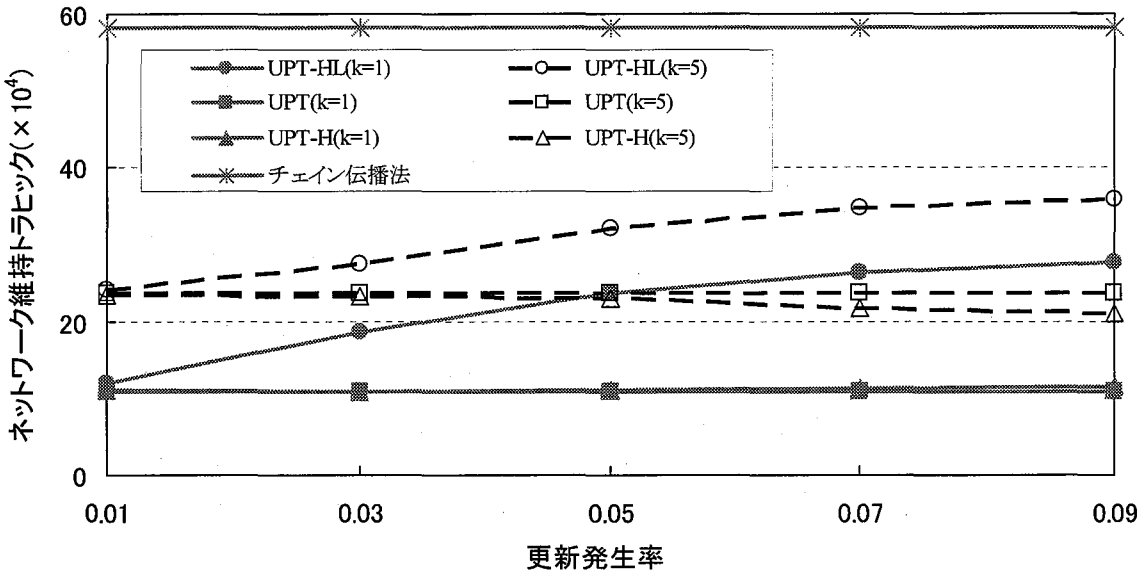


図 3.7: 論理ネットワーク維持トラヒック

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．チェイン伝播法では，参加，脱退の度に左右 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三つの手法に比べて論理ネットワーク維持トラヒックが大きくなる．

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する．しかし，図 3.6 および図 3.7 の結果からわかるように，論理ネットワーク維持トラヒックの増加量に比べ，更新情報伝播トラヒックの減少量の方が大きく，全体のトラヒックは減少する．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トラヒックが減少することが確かめられる．ここで，図 3.6 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，更新情報伝播トラヒックがすべての手法において減少する．例えば，更新データのサイズを 50 とした場合は，更新情報伝播トラヒックはおよそ半分になる．一方，図 3.7 の結果は更新データのサイズに依存しないため，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トラヒック量が逆転し，UPT-HL 法におけるトラヒック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

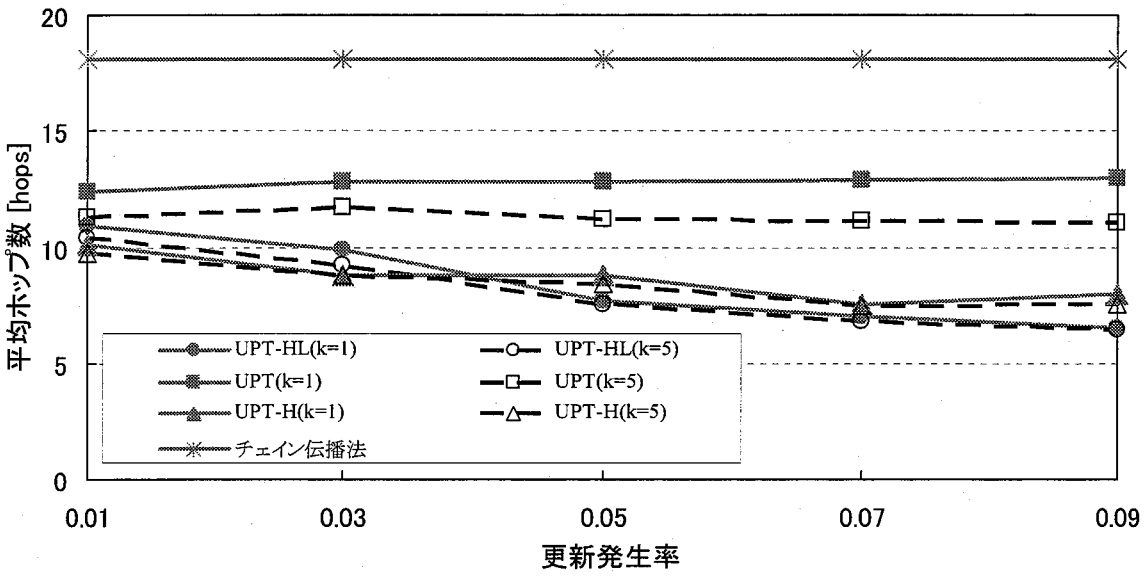


図 3.8: 平均ホップ数

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 法，UPT-H 法，および UPT-HL 法において $k = 5$ に設定した場合，論理ネットワーク維持トラヒックが $k = 1$ 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各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が多くなるためである．

更新伝播時の平均ホップ数

更新伝播時の平均ホップ数の結果を図 3.8 に示す．図 3.8 において，横軸は更新発生率を表し，縦軸は平均ホップ数を表す．結果より，更新発生率によらず，UPT 法に比べ，UPT-HL 法と UPT-H 法におけるホップ数が小さくなっている．これは，UPT 法では複製を所持する全てのピアに更新データを伝播させなければならないのに対し，UPT-HL 法および UPT-H 法では，更新データを伝播させるべきピアが H ピアのみに限定されるためである．また，これら二つの手法では，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，更新データを伝播すべきピア（H ピア）の数がより減少する．そのため，更新発生率が大きくなるにつれて，平均ホップ数が小さくなる．

チェイン伝播法では，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，ホップ数が大きくなる．m の値を大きくすることでホップ数は減少するが，その場合は前述した論理ネットワーク維持トラヒックが増加する．

一方，$k = 5$ に設定した場合，$k = 1$ の場合に比べて，UPT 法では多少ホップ数が減少するものの，UPT-H 法，および UPT-HL 法における平均ホップ数の減少がわずかであることがわかる．前述したように，k の値を大きくすることによって更新伝播木の高さを抑制できるが，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 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，平均ホップ数が減少しにくくなっている．

ただし，2.5.2 項でも述べたように，論理ネットワーク上でのホップ数が小さい場合に，必ずしも物理ネットワーク上での遅延も小さくなるとは限らないため，物理ネットワークを考慮したさらなる検証が必要である．

検索成功率

更新発生率が 0.01, 0.05, および 0.1 の場合の検索成功率を図 3.9 に示す．図 3.9 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 ピアの情報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，前述したように，k の値を変化させても，複製を所持しているピア数は変化しないため，検索成功率は k の値に依存しない．そのため，本項では $k = 1$ の場合の結果のみ示す．

まず，図 3.9(a) より，更新発生率が小さい環境において，三つの手法で検索成功率に大きな差が見られないことがわかる．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，三つの手法における更新伝播木に参加するピア数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 3.9(b) および (c) より，更新発生率が大きくなるにつれて，UPT-H 法における各 TTL での検索成功率が低下しており，更新発生率が 0.05 の場合に最大約 20%，0.1 の場合に最大約 40% 減少している．これは，UPT-H 法では，更新発生率が大きい環境では，

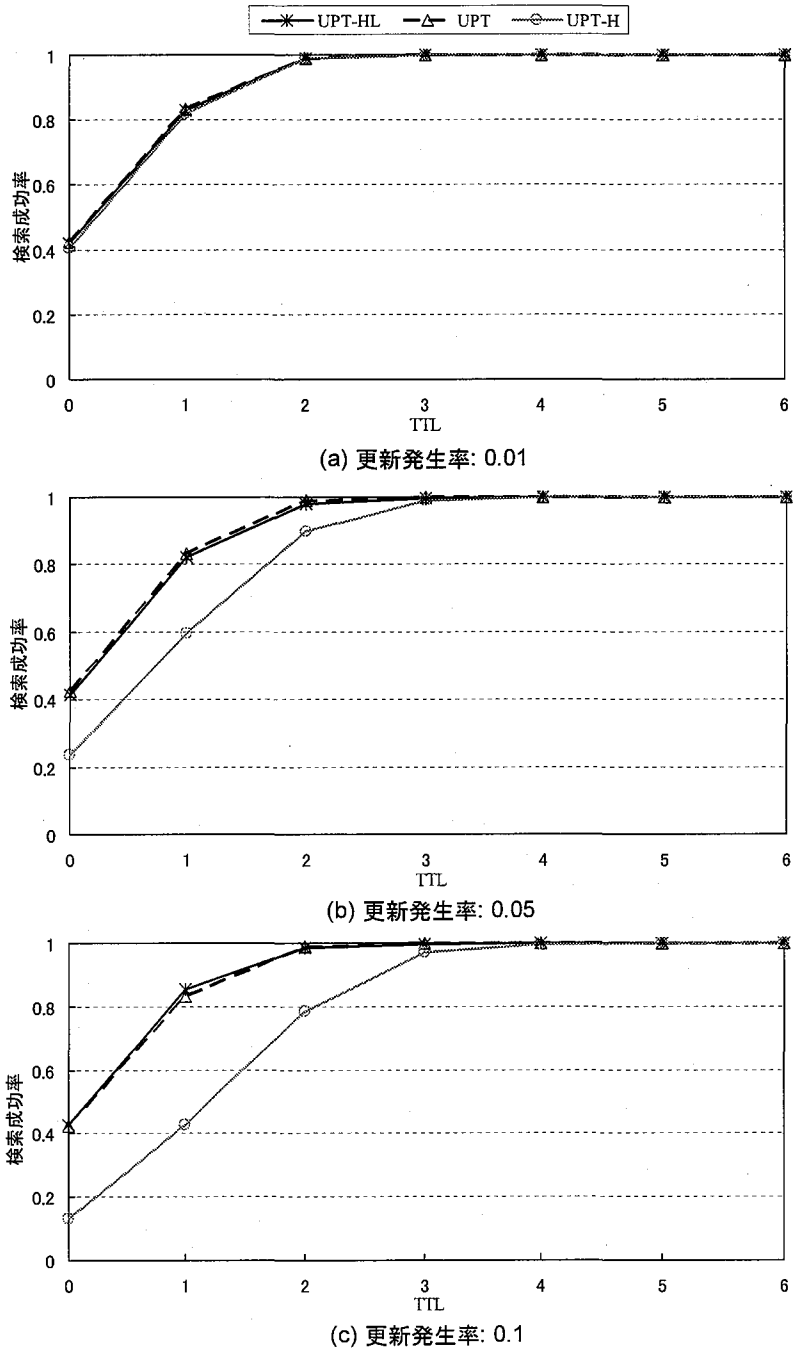


図 3.9: 検索成功率

複製を所持する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．

検索トラヒック

更新発生率が0.01, 0.05, および0.1の場合の検索トラヒックを図3.10に示す．図3.10において，横軸はTTLを表し，縦軸は検索トラヒックを表す．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によるトラヒックを表し，折れ線グラフは，その累計を表す．本項においても，検索成功率の評価の項と同様の理由により，$k=1$の場合の結果のみ示す．

図3.10より，UPT法に比べて，UPT-HL法の検索トラヒックは，最大約50%増加している．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ピアにクエリを転送す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トラヒックが大幅に増加している．また，更新発生率の増加に伴って，その差が顕著となっている．例えば，更新発生率が0.01の場合，UPT-H法における検索トラヒックは，UPT法における検索トラヒックの約1.2倍である一方で，更新発生率が0.1の場合，UPT-H法における検索トラヒックは，UPT法における検索トラヒックの5倍以上になっている．これは，図3.9の結果からわかるように，UPT-H法では，他の手法と比べて検索成功率が低く，データを発見するまでに大量のクエリが発行されるうえ，更新発生率の増加に伴い，UPT-H法における検索成功率がさらに低下するためである．

3.3.3 Hピアの負荷

UPT-HL法では，Hピアの数が少なくなると，更新データ伝播時のトラヒックや遅延が減少するが，一つのHピアにクエリが集中しやすくなる．そこで，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．

- 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ピアである時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3.11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3.11において，横軸はピア番号を表し，縦

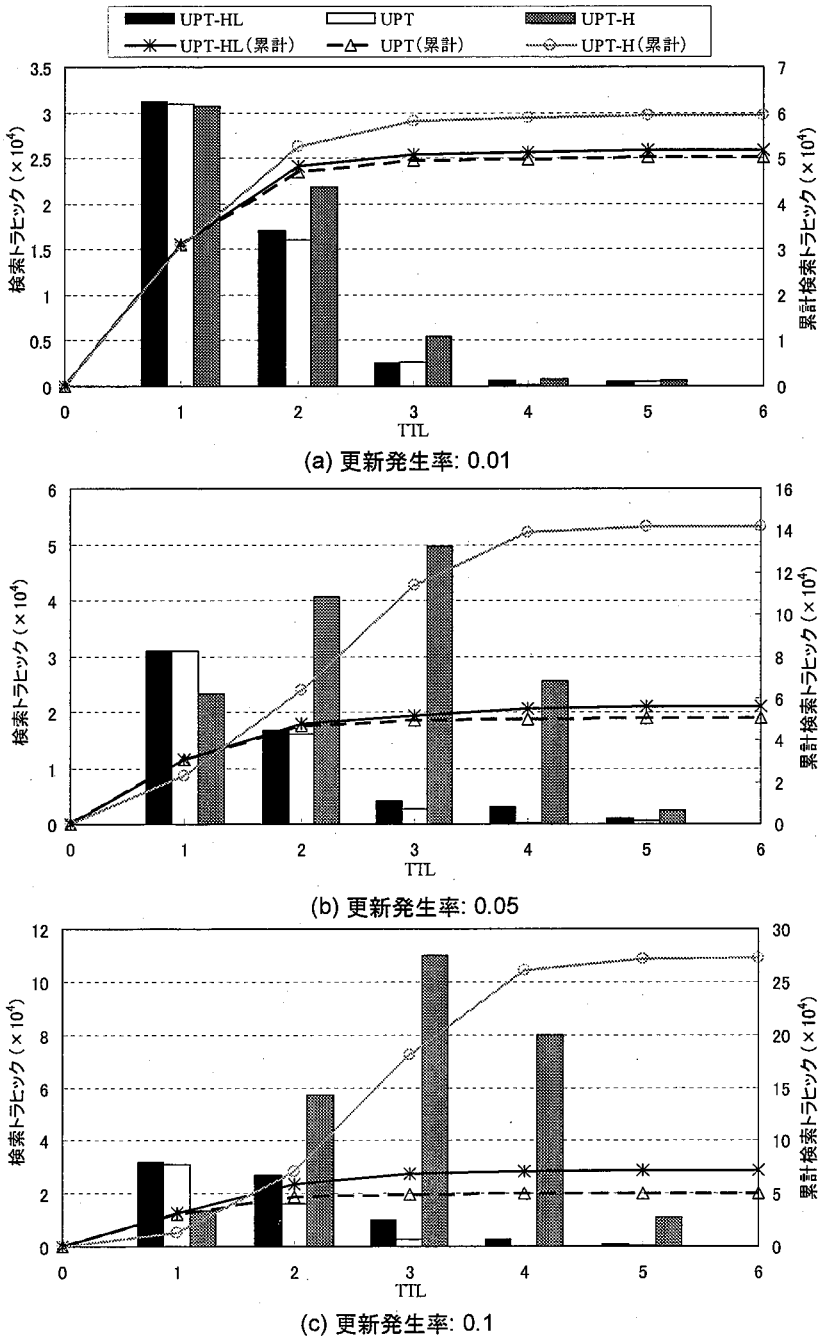


図 3.10: 検索トラヒック

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，一つのHピアが受け取るクエリ数が多くなる一方で，更新発生率が低い場合，受け取るクエリ数が少なくなっている．

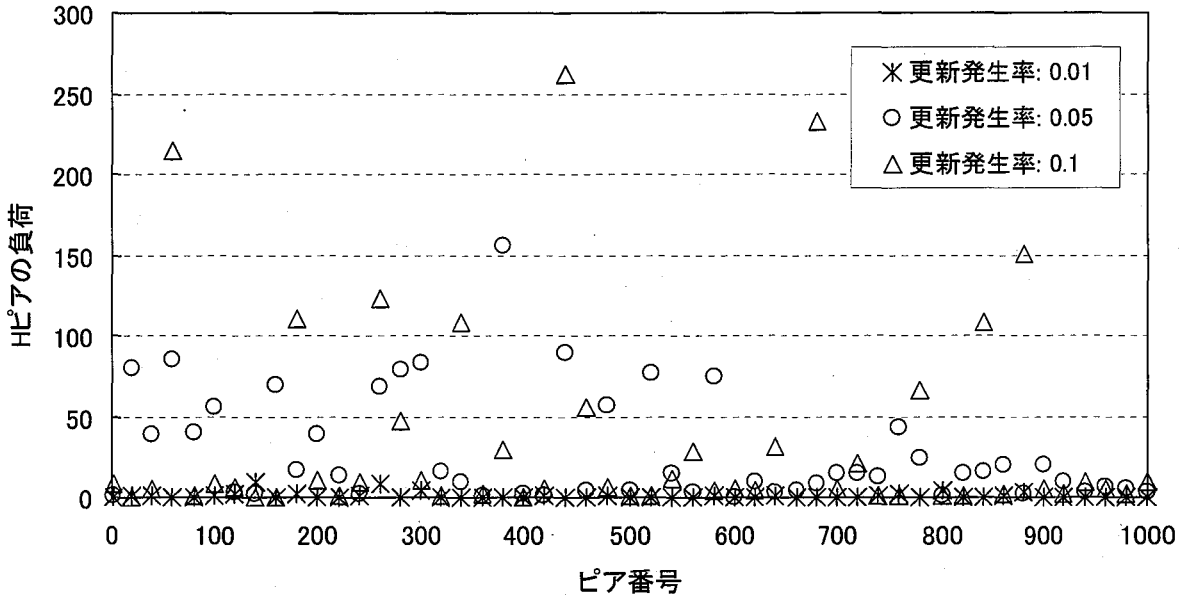


図 3.11: H ピアの負荷

これは，更新発生率が高い場合，L ピアの割合が多くなり，一つの 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 ピアからの最新データの要求に応答するため，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 0.1 の場合，極めて負荷の高い 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，一つの H ピアに負荷が集中することがなくなっていることがわかる．

3.3.4 様々な環境における提案手法の有効性に関する考察

本節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．本項では，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．

データアクセス頻度の低いデータを対象とした場合

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．3.3.2 項の結果からわかる通り，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．

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に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，更新データを伝播す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，更新伝播時に発生するトラヒック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トラヒックも小さくなるものと考えられる．

一方，本節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も，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．

3.4 むすび

本章では，ピアが共有するデータに更新が発生する P2P ネットワーク環境を想定し，更新データ伝播時に発生するトラヒックや遅延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，第 2 章で提案した UPT 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構築する．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させる一方で，データアクセス頻度の低いピアにはサイズの小さい無効化情報のみを伝播させることで，更新伝播時のトラヒックや遅延を減少させる．

UPT-HL 法の性能を評価するため，シミュレーション実験により，他の更新伝播法との比較を行った．その結果から，UPT-HL 法では，木構造維持のためのトラヒックが増加す

る一方で，更新データ伝播によるトラヒックを大幅に削減でき，ネットワーク全体のトラヒックを削減できていることを確認した．また，同時に，更新伝播時の遅延も抑制できていることを確認した．次に，検索効率および検索時に発行されるメッセージ数を調べた．その結果，UPT-HL 法では，データの検索効率を下げることなく，検索時に発行されるメッセージ数を小さく抑えられることを確認した．最後に，更新発生率を変化させることによる，各 H ピアの負荷への影響を調べた．その結果，更新発生率が大きい場合には，各 H ピアの負荷が増加することを確認した．

第4章

データの更新量を考慮した複製更新伝播法

4.1 まえがき

第3章では，第2章で提案したUPT法を拡張し，データアクセス頻度を考慮した複製更新伝播法であるUPT-HL法を提案した．UPT-HL法では，各ピアのデータアクセス頻度に応じて伝播させるデータを変更することで，更新伝播時のトラヒックの削減とそれにともなう遅延の減少を実現した．一方，複製所持ピアの中には，データが更新された際，その更新内容が微々たるものである場合には更新データを受信せず，データが大きく更新された場合にのみ自身の複製を更新することを望むものが存在する．例えば，ソフトウェアのアップデートが発生した場合，その内容が軽微な機能拡張のみであれば，そのアップデートを必要としないユーザが多く存在するものと考えられる．また，気象図や道路状況などの交通情報図などにおいても，微小な変化が頻繁に発生することが考えられるため，ユーザが指定している程の大きな変化が発生した場合にのみ最新のデータを必要とすることが考えられる．このような環境では，各複製所持ピアが，データがどの程度更新された場合に更新データを受信するかという条件（データ要求条件）を設定しておき，この条件を満たす場合にのみ更新データを受信することで，データ更新にともなうネットワーク上のトラヒックを削減できる．

そこで，本章では，データの更新量を考慮した二つの更新伝播手法を提案する．一つ目の手法であるUPDD-S (Update Propagation strategy considering Degree of Data update with Same-condition trees) 法では，同一のデータ要求条件を設定しているピアごとに木構造（同条件更新伝播木）を構築する．更新発生時にはデータのオリジナルを所持するオ

リジナルノードがそれぞれの同条件更新伝播木の根ノードに直接更新データを送信することにより，更新データを必要としていないピアへの更新伝播を抑制する．しかしこの手法では，オリジナルノードの負荷が非常に高くなってしまう．そこで，二つ目の手法であるUPDD-SO (UPDD with Same-condition trees and Ordered-condition trees) 法では，同条件更新伝播木に加え，同条件更新伝播木の根ノードで構成される新たな木構造（条件順序木）を構築することにより，複製所持ピアを階層的に管理する．これにより，UPDD-S 法と比較して，オリジナルノードの負荷を軽減する．

以下では，4.2 節で想定環境を説明する．4.3 節で本章で提案する一つ目の手法であるデータの更新量を考慮した複製更新伝播法である UPDD-S 法について説明し，4.4 節で UPDD-S 法を拡張した UPDD-SO 法について説明する．4.5 節でシミュレーション実験の結果を示し，最後に 4.6 節で本章のまとめを行う．

4.2 想定環境

本章では，第 2 章や第 3 章と同様，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所持するデータにアクセスする環境を想定する．また，各ピアは，他のピアが所持する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．

本章における提案手法では，データの更新前と更新後の差分を数値で表すことができるデータを対象とし，この差分値をデータの更新量と定義する．例えば，アップデートされたソフトウェアや画像データなどは，更新によって変更されたファイルサイズやその割合などでデータの更新量を表すことができる．また，オンラインショップの価格情報や株価などの数値データは，数値の変化量をデータの更新量として表すことができる．なお，本章では，説明の容易さから，数値データを扱い，その変化量がデータの更新量である場合を想定して説明する．

各複製所持ピアは，オリジナルデータが自身が所持する複製からどの程度変化した場合に更新データを受信するか，という差分の値をデータ要求条件として指定する．図 4.1 にデータ要求条件の例を示す．この図では，現在のオリジナルデータおよびそのデータの複製の値が共に 230 であり，複製を所持しているピアは自身のデータ要求条件を 10 に設定している．この場合，複製所持ピアは，オリジナルデータと自身が所持している複製の値の差

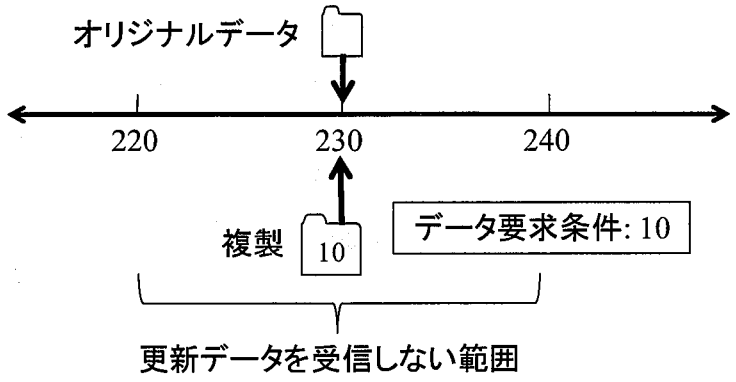


図 4.1: データ要求条件

が 10 以上（オリジナルデータの値が 240 以上もしくは 220 以下）になった場合，更新データを受信するものとする．

4.3 UPDD-S法

UPDD-S 法では，データが更新された場合に，その更新データを必要としていないピアへの更新データの伝播を抑制することにより，更新伝播時のトラヒックを削減する．具体的には，同一のデータ要求条件を設定しているピアで構成される木構造（同条件更新伝播木）を構築し，それぞれの同条件更新伝播木の根ノードがオリジナルノードと接続する．本章では，同条件更新伝播木は全て 2 分木として説明する．

図 4.2 に，UPDD-S 法における更新伝播木の構成例を示す．図 4.2 において，各ピアはそれぞれ 2 から 21 までのデータ要求条件を設定しており，同条件更新伝播木 T_{S1}, T_{S5}, T_{S6} に参加するピアは値が 0 のデータ（複製）を，同条件更新伝播木 T_{S2}, T_{S3}, T_{S4} に参加するピアは値が 3 のデータ（複製）を所持している．なお，これ以降の図において，同条件更新伝播木 T_{S1} や T_{S2} の根ノードをそれぞれ R_{S1} および R_{S2} と表記する．

4.3.1 ピアが管理する情報

UPDD-S 法では，更新伝播木を維持するために，各複製所持ピアは更新伝播木上の周辺ピアの情報（例 : IP アドレス）を管理する．具体的に，オリジナルノードは，全ての同条件

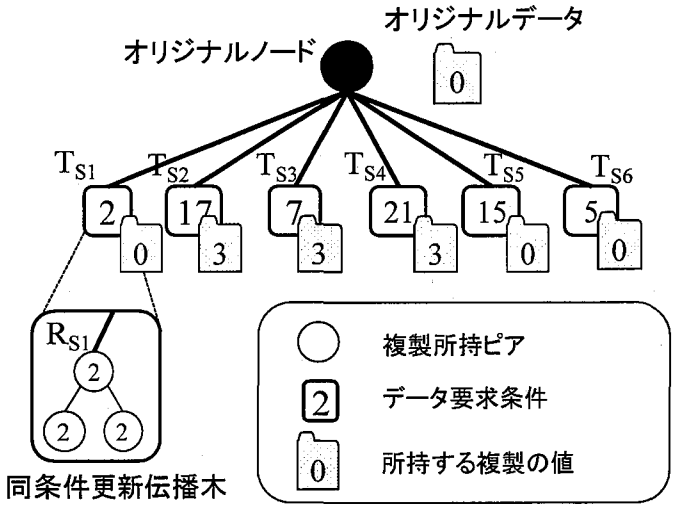


図 4.2: 更新伝播木の構成（UPDD-S 法）

更新伝播木の根ノードの情報を管理する．同条件更新伝播木の根ノードは，オリジナルノードおよび同条件更新伝播木上の自身の子ノードの情報を管理し，根ノード以外の複製所持ピアは，同条件更新伝播木上の親ノードおよび子ノードの情報を管理する．また，UPDD-S 法では，オリジナルデータが更新された場合，その時点で更新データを必要としているピアにのみ更新データを伝播させる．そのために，オリジナルノードは，各同条件更新伝播木に属するピアが更新データを必要としている条件を更新伝播先リストとして管理する．

図 4.2 の表に，更新伝播先リストの例を示す．この図において，オリジナルデータの値 (0) と同条件更新伝播木 T_{S1} に属するピアが所持する複製の値 (0) との差は 0 であり，同条件更新伝播木 T_{S1} に属するピアが設定しているデータ要求条件は 2 である．この場合，同条件更新伝播木 T_{S1} に属するピアは，オリジナルデータの値が 2 以上増減した場合に更新データを受信する必要がある．したがって，オリジナルノードは，同条件更新伝播木 T_{S1} の情報として ± 2 を更新伝播先リストに格納する．同様に，オリジナルデータと同条件更新伝播木 T_{S2} に属するピアが所持する複製の値 (3) との差は 3 であり，オリジナルデータの値が 20 以上もしくは −14 以下に変更されたとき，二つのデータの値の差が 17（同条件更新伝播木 T_{S2} に属するピアが設定しているデータ要求条件）以上となる．したがって，オリジナルノードは，同条件更新伝播木 T_{S2} の情報として (+20, −14) を更新伝播先リストに格納する．

4.3.2 更新情報の伝播

更新発生時には，オリジナルノードは更新伝播先リストを参照し，その更新データを必要としているピアで構成される同条件更新伝播木の根ノードに更新データを送信する．具体的には，更新伝播先リストに格納された各同条件更新伝播木の条件の値を調べ，オリジナルデータの更新量がその値よりも大きい場合にのみ，その同条件更新伝播木の根ノードに更新データを送信する．更新データを受け取ったピアは，そのデータを同条件更新伝播木に沿って伝播させる．

また，オリジナルデータが更新された場合，オリジナルノードは更新データを必要としている同条件更新伝播木の根ノードに更新データを送信すると同時に，自身の更新伝播先リストの情報を更新する．図4.3に，UPDD-S法における更新データの伝播の様子を示す．オリジナルデータの値が0から5に更新された場合，オリジナルノードは，自身の更新伝播先リストを参照し，データ要求条件を満たす（5以下の値を更新伝播先リストに格納しているピア，ただし負の値は除く）同条件更新伝播木 T_{S1} と T_{S6} の根ノードに更新データを送信する．その後，オリジナルノードは，更新伝播先リストの情報を更新する．具体的には，更新データを伝播させた同条件更新伝播木 T_{S1}, T_{S6} に関しては，オリジナルデータとそれらの木に属するピアが所持する複製の値の差は0であるため，それらの木構造の情報として，その木構造に属するピアが設定しているデータ要求条件の値，およびその負の値を設定する．つまり，同条件更新伝播木 T_{S1} および T_{S6} の情報として，それぞれ±2および±5を設定する．一方，更新データを伝播させなかった同条件更新伝播木 T_{S2}, T_{S3}, T_{S4}, T_{S5} に関しては，オリジナルデータとそれらの木に属するピアが所持する複製の値の差が変化する．例えば，同条件更新伝播木 T_{S2} の場合，オリジナルデータの値が0から5に変化することにより，オリジナルデータの値 (5) と同条件更新伝播木 T_{S2} に属するピアが所持する複製の値 (3) の差が2となる．この場合，もしオリジナルデータがその値からさらに15以上増加（オリジナルデータの値は20以上となる）もしくは19以上減少された場合（オリジナルデータの値は−14以下になる），オリジナルデータと同条件更新伝播木 T_{S2} に属するピアが所持する複製の値の差が17（同条件更新伝播木 T_{S2} に属するピアが設定しているデータ要求条件）を超える．したがって，オリジナルノードは，同条件更新伝播木 T_{S2} の情報を，(+15, −19) に変更する．同様に，同条件更新伝播木 T_{S3}, T_{S4}, T_{S5} の情報も更新する．

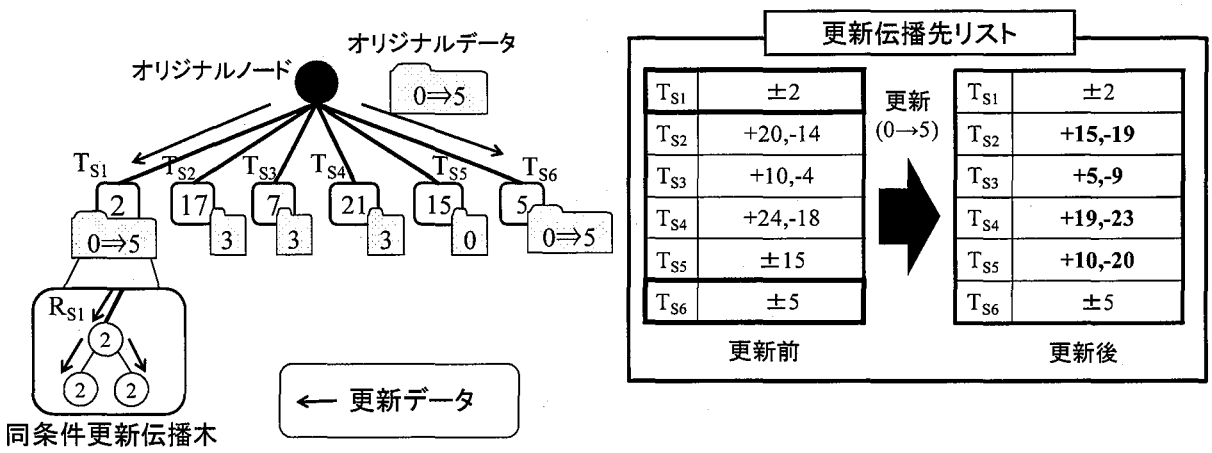


図 4.3: 更新データの伝播（UPDD-S 法）

4.3.3 ピアの参加

新たにデータにアクセスし，複製を作成した新規参加ピアは，自身のデータ要求条件を設定し，そのデータに関する更新伝播木に参加する．ピアが参加する際，新規参加ピアは，自身と等しいデータ要求条件を設定している同条件更新伝播木に参加する必要がある．UPDD-S 法では，オリジナルノードが全ての同条件更新伝播木の情報を管理しているため，最初にオリジナルノードに問い合わせることで，新規参加ピアの参加位置を発見できる．以下に，新規参加ピアの参加手順の詳細を説明する．

1. 新規参加ピアはまず，オリジナルノードに参加メッセージを送信する．参加メッセージを受け取ったオリジナルノードは，各同条件更新伝播木の情報を調べる．ここで，新規参加ピアと等しいデータ要求条件を設定している同条件更新伝播木が存在していない場合は，新規参加ピアを根ノードとする新たな同条件更新伝播木としてオリジナルノードに接続し，参加処理を終了する．新規参加ピアと等しいデータ要求条件を設定している同条件更新伝播木が存在する場合は，その同条件更新伝播木の根ノードに参加メッセージを送信し，手順 2. に進む．

2. 参加メッセージを受け取ったピアは，自身の子ノードの数を確認し，子ノードの数に空きがある場合は，新規参加ピアを自身の子ノードとして参加処理を終了する．子ノードの数に空きがない場合は，自身の子ノードの中からランダムに一つの子ノード

を選択し，参加メッセージを送信する．手順2.の操作を繰り返すことにより，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．

4.3.4 ピアの脱退

複製の置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．

脱退するピアは，2.4.2項で説明したように，UPT法におけるピアの脱退と同様の手順に従って，同条件更新伝播木から脱退する．

4.4 UPDD-SO法

UPDD-S法では，各ピアのデータ要求条件を考慮することで，更新データを必要としていないピアへの更新データの伝播を抑え，更新データの伝播にともなうトラヒックを削減できる．しかしUPDD-S法では，各ピアが設定する全てのデータ要求条件に対して個別に同条件更新伝播木を構築し，オリジナルノードが管理する必要がある．また，更新発生時には，オリジナルノードが直接同条件更新伝播木の根ノードに更新データを送信する．そのため，各ピアが様々なデータ要求条件を設定している環境では，多くの同条件更新伝播木を構築する必要があり，オリジナルノードの負荷が増大する．本節では，この問題を解決するために，UPDD-S法を拡張したUPDD-SO法を提案する．UPDD-SO法では，複製所持ピアを階層的に管理することで，オリジナルノードの負荷を他のピアに分散させる．

図4.4に，UPDD-SO法における更新伝播木の構成例を示す．UPDD-SO法では，UPDD-S法と同様に，同一のデータ要求条件を設定しているピアごとに同条件更新伝播木を構築する．さらに，オリジナルノードは，データ要求条件の値が近い同条件更新伝播木をまとめてG個のグループを作成し，それぞれの同条件更新伝播木の根ノードで構成される別の木構造（条件順序木）を構築する．この図では，データ要求条件を1から20，21から40，41から60と設定しているピアごとにグループを作成し，三つの条件順序木を構築している$(G = 3)$．この条件順序木は，データ要求条件の値が小さいピア（同条件更新伝播木の根ノード）の深さがより小さくなるn分木とする．

UPDD-SO法では，各ピアが設定し得るデータ要求条件の種類の数Dの値が増加するにつれ，各同条件更新伝播木に参加するピア数が減少する．その一方で，条件順序木に参加するピア数が増加するため，更新伝播時に発生するトラヒックや遅延が，UPDD-S法と大

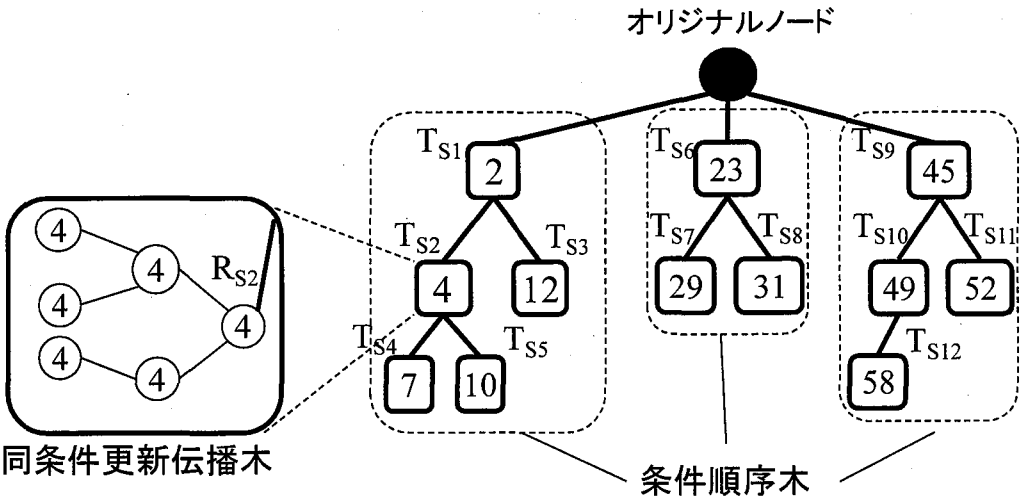


図 4.4: 更新伝播木の構成（UPDD-SO 法：$G = 3$）

きく異なってくることが考えられる．また，条件順序木の数 G の値が増加するにつれ，オリジナルノードが直接更新データを送信すべきピア数が増加するため，オリジナルノードおよび複製所持ピアの更新伝播時の負荷が変化することが考えられる．なお，本章では，$n = 2$（2 分木）として説明する．

4.4.1 ピアが管理する情報

UPDD-SO 法では，UPDD-S 法と同様，更新伝播木を維持するために，各複製所持ピアは更新伝播木上の周辺ピアの情報（例：IP アドレス）を管理する．まず，同条件更新伝播木を維持するために，同条件更新伝播木の根ノードは，同条件更新伝播木上の自身の子ノードの情報を管理し，根ノード以外の複製所持ピアは，4.3.1 項と同様に，同条件更新伝播木上の親ノードおよび子ノードの情報を管理する．さらに，オリジナルノードおよび各同条件更新伝播木の根ノードは条件順序木にも参加しているため，条件順序木を維持するための情報も管理する．条件順序木を維持するために必要な情報については，4.4.3 項で説明する．

また，UPDD-SO 法では，条件順序木上の各ピア（オリジナルノードおよび各同条件更新伝播木の根ノード）は，各同条件更新伝播木に属するピアが更新データを必要としている条件を更新伝播先リストとして管理する．図 4.5 に，更新データ伝播のために条件順序木上のピアが管理する情報を示す．条件順序木上の各ピアは，UPDD-S 法におけるオリジナル

ノードと同様に，条件順序木上の子ノードが次に更新データを受信する条件を更新伝播先リストとして管理する．ただし，自身の子孫ノードのうち，子ノードよりも次に更新データを受信する条件を表す値の絶対値が小さいノードが存在する場合には，その子孫ノードの情報も更新伝播先リストに加える．例えば，図4.5のピアR_{S1}は，条件順序木上における自身の子ノードである同条件更新伝播木T_{S2}および同条件更新伝播木T_{S3}に属するピアが次に更新データを受信する条件である± 4と± 12を，自身の更新伝播先リストに格納する．また，同条件更新伝播木T_{S3}の子ノードである同条件更新伝播木T_{S4}に属するピアは，所持している複製の値(0)とオリジナルデータの値(5)との差が5であり，オリジナルデータの値がさらに2以上増加した場合に更新データを必要としている．この値の絶対値(2)は，同条件更新伝播木T_{S2}に属するピアが次に更新データを受信する条件の絶対値(4)より小さいため，ピアR_{S1}は，同条件更新伝播木T_{S4}に属するピアが次に更新データを受信する条件(+2, −12)を自身の更新伝播先リストに追加する．この操作により，ピアR_{S1}は同条件更新伝播木T_{S4}に属するピアが次に更新データを受信する条件を知ることができるため，たとえ同条件更新伝播木T_{S2}に属するピアが更新データを必要としていなくても，同条件更新伝播木T_{S4}に属するピアが更新データを必要としている場合には，自身の子ノードへと更新データを伝播させることができる．

4.4.2 更新情報の伝播

更新発生時には，オリジナルノードは更新伝播先リストを参照し，その更新データを必要としているピアで構成される同条件更新伝播木が存在するかを調べる．該当する同条件更新伝播木が存在する場合，条件順序木に沿って，該当する同条件更新伝播木の根ノードまで更新データを伝播させる．以下では，図4.6を用いて，更新データの伝播時の動作について説明する．図4.6において，初期状態では，オリジナルノードおよび全複製所持ピアは，値が0である最新のデータ（複製）を所持しているものとする．

1. データを更新したオリジナルノードは，4.3.2項で説明したUPDD-S法におけるオリジナルノードの動作と同様に，自身の更新伝播先リストを参照し，更新データを必要としているピアが属する同条件更新伝播木が存在するかどうかを調べる．該当する同条件更新伝播木が存在する場合，更新データを送信する．同時に，オリジナルノードは，自身の更新伝播先リストの情報を更新する．

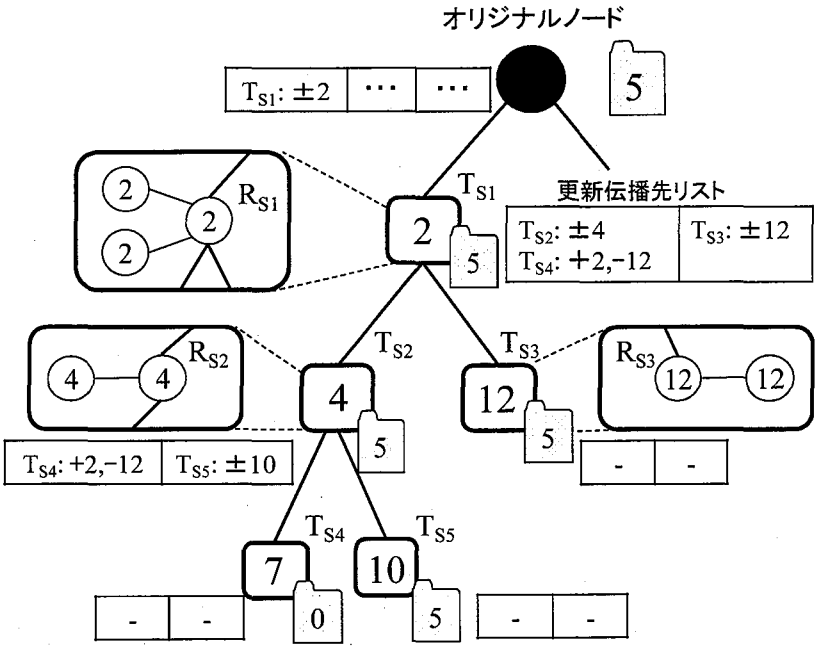


図 4.5: 各ピアが管理する情報（更新伝播用）

例えば図 4.6(a) において，データの値を 0 から 5 に更新 (+5) したオリジナルノードは，自身の更新伝播先リストの中で，次に更新データを受信する条件が ± 2 であるピア R_{S1} に更新データを送信する．この場合，次に同条件更新伝播木 T_{S1} に属するピアが更新データを受信する条件は変化しないため，オリジナルノードが管理する同条件更新伝播木 T_{S1} に属するピアの情報は更新しない．一方，次に更新データを受信する条件が ± 23 であるピア R_{S6} には更新データを送信せず，同条件更新伝播木 T_{S6} に属するピアが次に更新データを受信する条件の値を 5 だけ減少させる (+18, −28).

2. 更新データを受信したピアは，自身が所持する複製を更新し，自身が根ノードとなっている同条件更新伝播木に沿って更新データを伝播させる．同時に，自身の更新伝播先リストを参照し，更新データを受信する条件を満たす子孫ノードが存在する場合，該当する子ノード（自身の子孫ノードのみが更新データを必要としている場合は，その子孫ノードの先祖ノードとなっている子ノード）に更新データを送信する．その後，オリジナルノードと同様に，自身が管理する更新伝播先リスト内の情報を更新する．自身の更新伝播先リスト内に，更新データを受信する条件を満たす子孫ノードが存在

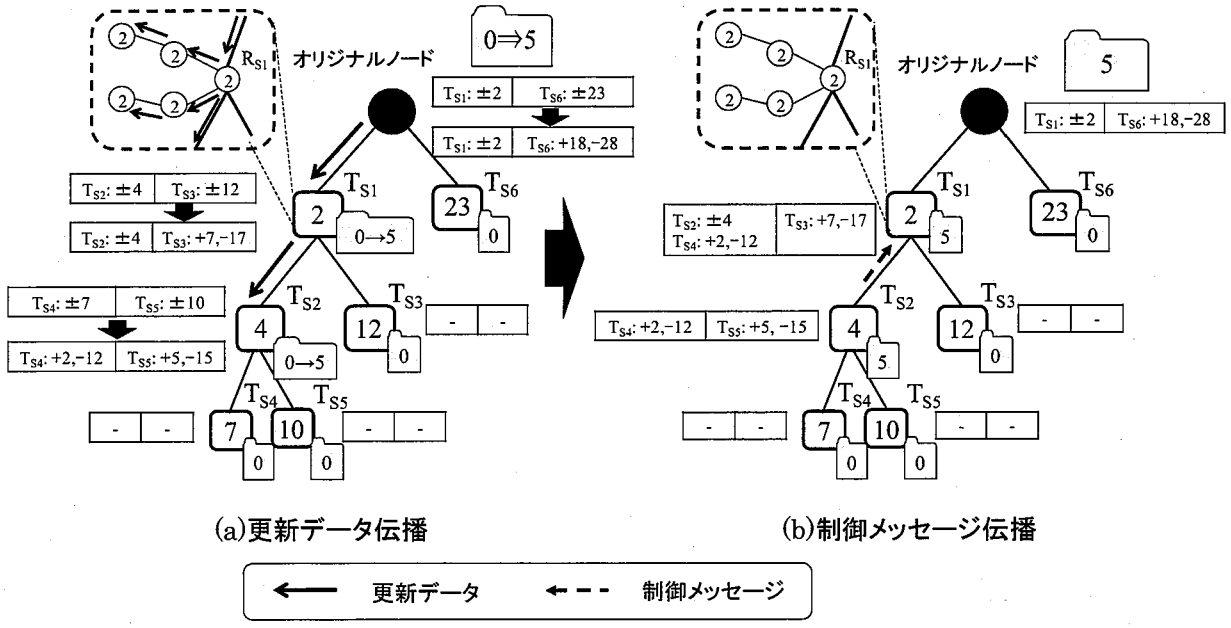


図 4.6: 更新データの伝播（UPDD-SO 法）

する限り，手順 2. の処理を繰り返す．

例えば，図 4.6(a) のピア R_{S1} は，自身の複製の値を 0 から 5 に更新し，同条件更新伝播木に沿って更新データを伝播させる．その後，自身の更新伝播先リストを参照し，値の絶対値がデータの更新量 (5) より小さいピア R_{S2}(± 4) に更新データを送信する．また，ピア R_{S3} は条件の値 (± 12) の絶対値が更新量よりも大きいため，ピア R_{S3} には更新データを送信せず，ピア R_{S1} が管理する更新伝播先リスト内のピア R_{S3} の条件の値を 5 減少させる (+7, −17).

3. 手順 2. において，次に更新データを受信する条件を満たす子孫ノードが存在しない場合，更新データの伝播を停止し，自身と自身の子ノードが次に更新データを受信する条件を比較する．子ノードの条件の方が最小の絶対値が小さい場合，その子ノードの条件を親ノードに送信する．UPDD-SO 法では，このメッセージを制御メッセージと呼ぶ．

   例えば，図 4.6(a) のピア R_{S2} は，更新伝播先リスト内のピア R_{S4} とピア R_{S5} がこの時点では更新データを必要としていないため，更新データの伝播を停止し，自身の更新伝播先リストを更新する．その後，自身が次に更新データを受信する条件 (± 4) と，

子ノードが次に更新データを受信する条件 (+2, −12) および (+5, −15) を比較し，ピア R_{S2} 自身の条件よりも最小の絶対値が小さい同条件更新伝播木 T_{S4} に属するピアの情報を，制御メッセージとしてピア R_{S1} に通知する（図 4.6(b)）.

4. 制御メッセージを受信したピアは，メッセージに含まれる同条件更新伝播木の情報を，自身の更新伝播先リストに追加する．その後，自身と更新伝播先リストに含まれる全ての子孫ノードの条件を比較し，その最小の絶対値が自身の条件よりも小さい子孫ノードが存在する場合，その子孫ノードの情報を含む制御メッセージを親ノードに送信する．該当する子孫ノードが存在する限り以上の操作を繰り返し，該当する子孫ノードが存在しない場合は，制御メッセージの伝播を停止する．

   例えば，図 4.6(b) のピア R_{S1} は，ピア R_{S2} から送られてきた制御メッセージに含まれる，同条件更新伝播木 T_{S4} に属するピアが次に更新データを受信する条件 (+2, −12) を追加する．この時点で，ピア R_{S1} の更新伝播先リスト内に含まれる条件の中で，ピア R_{S1} の条件 (± 2) よりも最小の絶対値が小さい条件が存在しないため，ここで制御メッセージの伝播を停止する．

手順 3. のように，自身の子ノードの情報を親ノードに伝えることは，更新データを必要としているピアに確実に更新データを伝播させるために必要な操作である．もし，条件順序木上の各ノードが自身の子ノードの情報しか管理していない場合，その子ノードが更新データを受信する条件を調べ，条件を満たしている場合にのみその子ノードに更新データを送信する．そのため，自身の子ノードは更新データを必要としてないが，さらに下位のノードが更新データを必要としている場合には，その下位のノードに更新データを伝播させることができない．そのため，必要に応じて，自身の子孫ノードの情報を管理する必要がある．

4.4.3 木構造の維持

UPDD-SO 法では，ピアがデータの複製を配置，削除した場合，更新伝播木への参加や脱退が発生するため，更新伝播木を維持する必要がある．図 4.7 に，条件順序木を維持するために，条件順序木上の各ピアが管理する情報を示す．各ピアは，条件順序木上の親ノードと子ノードの情報（例：IP アドレス）に加え，各子ノードの子孫ノードのデータ要求条件を，子ノードごとに管理する．例えば，ピア R_{S1} は，第一子ノード R_{S2} およびその子孫

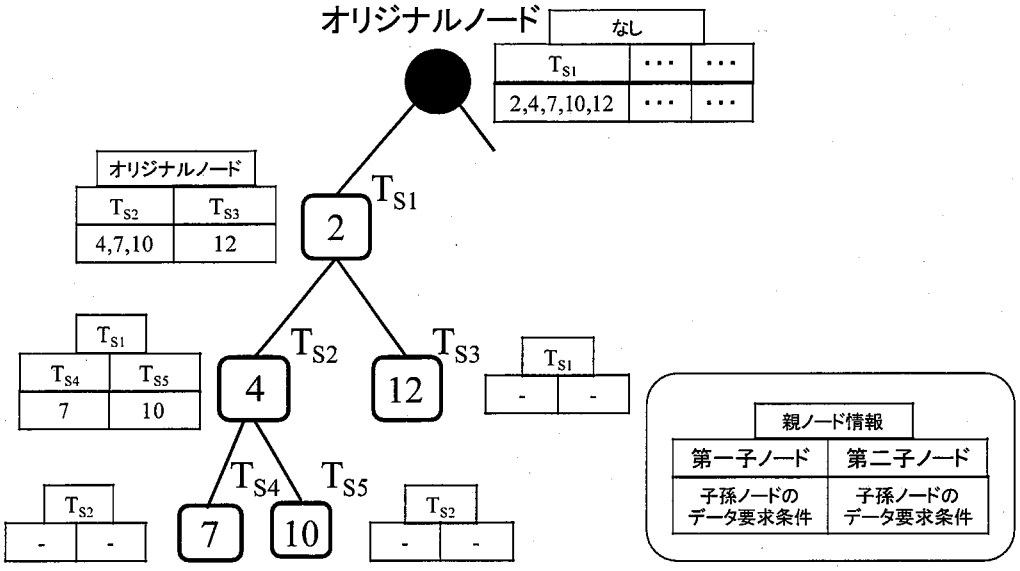


図 4.7: ピアが管理する情報（木構造維持用）

ノード R_{S4}, R_{S5} のデータ要求条件として 4, 7, 10 を，第二子ノード R_{S3} のデータ要求条件として 12 を管理する．また，同条件更新伝播木に関しては，4.3.3 項および 4.3.4 項と同様の手順で構築や維持を行うものとする．

ピアの参加

新たにデータにアクセスし複製を作成した新規参加ピアは，条件順序木および同条件更新伝播木に参加する．ピアが参加する際，新規参加ピアは，自身と等しいデータ要求条件を設定している同条件更新伝播木に参加する必要がある．ここで，条件順序木上では，データ要求条件の値が大きいピアほど木の深い位置に参加している．そのため，条件順序木に沿って参加メッセージを伝播させることで，条件順序木上における新規参加ピアが参加するべき場所を決定する．その後，新規参加ピアと等しいデータ要求条件を設定している同条件更新伝播木が存在している場合，その同条件更新伝播木に参加する．以下に，新規参加ピアの参加手順の詳細を図 4.8 を用いて説明する．

1. 新規参加ピアは，まずオリジナルノードに参加メッセージを送信する．参加メッセージを受け取ったオリジナルノードは，自身が管理する子孫ノードのデータ要求条件を調べる．新規参加ピアと等しいデータ要求条件を設定しているピアがすでに存在して

いれば，そのピアが参加している条件順序木に属する自身の子ノードへ参加メッセージを送信し，手順2.に進む．新規参加ピアと等しいデータ要求条件を設定しているピアが存在していない場合，新規参加ピアのデータ要求条件に応じて，新規参加ピアが参加すべき条件順序木に属する自身の子ノードへ参加メッセージを送信する．その後，参加メッセージを送信した子ノード以下の子孫ノードのデータ要求条件として，新規参加ピアのデータ要求条件を追加し，手順2.に進む．

例えば，図4.8において，データ要求条件を8に設定したピアが新たに参加する場合を想定する．新規参加ピアから参加メッセージを受け取ったオリジナルノードは，自身が管理する子孫ノードの情報を調べる．ここで，データ要求条件を8に設定しているピアは他に存在していないが，第一子ノード以下の子孫ノードのデータ要求条件の範囲は2から12であり，データ要求条件が8のピアも第一子ノードが属する条件順序木に分類される．従って，オリジナルノードは第一子ノード T_{S1} の根ノードであるピア R_{S1} に参加メッセージを送信し，T_{S1} 以下の子孫ノードのデータ要求条件に8を追加している．

2. 参加メッセージを受け取ったピアは，自身と新規参加ピアのデータ要求条件を比較する．自身と新規参加ピアのデータ要求条件が等しい場合は，自身を根ノードとする同条件更新伝播木に新規参加ピアを参加させる．具体的には，参加メッセージをその同条件更新伝播木に沿って葉ノードまで伝播させ，新規参加ピアをその葉ノードの子ノードとして参加させ，参加処理を終了する．自身と新規参加ピアのデータ要求条件が異なる場合，手順3.に進む．

3. 参加メッセージを受け取ったピアは，自身の子孫ノードのデータ要求条件を参照し，新規参加ピアと等しいデータ要求条件を設定しているピアが存在するかを調べる．存在する場合，その子孫ノードが参加している部分木に属する自身の子ノードへ参加メッセージを送信し，手順2.に戻る．等しいデータ要求条件を設定しているピアが存在しない場合は手順4.に進む．

4. 参加メッセージを受け取ったピアは，自身と新規参加ピアのデータ要求条件を比較し，新規参加ピアのデータ要求条件の方が大きい場合，条件順序木上の自身の子ノードの数に空きがあれば，そこに新規参加ピアを参加させる．その後，自身の子ノードのデータ要求条件として，新規参加ピアのデータ要求条件を追加し，参加処理を終了

する．子ノードの数に空きがなければ，自身の子ノードの中からランダムに一つを選択して参加メッセージを送信する．その後，参加メッセージを送信した子ノード以下の子孫ノードのデータ要求条件として，新規参加ピアのデータ要求条件を追加し，手順2.に戻る．新規参加ピアのデータ要求条件の方が小さい場合は手順5.に進む．

例えば，図4.8では，参加メッセージを受け取ったピアR_{S1}のデータ要求条件(2)よりも，新規参加ピアのデータ要求条件(8)の方が大きく，ピアR_{S1}の子ノードの数に空きがない．そのため，ランダムに選択したピアR_{S3}に参加メッセージを送信し，T_{S3}以下の子孫ノードのデータ要求条件に8を追加している．

5. 参加メッセージを受け取ったピアは，自身が参加していた位置に新規参加ピアを参加させ，自身が管理していた情報を新規参加ピアに渡す．その後，自身が新たな新規参加ピアとなり，自身と位置を交換した元新規参加ピアに参加メッセージを送信し，手順2.に戻る．

   例えば，図4.8では，参加メッセージを受け取ったピアR_{S3}のデータ要求条件(12)よりも，新規参加ピアのデータ要求条件(8)の方が小さいため，新規参加ピアはピアR_{S3}と位置を入れ替える．その後，新たに新規参加ピアとなったピアR_{S3}はピアR_{S6}に参加メッセージを送信し，手順4.に従って，ピアR_{S6}の子ノードとして更新伝播木に再参加する．

ピアの脱退

複製の置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．UPDD-SO法では，UPDD-S法と同様に，脱退ピアは，自身が参加している同条件更新伝播木から脱退する．ここで，脱退ピアと等しいデータ要求条件を設定しているピアが他に存在しない場合，その同条件更新伝播木が削除されるため，条件順序木が分断されてしまう．そのため，必要に応じて，条件順序木も修復する必要がある．条件順序木を修復する際には，データ要求条件が小さいピアの深さがより小さくなるという条件順序木の性質を維持する必要がある．そのため，最初はUPT法と同様に，条件順序木上の葉ノードを用いて脱退箇所を修復する．その後，脱退箇所に移動したピアとその子ノードのデータ要求条件を比較し，必要に応じて位置を入れ替えることにより，条件順序木の性質を維持する．ピアの脱退手順の詳細は以下の通りである．

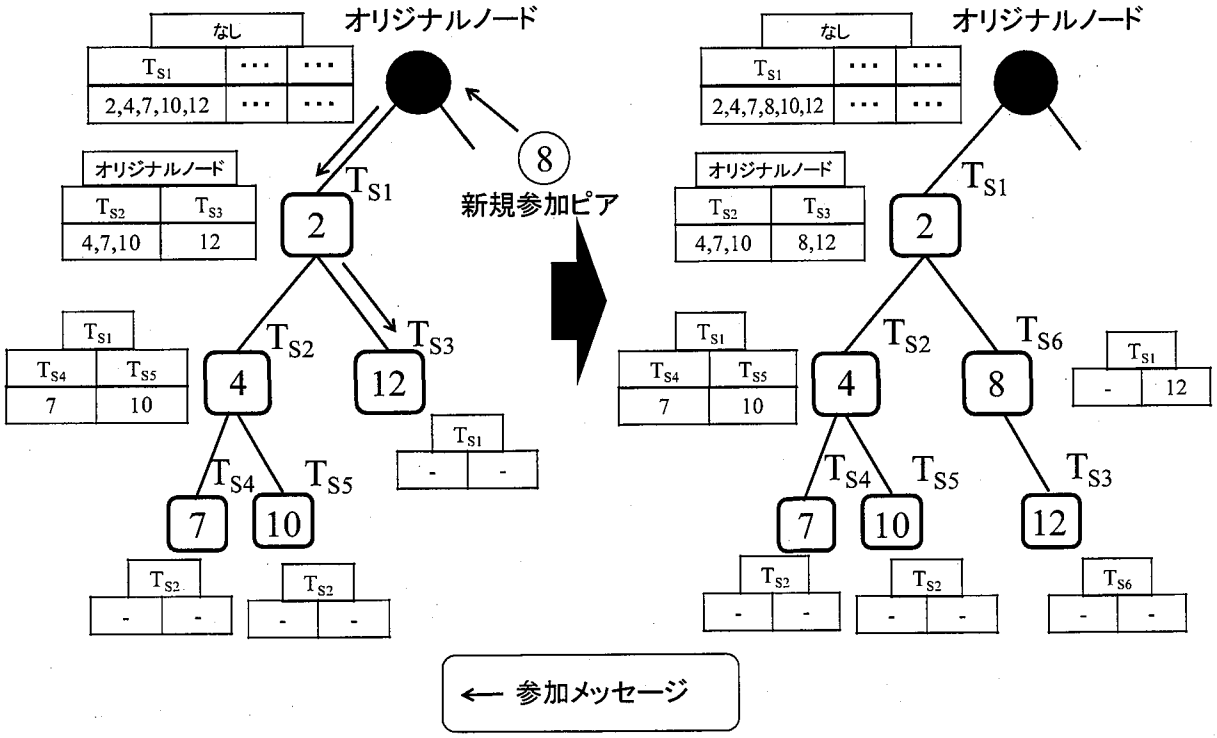


図 4.8: ピアの参加（UPDD-SO 法）

1. 脱退ピアが属する同条件更新伝播木が複数のピアで構成されている，つまり，脱退ピアとデータ要求条件が等しいピアが他に存在する場合，脱退ピアは，その同条件更新伝播木からの脱退のみを行う．具体的には，脱退ピアは，自身が属する同条件更新伝播木上の子ノードの一つをランダムに選択し，脱退メッセージを送信する．この操作を，脱退メッセージが同条件更新伝播木上の葉ノードに到達するまで繰り返す．その後，その葉ノードと位置を入れ替え，脱退処理を終了する．

2. 脱退ピアが属する同条件更新伝播木がその脱退ピアのみで構成されている，つまり，脱退ピアとデータ要求条件が等しいピアが他に存在しない場合，脱退ピアは，条件順序木から脱退する．具体的には，条件順序木上の子ノードの一つをランダムに選択し，脱退メッセージを送信する．この操作を，脱退メッセージが条件順序木上の葉ノードに到達するまで繰り返す．その後，その葉ノードと位置を入れ替えることにより条件順序木から脱退し，手順 3. に進む．

3. データ要求条件が小さいピアの深さがより小さくなるという条件順序木の特徴を維

持するために，脱退ピアと位置を入れ替えたピアは，自身と条件順序木上の新たな子ノードのデータ要求条件を比較する．ここで，子ノードの方がデータ要求条件が小さい場合，その子ノード（該当する子ノードが二つ以上存在する場合はデータ要求条件が最も小さいもの）と位置を入れ替える．自身よりもデータ要求条件が小さな子ノードが存在している限りこの操作を繰り返し，該当する子ノードが存在しない場合に脱退処理を終了する．

4.5 性能評価

本節では，UPDD-S 法および UPDD-SO 法において，各ピアが設定しているデータ要求条件の種類 D やオリジナルノードが管理する条件順序木の数 G の値が変化した場合でも，更新伝播時に発生するトラヒックや遅延，および各ピアの負荷の面で有効であるか，およびこれらの値はどのように変化するかを調査する．以下では，UPDD-S 法および UPDD-SO 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．

4.5.1 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，それらがべき法則に従ってネットワークを構成するものとした．各ピアの隣接ピア数は式 (3.1) を用いて決定した．

データの種類を 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 タイムスロット毎に 0.01 の確率であるデータを要求する．データ要求の発行は，Zipf 係数 0.5 の Zipf 分布に従うものとし，式 (2.2) を用いて決定した．

複製の配置方式には，オーナー複製法を用いた．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，データのオリジナルを所持するピアは，オリジナルの他に 10 個の複製を所持できるものとした．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．

データの更新量が小さい場合，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，全体的にはデータの値が増加している場合を網羅することを考慮し，オリジナル

ノードは，初期値が100の数値データを所持し，シミュレーション時間全体で1000回の更新発生によって，図4.9で示されるようにデータの値を更新するものとした．また，各ピアが大小様々なデータ要求条件を設定している環境として，各ピアが設定しているデータ要求条件の値の最大値をD_{max}とし，各ピアはD種類のデータ要求条件($D_{max}/D \times 1$,$D_{max}/D \times 2$,$\cdots$,$D_{max}/D \times D$)の中からランダムに一つを選択するものとした．このD_{max}の値は，一度の更新によるデータの更新量に対して十分に大きな値である100に設定した．ただし，D_{max}の変化によるトラヒックへの影響を調べるため，4.5.3項でのみ20から100まで変化させた．オリジナルノードは，更新伝播時の自身の負荷が大きくなりすぎないように，データ要求条件の大きさに応じて同条件更新伝播木を五つのグループに分け，それぞれで条件順序木を構築した($G = 5$)．例えば，$D_{max} = 100$, $G = 5$の場合，データ要求条件が[1,20],[21,40],[41,60],[61,80],[81,100]のピアでそれぞれ条件順序木を構築した．ただし，4.5.2項の更新伝播負荷の評価では，Gの値による影響を調べるために，Gの値を大きく設定した$G = 20$の場合の評価も行っている．また，同条件更新伝播木および条件順序木はともに2分木とした．

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い，データ要求条件を考慮せず木構造を用いて常に全ピアに更新データを伝播させるUPT法と，UPDD-S法およびUPDD-SO法における性能を評価した．UPT法に関しては，UPDD-SO法で構築される木構造の形状に近づけるために，オリジナルノードのみG個の子ノードをもち，それ以下の部分木は2分木となる更新伝播木を構築した．ピアの参加，脱退時には，UPDD-S法およびUPDD-SO法における同条件更新伝播木からの参加，脱退と同様の動作を行うものとした．なお，シミュレーション実験では，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調査した．ただし，他のデータを対象とした場合も，データ番号1のデータと同様の特徴をもつ結果が得られると考えられる．他のデータを対象とした場合の結果に関する考察は4.5.4項で行う．

4.5.2 データ要求条件数の影響

最初に，各ピアが設定するデータ要求条件の数Dの値を変化させた場合の，以下の値を評価した．

- ネットワークトラヒック

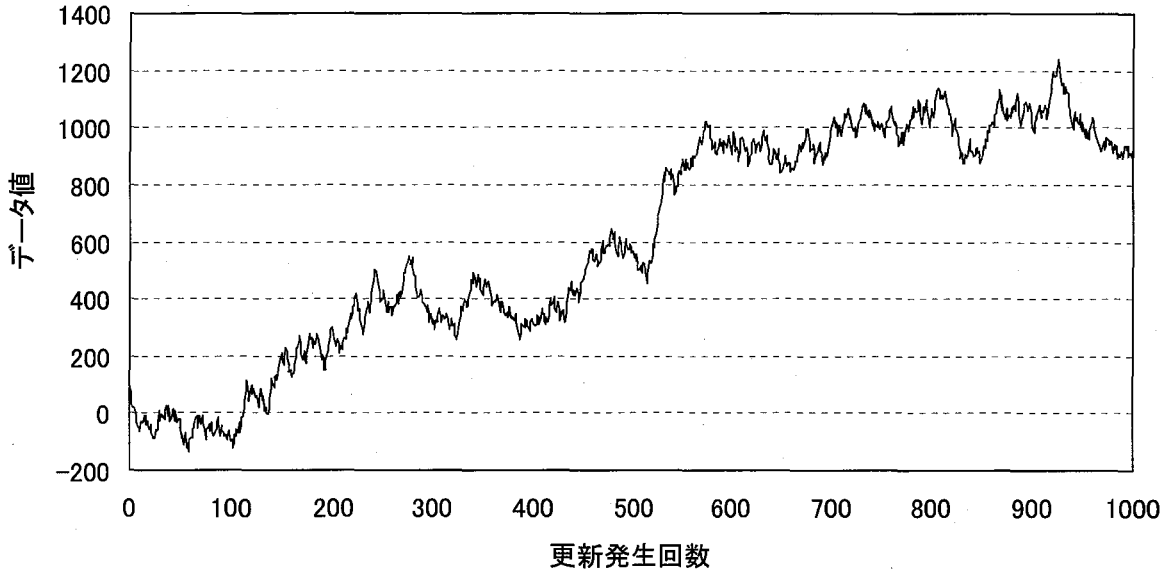


図 4.9: データの更新度合い

更新データを伝播させることによるトラヒック，および，木構造維持のためにピア間で交換されるメッセージ（木構造維持メッセージ）や制御メッセージを伝播させることによるトラヒックの合計．ただし，更新データのサイズを 10，二つのメッセージのサイズを 1 とし，伝播されるデータの総量をトラヒックとした．

- 平均ホップ数

  オリジナルノードから各複製所持ピアへ更新データが伝播される際に要した論理ホップ数の平均値

- 最大ホップ数

  オリジナルノードから各複製所持ピアへ更新データが伝播される際に要した論理ホップ数の最大値

- オリジナルノードの更新伝播負荷

  オリジナルノードが他のピアに更新データを送信した回数の合計（シミュレーション時間）

- オリジナルノード以外のピアの更新伝播負荷

  オリジナルノード以外の各複製所持ピアが他のピアに更新データを送信した回数の合計（シミュレーション時間全体）の平均値

ネットワークトラヒック

ネットワークトラヒックの結果を図4.10に示す．図4.10において，縦軸はネットワークトラヒックを表す．結果より，UPT法と比較して，UPDD-S法およびUPDD-SO法では，更新データを必要としていないピアへの更新データの伝播を抑えることで，更新データの伝播によるトラヒックを約70%削減できていることがわかる．ここで，UPDD-S法とUPDD-SO法を比較した場合，UPDD-SO法におけるネットワークトラヒックがわずかに増加している．これは，UPDD-SO法における更新データの伝播方法や木構造の維持方法に起因する．UPDD-SO法では，更新データが条件順序木に沿って伝播される．そのため，条件順序木の葉ノードのみが更新データを必要としている場合でも，オリジナルノードとその葉ノードの間に位置するノードは，たとえその時点で更新データを必要としていなくても更新データを受信する必要がある．したがって，更新データの伝播によるトラヒックが増加している．また，UPDD-SO法では，同条件更新伝播木だけでなく条件順序木も構築，維持する必要があるため，木構造維持メッセージによるトラヒックが大きくなる．さらに，更新データを必要としているピアに確実に更新データを伝播させるために，ピア間で制御メッセージを送受信する必要がある．以上の理由から，UPDD-S法に比べてUPDD-SO法におけるネットワークトラヒックが大きくなっている．

また，各ピアが設定しているデータ要求条件の数Dが増加した場合でも，UPT法に比べ，UPDD-S法およびUPDD-SO法におけるネットワークトラヒックは小さく抑えられている．しかし，UPT法およびUPDD-S法では，Dの値が増加してもネットワークトラヒックがほとんど変化していない一方で，UPDD-SO法におけるネットワークトラヒックは増加している．UPT法およびUPDD-S法では，更新データを必要としているピアにのみ更新データを伝播させるため，データ要求条件の数が変化しても，更新データを伝播させるべきピア数はあまり変化しない．一方，UPDD-SO法では，Dの値が増加するにつれて条件順序木上のノード数が増加し，条件順序木が高くなる．そのため，上述したように，更新データを葉ノードに伝播させるために更新データを中継するノード数が多くなり，余分な更新データ伝播によるトラヒックが増加する．

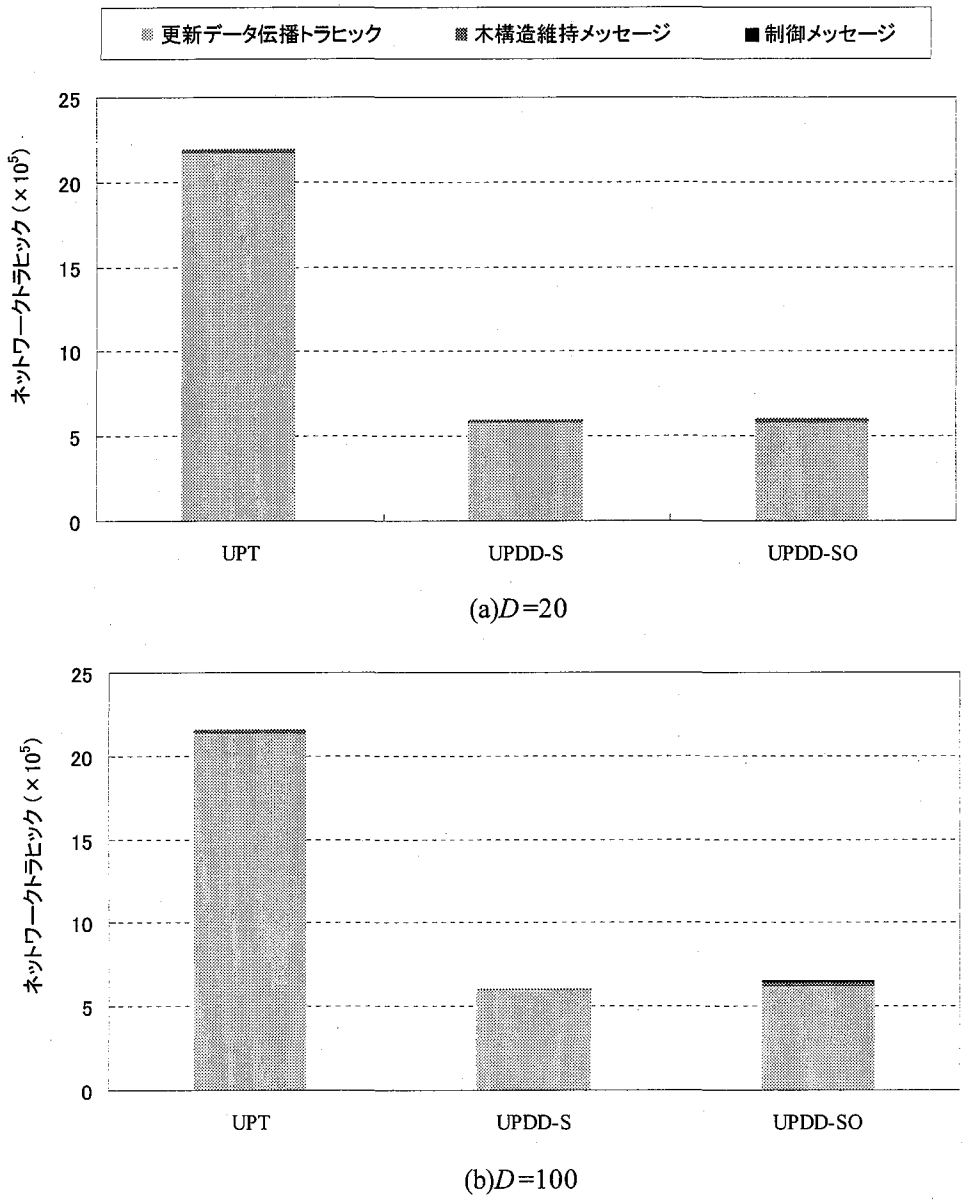


図 4.10: ネットワークトラヒック

更新伝播時の平均ホップ数，最大ホップ数

平均ホップ数，最大ホップ数の結果を図 4.11 および図 4.12 に示す．これらの図において，横軸はデータ要求条件の種類 D を表し，縦軸はそれぞれ平均ホップ数および最大ホップ数を表す．結果より，D の値によらず，全ての複製所持ピアに更新データを伝播させる UPT

法に比べ，更新データを伝播させるピア数を制限するUPDD-S法およびUPDD-SO法では，平均ホップ数，最大ホップ数ともに小さく抑えられている．ここで，UPDD-S法におけるホップ数は非常に小さく抑えられている一方で，UPDD-SO法におけるホップ数の減少量は小さい．これは，UPDD-SO法で階層的に複製所持ピアを管理していることに起因する．UPDD-SO法では，更新データ伝播時に，最初に条件順序木に沿って各同条件更新伝播木の根ノードに更新データを伝播させ，その後，同条件更新伝播木に沿って更新データを伝播させる．そのため，木構造上でのオリジナルノードから各ノードまでのホップ数が大きくなりやすい．また，条件順序木の葉ノードが更新データを必要としている場合，更新データを条件順序木上の葉ノードまで伝播させた後，同条件更新伝播木の葉ノードまで伝播させる必要があり，特に最大ホップ数が大きくなりやすい．

また，UPDD-S法では，各ピアが設定しているデータ要求条件の種類Dの値が大きくなるにつれて，平均ホップ数，最大ホップ数ともに小さくなっている．UPDD-S法では，ピアが様々なデータ要求条件を設定している場合，オリジナルノードがそれぞれのデータ要求条件ごとに同条件更新伝播木を作成する．そのため，一つの同条件更新伝播木に参加するピア数が少なくなりやすい．したがって，同条件更新伝播木上の各ピアまでのホップ数が小さくなる．

ただし，2.5.2項でも述べたように，論理ネットワーク上でのホップ数が小さい場合に，必ずしも物理ネットワーク上での遅延も小さくなるとは限らないため，物理ネットワークを考慮したさらなる検証が必要である．

更新伝播負荷

更新伝播負荷の結果を図4.13に示す．図4.13において，横軸はデータ要求条件の種類Dを表し，縦軸は更新伝播負荷を表す．ここで，更新発生時に常に全ての複製所持ピアに更新データを伝播させるUPT法では，他の手法と比べて更新伝播負荷が極端に大きくなるため，結果は省略している．結果より，UPDD-S法では，オリジナルノードがすべての同条件更新伝播木の情報を管理し，直接更新データを送信する必要があるため，データ要求条件の数Dが増加するにつれて，オリジナルノードの負荷も大きく増加している．一方，UPDD-SO法では，条件順序木を構築することで，異なるデータ要求条件を設定しているピア同士で更新データの送受信を行うことができる．そのため，他の複製所持ピアの更新伝播負荷を小さく抑えつつも，オリジナルノードの負荷を約75% ～85% 軽減できる．また，

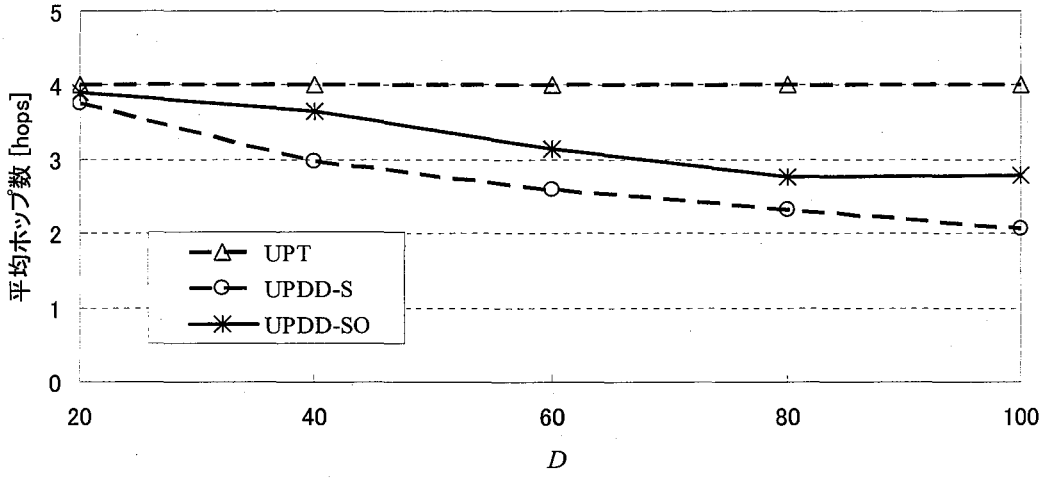


図 4.11: 平均ホップ数

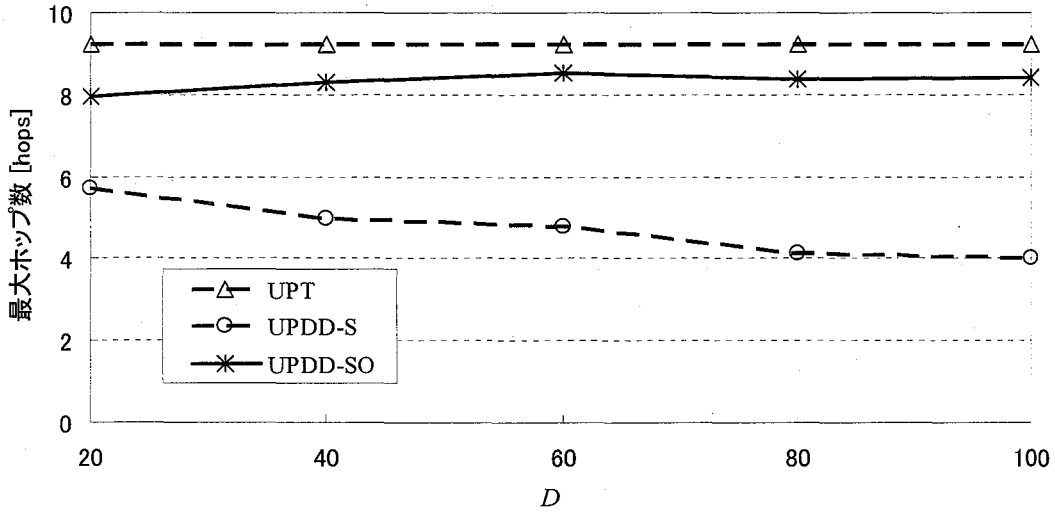
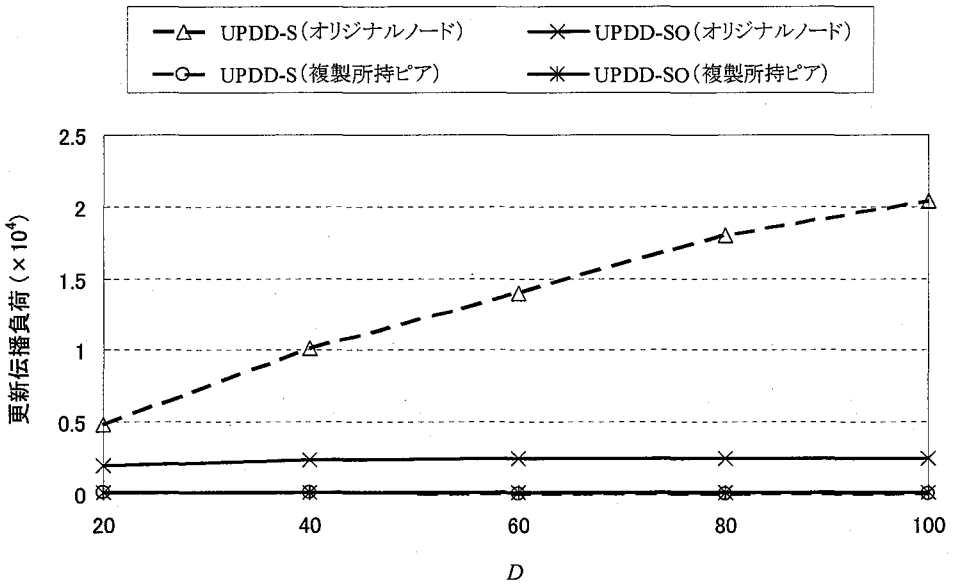
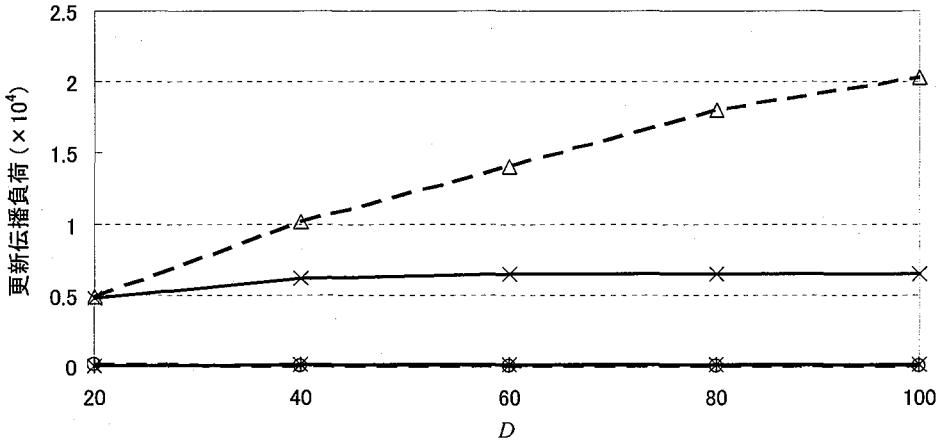


図 4.12: 最大ホップ数

UPDD-SO 法では，条件順序木の数 G の値が増加した場合に，オリジナルノードの負荷が増加している．これは，G の値が大きくなるにつれ，オリジナルノードが直接更新データを送信するピア数が増加するためである．ただし，UPDD-SO 法におけるオリジナルノー



(a)G=5



(b)G=20

図 4.13: 更新伝播負荷

ドの負荷の増加量は，UPDD-S 法におけるオリジナルノードの負荷に比べると非常に小さい．この結果から，各ピアが設定するデータ要求条件が増加しても，オリジナルノードの負荷を十分に他の複製所持ピアに分散できることがわかる．

ここで，各ピアの負荷としては，更新伝播負荷だけでなく，ピアが条件順序木上を移動する際に，自身と周辺ピアのデータの差分計算のための負荷も発生する．しかし，UPDD-SO

法において，条件順序木上のノードの移動は，新たなデータ要求条件を設定しているピアが参加する場合や，一つの同条件更新伝播木に参加するピアが全ていなくなった場合などに限られ，頻繁には発生しない．したがって，ノードの移動によって発生する各ピアの負荷は，更新伝播負荷に比べて，無視できるほど十分に小さいといえる．

4.5.3 更新データ伝播トラヒック

次に，各ピアが設定しているデータ要求条件の最大値 D_{max} の値を変化させた場合の，以下の値を評価した．

- 更新データ伝播トラヒック

  更新データを伝播させることによるトラヒック．ただし，更新データのサイズを 10 とし，伝播されるデータの総量をトラヒックとした．

更新データ伝播トラヒックの結果を図 4.14 に示す．図 4.14 において，横軸は D_{max} を表し，縦軸は更新データ伝播トラヒックを表す．結果より，D_{max} の値が大きくなるにつれ，UPDD-S 法，UPDD-SO 法ともに，更新データ伝播トラヒックが小さくなっている．この結果から，各ピアが更新データを必要としている条件に応じて適切に更新データを伝播させる本章の提案手法は，各ピアが類似したデータ要求条件を設定している環境よりも，各ピアが幅広い範囲で自由にデータ要求条件の値を設定できる環境で，より有効性が高まることがわかる．

また，更新データ伝播トラヒックの値は，UPDD-S 法に比べて UPDD-SO 法の方が大きくなっている一方で，D_{max} が大きくなるにつれ，その差が大きくなっていることがわかる．これは，D_{max} の値が小さい場合，UPDD-SO 法において，一つの条件順序木に参加しているピアが設定しているデータ要求条件の値が近くなり，それらのピアは同時に更新データを必要とする可能性が高くなる．一方，D_{max} の値が大きい場合，一つの条件順序木に参加しているピアが設定しているデータ要求条件の値の範囲が広くなり，それぞれのピアが異なるタイミングで更新データを必要とする可能性が高くなる．この場合，4.5.2 項のネットワークトラヒックの評価で述べたように，不必要な更新データの伝播が発生する回数が多くなる．したがって，D_{max} の値を大きくすることにより，二つの手法間での更新データ伝播トラヒックの差が大きくなっている．

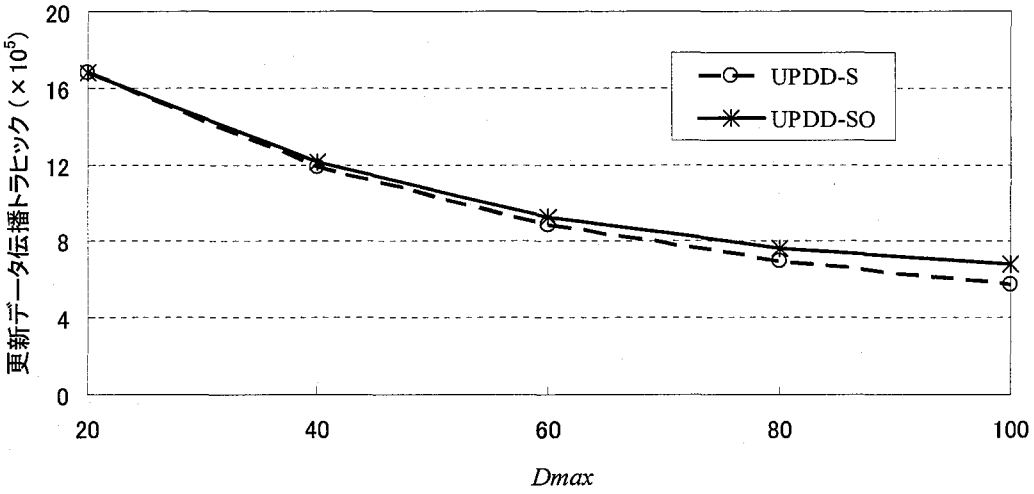


図 4.14: 更新データ伝播トラヒック

4.5.4 異なるアクセス頻度のデータを対象とした場合の考察

本節におけるシミュレーション実験では，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，このデータに関する評価を行った．アクセス頻度が高いデータを対象とした場合，そのデータの複製が作成される機会が多く，更新伝播木に参加するピア数が増加する．この場合，木構造維持メッセージや4.5.2項で説明した余分な更新データの伝播が発生する可能性が高くなり，提案手法の性能が悪くなることが考えられる．

しかし，このように提案手法に不利な環境においても提案手法の有効性が示されていることから，アクセス頻度が低いデータを対象とした場合においても，提案手法は有効であるといえる．

4.6 むすび

本章では，データの更新量を考慮した複製更新伝播法である，UPDD-S法およびUPDD-SO法を提案した．UPDD-S法では，等しいデータ要求条件を設定しているピアからなる木構造を構築し，オリジナルノードがそれらの木構造の情報を管理する．これにより，更新発生時にオリジナルノードは，更新データを必要としているピアにのみ更新データを伝

播させることができる．また，UPDD-S法におけるオリジナルノードの負荷を軽減するために提案したUPDD-SO法では，等しいデータ要求条件を設定しているピアからなる木構造の根ノードのみで構成されるもう一つの木構造を構築し，複製所持ピアを階層的に管理する．これにより，異なるデータ要求条件を設定しているピア同士で更新データの伝播を行うことができ，オリジナルノードの負荷を他のピアに分散させる．

UPDD-S法およびUPDD-SO法の性能を評価するためにシミュレーション実験を行った．その結果から，UPDD-S法およびUPDD-SO法では，更新伝播時に発生するネットワーク全体のトラヒックを削減できることを確認した．また，同時に，更新データ伝播時の遅延も軽減できていることを確認した．次に，UPDD-S法およびUPDD-SO法における，オリジナルノードおよび他の複製所持ピアの更新伝播時の負荷を調査した．その結果，UPDD-SO法では，UPDD-S法に比べて，他の複製所持ピアの負荷を小さく抑えつつも，オリジナルノードの負荷を軽減できていることを確認した．最後に，各ピアが設定するデータ要求条件の値の大きさによる更新データ伝播時のトラヒックへの影響を調べた．その結果，UPDD-S法およびUPDD-SO法は，各ピアが幅広い範囲で自由にデータ要求条件の値を設定できる環境において，より有効性が高まることを確認した．

第5章

結論

5.1 本論文のまとめ

本論文では，P2P ネットワークにおける複製更新伝播法について議論した．

まず，第1章では，P2P ネットワークサービスの種類やその分類について明記し，複製を配置することや効率的に更新データを伝播させる手法の必要性について述べた．

第2章では，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，木構造に基づく複製更新伝播法である UPT 法を提案した．UPT 法では，各ピアが n 分木の論理ネットワーク（更新伝播木）を構築し，この更新伝播木に沿って更新データを伝播させることにより，更新伝播時の負荷分散と遅延減少を実現する．UPT 法の性能を評価するため，シミュレーション実験を行った．その結果から UPT 法では，ピア数の増加に対して更新伝播時の負荷を定数オーダに抑えると同時に，遅延を対数オーダに抑えることを確認した．

第3章では，UPT 法を拡張し，各ピアのデータアクセス頻度を考慮することにより，更新データ伝播時に発生するトラヒックや遅延を抑制する複製更新伝播法である UPT-HL 法を提案した．UPT-HL 法では，データアクセス頻度に応じて，更新データもしくはサイズの小さな無効化情報のどちらを伝播させるかを決定する．UPT-HL 法の性能を評価するため，シミュレーション実験を行った．その結果から UPT-HL 法では，木構造維持のためのトラヒックが増加する一方で，更新データ伝播によるトラヒックを大幅に削減することができ，ネットワーク全体のトラヒックを削減できることを確認した．さらに，無効化情報を受信したピアも L ピアとして H ピアの情報を保持しておくことにより，データ検索時に

必要なメッセージ数を抑えられることを確認した.

第4章では,データの更新量を考慮した複製更新伝播法である,UPDD-S法およびUPDD-SO法を提案した.UPDD-S法では,更新データを受信する条件が等しいピアからなる木構造を構築し,オリジナルノードがそれらの木構造に直接更新データを送信する.また,UPDD-S法を拡張したUPDD-SO法では,データ要求条件が等しいピアからなる木構造をさらに階層的に管理することで,異なるデータ要求条件を設定しているピア同士で更新データの送受信を行う.UPDD-S法およびUPDD-SO法の性能を評価するため,シミュレーション実験を行った.その結果からUPDD-S法およびUPDD-SO法では,更新伝播時に発生するネットワーク全体のトラヒックを削減できることを確認した.さらに,UPDD-SO法では,他の複製所持ピアの更新データ伝播時の負荷を小さく抑えつつも,オリジナルノードの負荷を軽減できることを確認した.

ネットワーク上で共有されるデータ数やデータサイズ,ネットワーク上のサービスを利用するユーザ数の急激な増加にともない,P2Pネットワークを利用したデータ共有サービスは,今後ますます重要性が高まる.また,共有されるデータの種類も多岐に渡り,共有されているデータに更新が発生する場合も考えられる.したがって,P2Pネットワークに適応できる更新伝播法の確立が必要である.本論文で提案した手法は,データの更新の発生頻度や更新内容を考慮し,効率的に更新データを伝播させることによってネットワーク上のトラヒックを抑制するため,様々な種類のデータに対応でき,ネットワーク上の資源を有効に利用できる.また,更新データを必要としているピアに確実に更新データを伝播させることができるため,更新伝播の信頼性も高い.そのため,多数のユーザが参加する大型データの共有やクラウドコンピューティング環境における基幹技術として,本論文の提案手法は重要な役割を果たすものと考えられる.

5.2 検討課題

本節では,P2Pネットワークにおける複製更新伝播に関する今後の研究課題について述べる.

本論文では,データサイズや所持できる複製の数が全てのピアで一定である環境を想定し,一種類のデータに関する更新伝播木に着目して評価を行っていた.しかし実際には,所持しているデータのサイズやその数はピアによって様々であり,容量の大きなデータの複製を多数所持しているピアは,更新データ伝播による負荷が非常に高まってしまう.ここ

で，木構造を用いた複製更新伝播法では，木構造の葉ノードに位置するピアは他のピアに更新データを伝播させる必要がなく，更新伝播時の負荷が軽減される．そのため，所持する全データに対する更新データ伝播時の負荷を調査し，負荷が大きくなっているピアに関しては，そのピアが参加している更新伝播木上での参加位置を葉ノードに変更させるなどの対策をとり，許容範囲を超えた負荷の集中を防ぐことが考えられる．

また，本論文では，2章で提案した木構造に基づく複製更新伝播法を拡張し，3章でデータアクセス頻度を考慮した複製更新伝播法，4章でデータの更新量を考慮した複製更新伝播法を提案した．ここで，各ピアが設定しているデータ要求条件とデータの更新量を考慮した更新データの受信回数と，各ピアがその複製にアクセスする回数を比較し，更新データもしくは無効化情報のどちらを伝播させるか決定するなど，本論文で提案した手法を統合することで，より様々なネットワーク環境に対応できる手法へと拡張する必要がある．

最後に，本論文では，更新伝播時の遅延を論理ネットワーク上のホップ数として評価を行った．ここで，各章における遅延の評価結果に対する考察でも述べたように，実環境における物理ネットワーク上での遅延は，論理ネットワーク上での評価と類似した特徴を示すと考えられるが，論理ネットワーク上でのホップ数が小さい場合に，必ずしも物理ネットワーク上での遅延も小さくなるとは限らない．また，リンク切断やパケットの再送，経路長の変更などによる細かな遅延の変化までは評価することができず，実環境を想定した場合の詳細な特徴までは検証できていない．さらに，本論文では，データの更新発生頻度や各ピアのデータアクセス頻度に関しても，一定もしくはある分布に従って決定している．しかし実環境では，これらの頻度が突発的に激しく変化する場合もあれば，一定時間ほとんど変化しない場合も考えられる．提案手法はこれらの頻度の変化にも対応はできるが，非常に頻繁にデータが更新されたり，多くのピアが常に更新データを必要としている場合等は，提案手法の性能が向上しにくいことが考えられる．以上の理由より，今後は，提案手法を実システム上に実装して，提案手法の有効性を検証する必要がある．

謝辞

本研究全般に関して，懇切なる御指導と惜しみない御助言を頂きました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 西尾章治郎教授に謹んで御礼申し上げます．

本研究を推進するにあたり，直接の御指導，御助言，御討論を頂きました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 原隆浩准教授に衷心より感謝申し上げます．

本論文をまとめるにあたり，大変有益な御指導と御助言を多数賜りました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 藤原融教授，大阪大学サイバーメディアセンター 馬場健一准教授に心より感謝申し上げます．

講義，学生生活を通じて，学問に取り組む姿勢をご教授頂きました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 岸野文郎教授，薦田憲久教授に厚く感謝申し上げます．

本研究において，多大なる御指導と御助言をご教授頂きました独立行政法人情報通信研究機構/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 下條真司先生，ならびに独立行政法人情報通信研究機構 木俵豊博士に厚く感謝申し上げます．

本研究において，ともに研究を進め，多大なる御協力を頂いた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 神崎映光助教に深く御礼申し上げます．

本研究において，多大なる御助言，御協力，御支援を頂きました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 寺西裕一准教授，大阪大学大学院工学研究科 春本要准教授，大阪大学サイバーメディアセンター 義久智樹准教授，神戸大学大学院工学研究科 寺田努准教授，東京大学情報基盤センター 小川剛史講師，東京大学知の構造化センター 中山浩太郎特任助教に深謝致します．

筆者の所属する研究グループにおいて，有益な御助言を頂いた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 小林由依氏，坂下卓氏，大阪大学工学部電子情報工学科 清野航氏に深く御礼申し上げます．

本研究を進める上で惜しみない御助言，ご協力，研究活動を進めるにあたっての多大な

るご支援を頂いた，パナソニック株式会社 前田和彦氏，株式会社コーエー 木戸裕樹氏，株式会社NTTデータ 宮崎知美氏，NTTコムウェア株式会社 趙勇氏に感謝の意を表します．

本研究を進めるにあたり，多くの御討論や御助言を頂きました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 西尾研究室の諸氏に心より感謝申し上げます．

最後に，研究生活を送る上で，暖かい御支援と多大なる御理解を頂いた両親を始めとする家族に心からの感謝と御礼を申し上げます．

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 (Sept. 2001).

[2] Andrzejak, A. and Xu, Z.: Scalable Efficient Range Queries for Grid Information Services, in *Proceedings of International Conference on Peer-to-Peer Computing (P2P 2002)*, pp. 33–40 (July 2002).

[3] Balakrishnan, H., Kaashoek, M.F., Karger, D., Morris, R., and Stoica, I.: Looking up Data in P2P Systems, *Communications of ACM*, Vol. 46, No. 2, pp. 43–48 (Feb. 2003).

[4] Breslau, L., Cao, P., Fan, L., Phillips, G., and Shenker, S.: Web Caching and Zipflike Distributions: Evidence and Implications, in *Proceedings of IEEE INFOCOM 1999*, pp. 126–134 (Mar. 1999).

[5] Bu, T. and Towsley, D.: On Distinguishing Between Internet Power Law Topology Generators, in *Proceedings of INFOCOM 2002*, Vol. 2, pp. 638–647 (June 2002).

[6] Castro, M., Druschel, P., Kermarrec, A.M., and Rowstron, A.: Scribe: A Large-Scale and Decentralized Application-Level Multicast Infrastructure, *IEEE Journal on Selected Areas in Communications*, Vol. 20, No. 8, pp. 1489–1499 (Oct. 2002).

[7] Chawathe, Y.: Scattercast: An Adaptable Broadcast Distribution Framework, *Multimedia Systems*, pp. 104–118 (July 2003).

[8] Chawathe, Y., Ramabhadran, S., Ratnasamy, S., Lamarca, A., Shenker, S., and Hellerstein, J.: A Case Study in Building Layered DHT Applications, in *Proceedings*

of International conference on Applications, Technologies, Architectures, and Protocols for Computer Communications (SIGCOMM 2005), Vol. 35, No. 4, pp. 97–108 (Oct. 2005).

[9] Chen, Y., Katz, R.H., and Kubiatowicz, J.D.: Dynamic Replica Placement for Scalable Content Delivery, in *Proceedings of International Workshop on Peer-to-Peer Systems (IPTPS 2002)*, pp. 306–318 (Mar. 2002).

[10] Chen, X., Ren, S., Wang, H., and Zhang, X.: SCOPE: Scalable Consistency Maintenance in Structured P2P Systems, in *Proceeding of IEEE INFOCOM 2005*, pp. 1502–1513 (Mar. 2005).

[11] Christos, G., Milena, M., and Amin, S.: Random Walks in Peer-to-Peer Networks: Algorithms and Evaluation, in *P2P Computing Systems*, Vol. 63, No. 3, pp. 241–263 (Mar. 2006).

[12] Chu, Y.-H., Rao, S.G., and Zhang, H.: A Case for End System Multicast, in *Proceedings of SIGMETRICS 2000*, pp. 1–12 (June 2000).

[13] Chu, Y.-H., Rao, S.G., Seshan, S., and Zhang, H.: Enabling Conferencing Applications on the Internet Using an Overlay Multicast Architecture, in *Proceedings of ACM SIGCOMM 2001*, pp. 55–67 (Oct. 2001).

[14] 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* (July 2000).

[15] Cohen, E. and Shenker, S.: Replication Strategies in Unstructured Peer-to-Peer Networks, in *Proceedings of ACM SIGCOMM 2002*, pp. 177–190 (Aug. 2002).

[16] Cuenca-Acuna, F.M., Martin, R.P., and Nguyen, T.D.: Autonomousu Replication for High Availability in Unstructured P2P Systems, in *Proceedings of Symposium on Reliable Distributed Systems (SRDS 2003)*, pp. 99–108 (Oct. 2003).

[17] Datta, A., Hauswirth, M., and Aberer, K.: Updates in Highly Unreliable, Replicated Peer-to-Peer Systems, in *Proceedings of International Conference on Distributed Computing Systems (ICDCS 2003)*, pp. 76–85 (May 2003).

[18] Falchi, F., Gennaro, C., and Zezula, P.: A Content-Addressable Network for Similarity Search in Metric Spaces, *Lecture Notes in Computer Science*, Vol. 4125, pp. 98–110 (May 2007).

[19] Faloutsos, M., Faloutsos, P., and Faloutsos, C.: On Power-Law Relationships of the Internet Topology, in *Proceedings of ACM SIGCOMM 1999*, pp. 251–262 (Aug. 1999).

[20] Ferreira, R.A., Ramanathan, M.K., Awan, A., Grama, A., and Jagannathan, S.: Search with Probabilistic Guarantees in Unstructured Peer-to-Peer Networks, in *Proceedings of International Conference on Peer-to-Peer Computing (P2P 2005)*, pp. 165–172 (Aug./Sept. 2005).

[21] Clarke, I., Sandberg, O., Wiley, B., and Hong, T.W.: Freenet: A Distributed Anonymous Information Storage and Retrieval System, in *Proceedings of International Computer Science Institute Workshop Design Issues in Anonymity and Unobservability*, pp. 311–320 (July 2000).

[22] Gnutella, *http://rfc-gnutella.sourceforge.net/index.html.*

[23] Gupta, A., Agrawal, D., and Abbadi, A.E.: Approximate Range Selection Queries in Peer-to-Peer Systems, in *Proceedings of International Conference on Innovative Data Systems Research (CIDR 2003)* (Jan. 2003).

[24] Gyuwon, S., Suhyun, K., and Daeil, S.: Replica Placement Algorithm for Highly Available Peer-to-Peer Storage Systems, in *Proceedings of International Conference on Advances in P2P Systems (AP2PS 2009)*, pp. 160–167 (Oct. 2009).

[25] Hosseini, M., Ahmed, D.T., Shirmohammadi, S., and Georganas, N.D.: A Survey of Application-Layer Multicast Protocols, *IEEE Communications Surveys & Tutorials*, Vol. 9, No. 3, pp. 58–74 (Mar. 2007).

[26] Jannotti, J., Gifford, D.K., Johnson, K.L., Kaashoek, M.F., and O'Toole, J.W.: Overcast: Reliable Multicasting with an Overlay Network, in *Proceedings of Symposium on Operating Systems Design and Implementation (OSDI 2000)*, pp. 197–212 (Oct. 2000).

[27] JXTA, *https://jxta.dev.java.net/*.

[28] Shin, S., Jung, J., and Balakrishnan, H.: Malware prevalence in the KaZaA file-sharing network, in *Proceedings of ACM SIGCOMM Internet Measurement Conference (IMC 2006)*, pp. 333–338 (Oct. 2006).

[29] Keyani, P., Larson, B., and Senthil, M.: Peer Pressure: Distributed Recovery from Attacks in Peer-to-Peer Systems, in *Proceedings of Networking Workshop 2002*, pp. 306–320 (May 2002).

[30] 木戸裕樹, 前田和彦, 原 隆浩, 西尾章治郎: Peer-to-Peer ネットワークにおけるデータアクセス頻度を考慮した確率的な複製配置手法について, 情報処理学会データベースシステム研究会報告 (2005-DBS-135), Vol. 2005, No. 6, pp. 39–46 (Nov. 2005).

[31] Kleinberg, J.: The Small-World Phenomenon: An Algorithm Perspective, in *Proceedings of ACM Symposium on Theory of Computing (STOC 2000)*, pp. 163–170 (May 2000).

[32] 小林由依, 渡辺俊貴, 神崎映光, 義久智樹, 原 隆浩, 西尾章治郎: P2P ネットワークにおける動的なカテゴリ生成を考慮した検索手法, データ工学と情報マネジメントに関するフォーラム (DEIM 2009) 論文集 (May 2009).

[33] Kobayashi, Y., Watanabe, T., Kanzaki, A., Yoshihisa, T., Hara, T., and Nishio, S.: A Dynamic Cluster Construction Method Based on Query Characteristics in Peer-to-Peer Networks, in *Proceedings of International Conference on Advances in P2P Systems (AP2PS 2009)*, pp. 168–173 (Oct. 2009).

[34] Kubiatowicz, J., Bindel, D., Chen, Y., Czerwinski, S., Eaton, P., Geels, D., Gunnadi, R., Rhea, S., Weatherspoon, H., Weimer, W., Wells, C., and Zhao, B.: OceanStore:

An Architecture for Global-Scale Persistent Storage, in *Proceedings of Architectural Support for Programming Languages and Operating Systems (ASPLOS 2000)*, pp. 190–201 (Nov. 2000).

[35] Leontiadis, E., Dimakopoulos, V.V., and Pitoura, E.: Cache Updates in a Peer-to-Peer Network of Mobile Agents, in *Proceedings of International Conference on Peer-to-Peer Computing (P2P 2004)*, pp. 10–17 (Aug. 2004).

[36] Lv, Q., Cao, P., Cohen, E., Li, K., and Shenker, S.: Search and Replication in Unstructured Peer-to-Peer Networks, in *Proceedings of ACM International Conference on Supercomputing (ICS 2002)*, pp. 84–95 (June 2002).

[37] Markatos, E.P.: Tracing a Large-scale Peer to Peer System: An Hour in the Life of Gnutella, in *Proceedings of IEEE/ACM International Symposium on Cluster Computing and the Grid (CCGRID 2002)*, pp. 65–74 (May 2002).

[38] Maymounkov, P. and Mazieres, D.: Kademlia: A Peer-to-Peer Information System Based on the Xor Metric, in *Proceedings of International Workshop on Peer-to-Peer Systems (IPTPS 2002)*, pp. 53–65 (Mar. 2002).

[39] Ming, Z. and Kai, S.: Popularity-Biased Random Walks for Peer-to-Peer Search under the Square-Root Principle in *Proceedings of International Workshop on Peer-to-Peer Systems (IPTPS 2006)* (Feb. 2006).

[40] 宮崎知美, 渡辺俊貴, 神崎映光, 原 隆浩, 西尾章治郎: P2P ネットワークにおけるユーザの嗜好を考慮した検索方法, 情報処理学会マルチメディア, 分散, 協調とモバイルシンポジウム (DICOMO 2007) 論文集, Vol. 2007, No. 1, pp. 392–399 (July 2007).

[41] 宮崎知美, 渡辺俊貴, 神崎映光, 原 隆浩, 西尾章治郎: P2P ネットワークにおけるユーザの嗜好特性を考慮した検索方法に関する一考察, 情報処理学会マルチメディア, 分散, 協調とモバイルシンポジウム (DICOMO 2008) 論文集, Vol. 2008, No. 1, pp. 668–675 (July 2008).

[42] Miyazaki, T., Watanabe, T., Kanzaki, A., Hara, T., and Nishio, S.: Keyword Search Considering User's Preference in P2P Networks, in *Proceedings of 3rd International*

Conference on Ubiquitous Information Management and Communication (ICUIMC 2009), pp. 462–470 (Jan. 2009).

[43] Nabhendra, B. and Alhussein, A.A.: Optimizing Random Walk Search Algorithms in P2P Networks, *The International Journal of Computer and Telecommunications Networking*, Vol. 51, No. 6, pp. 1499–1514 (Apr. 2007).

[44] 中通 実, 内田 渉, 原 隆浩, 前田和彦, 西尾章治郎: 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, 電子情報通信学会データ工学ワークショップ (DEWS 2004) 論文集 (Mar. 2004).

[45] On, G., Schmitt, J., and Steinmetz, R.: The Effectiveness of Realistic Replication Strategies on Quality of Availability for Peer-to-peer Systems, in *Proceedings of International Conference on Peer-to-Peer Computing (P2P 2003)*, pp. 57–64 (Sept. 2003).

[46] OpenNap, *http://opennap.sourceforge.net/*.

[47] Pendarakis, D., Shi, S., Verma, D., and Waldvogel, M.: ALMI: An Application Level Multicast Infrastructure, in *Proceedings of USENIX Symposium on Internet Technologies and Systems (USITS 2001)*, pp. 49–60 (Mar. 2001).

[48] Pouwelse, J., Garbacki, P., Epema, D., and Sips, H.: The Bittorrent P2P File-Sharing System: Measurements and Analysis, *Lecture Notes in Computer Science*, Vol. 3640/2005, pp. 205–216 (Nov. 2005).

[49] Qiao, Y. and Bustamante, F.E.: Elders Know Best - Handling Churn in Less Structured P2P Systems, in *Proceedings of International Conference on Peer-to-Peer Computing (P2P 2005)*, pp. 77–86 (Aug./Sept. 2005).

[50] Quang, H.V.: Using Logical Identifiers of Nodes in Replica Maintenance, in *Proceedings of International Conference on Advances in P2P Systems (AP2PS 2009)*, pp. 92–97 (Oct. 2009).

[51] Ratnasamy, S., Francis, P., Handley, M., Karp, R., and Shenker, S.: A Scalable Content-Addressable Network, in *Proceedings of ACM SIGCOMM 2001*, pp. 161–171 (Aug. 2001).

[52] Ripeanu, M.: Peer-to-Peer Architecture Case Study: Gnutella Network, *University of Chicago Technical Report* TR-2001-26 (Aug. 2001).

[53] Ripeanu, M., Foster, I., and Iamnitchi, A.: Mapping the Gnutella Network: Properties of Large-Scale Peer-to-Peer Systems and Implications for System Design, *IEEE Internet Computing Journal*, Vol. 6, No. 1, pp. 50–57 (Jan. 2002).

[54] Rowstron, A. and Druschel, P.: Pastry: Scalable, Distributed Object Location and Routing for Large-Scale Peer-to-Peer Systems, in *Proceedings of Middleware 2001*, pp. 329–350 (Nov. 2001).

[55] Roussopoulos, M. and Baker, M.: CUP: Controlled Update Propagation in Peer-to-Peer Networks, in *Proceedings of USENIX Technical Conference (USENIX 2003)*, pp. 167–180 (June 2003).

[56] Saroiu, S., Gummadi, P.K., and Gribble, S.D.: A Measurement Study of Peer-to-Peer File Sharing Systems, Technical Report UW-CSE-01-06-02 (July 2001).

[57] Schmidt, C. and Parashar, M.: Enabling Flexible Queries with Guarantees in P2P Systems, *IEEE Internet Computing*, Vol. 8, No. 3, pp. 19–26 (May 2004).

[58] Stoica, I., Morris, R., Karger, D., Kaashoek, M.F., and Balakrishnan, H.: Chord: A Scalable Peet-to-Peer Lookup Service for Internet Applications, in *Proceedings of ACM SIGCOMM 2001*, pp. 149–160 (Aug. 2001).

[59] Stephanos, A.T. and Diomidis, S.: A Survey of Peer-to-Peer Content Distribution Technologies, *ACM Computing Surveys*, Vol. 36, No. 4, pp. 335–371 (Dec. 2004).

[60] Tan, X. and Datta, S.: Building Multicast Trees for Multimedia Streaming in Heterogeneous P2P Networks, in *Proceedings of International Conference on Multimedia Communications Systems (ICMCS 2005)*, pp. 141–146 (Aug. 2005).

[61] Tran, D.A., Hua, K.A., and Do, T.T.: Zigzag: An Efficient Peer-to-Peer Scheme for Media Streaming, in *Proceedings of IEEE INFOCOM 2003*, pp. 1283–1292 (Mar. 2003).

[62] Wang, Z., Das, S.K., Kumar, M., and Shen, H.: Update Propagation through Replica Chain in Decentralized and Unstructured P2P Systems, in *Proceedings of International Conference on Peer-to-Peer Computing (P2P 2004)*, pp. 64–71 (Aug. 2004).

[63] 渡辺俊貴, 木戸裕樹, 原 隆浩, 西尾章治郎: P2P ネットワークにおける障害耐性向上と遅延減少のための木構造に基づく複製更新伝播について, 情報処理学会マルチメディア, 分散, 協調とモバイルシンポジウム (DICOMO 2006) 論文集, Vol. 2006, No. 6, pp. 313–316 (July 2006).

[64] 渡辺俊貴, 原 隆浩, 木戸裕樹, 中通 実, 西尾章治郎: P2P ネットワークにおける木構造に基づく複製更新伝播法, 情報処理学会論文誌, Vol. 48, No. 2, pp. 527–538 (Feb. 2007).

[65] 渡辺俊貴, 神崎映光, 原 隆浩, 西尾章治郎: P2P ネットワークにおけるデータの使用頻度を考慮した木構造に基づく複製更新伝播法, 電子情報通信学会データ工学ワークショップ (DEWS 2007) 論文集 (Feb. 2007).

[66] Watanabe, T., Hara, T., Kido, Y., and Nishio, S.: An Update Propagation Strategy for Delay Reduction and Node Failure Tolerance in Peer-to-Peer Networks, in *Proceedings of International Symposium on Frontiers in Networking with Applications (FINA 2007)*, pp. 103–108 (May 2007).

[67] Watanabe, T., Kanzaki, A., Hara, T., and Nishio, S.: An Update Propagation Strategy Considering Data Access Frequency in Peer-to-Peer Networks, in *Proceedings of International Conference on Database Systems for Advanced Applications (DASFAA 2008)*, pp. 661–669 (Mar. 2008).

[68] 渡辺俊貴, 神崎映光, 原 隆浩, 西尾章治郎: 非構造型 P2P ネットワークにおける情報爆発を考慮した更新伝播に関する一考察, 情報処理学会 第 70 回全国大会論文集, Vol. 5, pp. 9–10 (Mar. 2008).

[69] 渡辺俊貴, 神崎映光, 原 隆浩, 西尾章治郎: 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, 情報処理学会論文誌, Vol. 49, No. 6, pp. 1819–1832 (June 2008).

[70] 渡辺俊貴, 神崎映光, 原 隆浩, 西尾章治郎: P2P ネットワークにおけるデータの更新量を考慮した更新伝播木管理手法, 情報処理学会研究報告 (データベースシステム研究会 2008-DBS-146), Vol. 2008, No. 88, pp. 85–90 (Sept. 2008).

[71] 渡辺俊貴, 神崎映光, 原 隆浩, 西尾章治郎: P2P ネットワークにおけるデータの更新量を考慮した更新伝播, 電子情報通信学会技術研究報告 (データ工学 DE2008-35), Vol. 108, No. 211, pp. 11–12 (Sept. 2008).

[72] 渡辺俊貴, 神崎映光, 原 隆浩, 西尾章治郎: P2P ネットワークにおけるデータの更新量を考慮した更新伝播手法の提案, 日本データベース学会論文誌, Vol. 7, No. 3, pp. 13–18 (Jan. 2009).

[73] Watanabe, T., Kanzaki, A., Hara, T., and Nishio, S.: Update Propagation Strategies Considering Degree of Data Update in Peer-to-Peer Networks, in *Proceedings of International Conference on Database Systems for Advanced Applications (DASFAA 2009)*, pp. 328–333 (Apr. 2009).

[74] Watanabe, T., Kanzaki, A., Hara, T., and Nishio, S.: A Selective Update Propagation Based on Degree of Data Update in Peer-to-Peer Networks, in *Proceedings of International Conference on Network-Based System (NBiS 2009)*, pp. 52–59 (Aug. 2009).

[75] 渡辺俊貴, 趙 勇, 神崎映光, 原 隆浩, 西尾章治郎: P2P ネットワークにおけるデータの分布を考慮した複製再配置, 電子情報通信学会 (データ工学研究会 DE2009-9), Vol. 109, No. 153, pp. 49–54 (July 2009).

[76] Watanabe, T., Zhao, Y., Kanzaki, A., Hara, T., and Nishio, S.: A Replica Relocation Method for Improving Search Efficiency in P2P Networks, in *Proceedings of International Conference on Advances in P2P Systems (AP2PS 2009)*, pp. 13–18 (Oct. 2009).

[77] Wolfson, O., Jajodia, S., and Huang, Y.: An Adaptive Data Replication Algorithm, in *ACM Transactions on Database Systems (TODS)*, pp. 255–314 (June 1997).

[78] Xiao, L., Zhang, X., and Xu, Z.: On Reliable and Scalable Peer-to-Peer Web Document Sharing, in *Proceedings of International Parallel and Distributed Processing Symposium (IPDPS 2002)*, pp. 15–19 (Apr. 2002).

[79] 山田太造, 相原健郎, 高須淳宏, 安達 淳: Peer-to-Peer システム上での効率的なデータ配置による問合せ処理とロードバランスへの寄与, 情報処理学会論文誌: データベース, Vol. 45, No. SIG5 (TOD22), pp. 93–101 (June 2004).

[80] Yin, L. and Cao, G.: DUP: Dynamic-tree Based Update Propagation in Peer-to-Peer Networks, in *Proceedings of International Conference on Data Engineering (ICDE 2005)*, pp. 258–259 (Apr. 2005).

[81] 趙 勇, 渡辺俊貴, 神崎映光, 原 隆浩, 西尾章治郎: P2P ネットワークにおけるクエリ統計情報を利用した自律分散型複製配置方式, 情報処理学会マルチメディア, 分散, 協調とモバイルシンポジウム (DICOMO 2007) 論文集, Vol. 2007, No. 1, pp. 384–391 (July 2007).

[82] Zhao, B.Y., Kubiatowicz, J.D., and Joseph, A.D.: Tapestry: An Infrastructure for Wide-area Fault-tolerant Location and Routing, *U. C. Berkeley Technical Report* UCB//CSD-01-1141 (Apr. 2000).

[83] Zipf, G.K.: Human Behavior and the Principle of Least Effort, *Addison-Wesley*, pp. 545–564 (1949).