情報処理学会論文誌 Vol. 52 No. 2 368–375 (Feb. 2011)

HyDiff：要求度が変化するオブジェクトのための普及度ベースP2P複合型検索手法

遠藤 伶[†1] 重野 寛[†1]

複合型検索は，オブジェクトの人気により効率が変化する2種の検索モデルを，人気に応じて使い分けることで検索の効率を高めている．既存手法ではオブジェクトの要求度を人気として定義しているが，検索効率には普及度の方がより大きな影響を与える．そのため，要求度が急激に変化し，要求度と普及度に大きな差が出る状況では検索の有用性が低下する．本論文では，オブジェクトの要求度が急激に変化しても有用性が低下しない，普及度ベース複合型検索手法HyDiffを提案する．HyDiffでは，各スーパノードが周囲のオブジェクト情報をもとにグローバル普及度を推定し，検索手法の選択に使用することで，選択ミスによる有用性低下を抑える．また，シミュレーションを行い，HyDiffの有用性を応答率・応答時間・維持コストなどの点から示す．

HyDiff: P2P Hybrid-search Based on Diffusion Rate for Objects with Varying Request Rate

Rei Endo[†1] and Hiroshi Shigeno[†1]

Hybrid-Search uses two kinds of search models whose efficiency changes with object popularity. Hybrid-Search correspond to a popularity of search target object and selects a better search model. An existing technique calculates the popularity from a request rate of the object. However a request rate gives a less influence on search efficiency than a diffusion rate. Therefore a search efficiency tends to decrease with increasing a difference between a request rate and a diffusion rate of a search target object when a request rate has a large change. We propose Hybrid-Search based diffusion rate, HyDiff, in this paper. A search efficiency is not decrease even when a request rate has a large change in HyDiff. Each super nodes estimate a global diffusion rate from a state of neighbor objects. Next, HyDiff selects a better search technique using the estimation global diffusion rate.In addition, we show that HyDiff is useful by simulation.

1. はじめに

P2P技術は個々のユーザが持つローカルなリソースを使用するためのネットワークツールとして大きな可能性を示し，様々なアプリケーションで使用されている[1)–6)]．その中でもGnutella[4),5)]などのファイル共有アプリケーションは，何百万ものユーザが参加するP2Pネットワークとなり，大規模P2Pネットワークに関する様々な課題が分かってきた．

P2Pアプリケーションの大規模化にともない，オブジェクトの検索効率が問題となってきた．P2P検索モデルとして大きく分けて，DHT検索などの構造型検索モデル，Flooding検索などの非構造型検索モデル，そして複合型検索モデルがある[7)–11)]．複合型検索は，構造型，非構造型検索を組み合わせた検索であり，多くのピアが保持する人気オブジェクトに対して非構造型検索，少数のピアしか保持しない不人気オブジェクトに対して構造型検索を使用することで高い検索効率を達成する．複合型検索では，検索対象の人気をどのように求めるか，検索モデルを切り替える閾値をどのように決定するかが問題となる[9)–11)]．既存手法GAB[10)]では，各ノードにおけるローカル人気度をGossipアルゴリズム[12)]を用いて収集し，ネットワーク全体でのグローバル人気度を推定し検索時に使用する．また，効用関数を定義し，一定間隔で各検索手法の有用性を測定することで，検索を切り替える閾値を決定する．

しかし，既存手法は要求度が時間経過で変化する点について考慮しておらず，人気度の再計算周期について十分検討されていない．また，適切に再計算を行ったとしても，要求度が急変するオジェクトを検索する場合，要求度から人気を決める既存手法では検索効率が悪化することがある．検索効率には，オブジェクトの人気，すなわち要求度より，オブジェクトがネットワーク中にどれだけ存在するか，すなわち普及度が大きな影響を持つためである．オブジェクトの要求度が上昇しても，所持ノードが少ないうちは，普及度は要求度に対して低い．そのような場合，不人気オブジェクトに非構造型検索が使用され検索効率が低下する．

本論文では，要求度が急激に変化するオブジェクトに対応するため，普及度ベースの複合型検索手法HyDiffを提案する．HyDiffは，検索対象となるオブジェクトの状態によって，使用する検索手法を切り替える複合型検索の一種である．オブジェクトの要求度が変化する場合，要求度と普及度が一致しないことがあるという点に着目し，適切な検索手法の選択ミ

†1 慶應義塾大学大学院理工学研究科
Graduate School of Science and Technology, Keio University

スによって発生する効率低下を抑える．基本的な構造は Gnutella[4),5)] ネットワークを改良したものであり，いくつかのスーパノードが他の大部分のノードの検索を代理で実行する．そのスーパノードが周囲のノードが所持するオブジェクト情報を定期的に収集し，ローカル普及度情報を保持する．そして，定期的に各スーパノード間で Gossip アルゴリズムを使い，ローカル普及度を収集しグローバル普及度を推定し，検索手法の選択に使用する．

以下，2 章で関連研究とその問題点を明らかにし，3 章で問題を解決する新しい手法を提案する．そして 4 章でシミュレーションにより提案手法の有用性を示し，5 章に結論を示す．

2. 関 連 研 究

2.1 構造型検索モデルと非構造型検索モデル

一般的に，構造型検索モデルとして DHT 検索，非構造型検索モデルとして Flooding 検索が使われる．DHT 検索は，ほぼすべてのオブジェクトの検索が可能で，どのようなオブジェクトでも検索にかかるコストは同じである．また，オブジェクト配置・トポロジに厳密なコントロールが必要であり，ノードの加入・離脱に対し脆弱性を持つ．一方，Flooding 検索は，数多く存在する人気オブジェクトを高速に発見できるが，数が少ない不人気オブジェクトの発見には時間がかかる，もしくは発見できない．また，オブジェクト配置・トポロジにいっさいの制限を持たないため，ノードの加入・離脱の影響をうけない．

以上のような特徴をふまえ，複合型検索では人気オブジェクトには Flooding 検索を，不人気オブジェクトには DHT 検索を使用する．また，ノードの離脱が激しく一時的に DHT が使用不能になっても，Flooding 検索を使用することで最低限の検索機能を維持できる．

2.2 Gossip ベースの複合型検索手法

複合型検索では一般に，Gossip アルゴリズムで人気情報を収集し，Flooding 検索と DHT 検索を使い分ける．スーパノードで，各単語が一定期間で検索された数，すなわちローカル人気度を計算する．次にローカル人気度を Gossip アルゴリズムにより集計し，各単語がネットワーク全体で一定期間検索された数，すなわちグローバル人気度を推定する．そして，どの人気で検索を切り替えるかを示す検索閾値を求め，Flooding 検索と DHT 検索のどちらを使うかを検索実行前に決定することで，応答時間，帯域消費量を削減している．

Gossip アルゴリズムによるグローバル人気度推定

グローバル人気度の推定に使用する Gossip アルゴリズムは，各スーパノードが値の交換を繰り返すことで平均値の近似解を計算する手法であり，計算に必要な交換回数はノード数 N に対して $O(logN)$ である[12)]．各スーパノードは，各単語のローカル人気度のリストを，

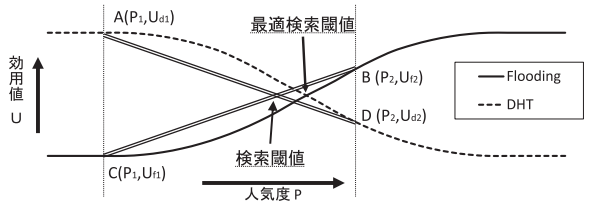

図 1　検索閾値の推定
Fig. 1　An estimate of a search thershold.

ランダムに選んだ相手と交換し人気度の平均値を算出する動作を繰り返すことで，最終的にグローバル人気度の近似解を得る．繰返しの終了条件は，すべての単語において平均値計算前と後の値の差分が，事前に設定された Gossip 閾値より小さくなった場合である．

効用関数を用いた検索閾値の決定

検索閾値を決めるには，人気の異なる 2 つのオブジェクトを選び，それぞれについて DHT 検索，Flooding 検索を実行する．その際に計測した応答時間や消費帯域などから，各人気における各検索効率を算出する．次に，検索効率が人気度の変化に対し線形に変化すると近似して，効率が逆転する推定人気度を求め，それが検索閾値となる．既存研究では検索効率を数値として扱うために，式 (1) のような効用関数を定義している．

$$U = (R > 0?1 : 0) \times \omega_1 + min(R, R_{max}) \times \omega_2 - T \times \omega_3 - B \times \omega_4. \quad (1)$$

ここで，R は検索に対する応答数，R_{max} は最大応答数，T は最短応答時間，B は消費帯域，ω は重みであり，式 (1) によって求まる値を効用値と呼び，任意の 1 回の検索に対する検索効率を数値で表す．式 (1) は R が高いほど，T，B が低いほど有用性が高いことを表し，R_{max} を含む第 2 項は，過剰な検索応答数は効用値に寄与しないことを表している．一般に応答数が多いほど検索効用値は高いが，過剰な応答数は有益ではないとされるため，効用値に寄与する応答数の上限値として最大応答数 R_{max} を使用する．

図 1 に，人気度 P_1，P_2 のオブジェクトを用いた検索閾値の推定に関して示す[⋆1]．DHT 検索の人気度による効用値の変化は 2 点 (P_1, U_{d1})，(P_2, U_{d2}) を結んだ線分によって，同様に，Flooding 検索の効率変化は 2 点 (P_1, U_{f1})，(P_2, U_{f2}) を結んだ線分によって近似でき，2 つの線分の交点が検索閾値となる．図に示すとおり，算出された検索閾値は，理論上の最

⋆1 図は分かりやすさのため単純化したが，正確には人気度の上昇にともない DHT の効用値も上昇し，Flooding の効用値はそれ以上に上昇する．

適な検索閾値の近似値でしかない．しかし，算出された検索閾値により近い人気を持つオブジェクトを 2 つ選びだし，再度同様の処理を繰り返すことで最適値に収束し，ノード数に限らず 20 回程度の繰返しで，検索閾値が最適値に十分近づくことが分かっている．さらに，ネットワーク状態の変化などにより最適値が変動しても，それに追従することができる．

2.3 問　題　点

Gossip ベース複合型検索に関する既存研究では，オブジェクト人気度が時間変化する点を考慮していない．人気度が変化した場合，人気度を再計算しなければ適切な検索モデルを選択できない．しかし，既存研究では再計算について十分検討していないため，人気度の急変に対処するには，定期的な人気度再計算方法，その実行間隔についての検討が必要である．

また，効用値に影響を与えるのは，そのオブジェクトがどれだけネットワーク中に存在するかを表す普及度であるにもかかわらず，既存研究では，一定期間に検索された回数，すなわちの要求度をオブジェクトの人気度として使う．既存研究で想定される P2P ファイル共有アプリケーションにおいて，一般のユーザは他ユーザからファイルを入手し，しばらく別の他ユーザへ送信を行った後ファイルを削除する．そのため，要求度が変化した後に普及度も追従して変化し，オブジェクト要求度の変化が緩やかな場合，要求度と普及度はほぼ同等と見なせるため，既存研究は高い検索効率を実現できる．しかし，オブジェクト要求度が急変した場合，普及度と要求度の差は大きくなり，同等と見なせない．あるオブジェクトの要求度が急上昇した場合，そのオブジェクトは，要求度は高いが普及度は低い状態がしばらく続く．そして，そのオブジェクトを検索した場合，そのオブジェクトは普及度が低い，つまりネットワーク中にほとんど存在してないため，DHT 検索でなければ発見できないにもかかわらず，要求度が高いため Flooding 検索を選択してしまう．また，要求度の変化が激しくなるほど，普及度の変化との食い違いが大きくなるため，検索モデルの選択失敗が増える．

3. HyDiff：Hybrid-Search Based Diffusion Rate の提案

3.1 概　　要

HyDiff は，要求度が変化して要求度と普及度の変化の差が大きくなり，適切な検索方法の選択に失敗するという問題に対処するため，検索方法の決定に要求度ではなく普及度を利用する Gossip ベースの複合型検索手法である．そのために，検索対象となるオブジェクトのグローバル普及度を周期的に求める必要がある．グローバル普及度とは，ネットワーク全体におけるオブジェクトの普及度のことであり，また，ネットワークの一部における部分的なオブジェクト普及度をローカル普及度と定義する．HyDiff では，P2P ネットワークのスーパノード単位でローカル普及度を計算し，さらにそれらローカル普及度を利用してグローバル普及度を推定する．HyDiff の基本的な構造は Gnutella[5] ネットワークを改良したものであり，いくつかのスーパノードが検索ネットワークを構築し，その他のノードは 1 つのスーパノードの子ノードとしてネットワークに参加する．そして，スーパノードは自身の子ノードの検索を代理で実行する．そのため，スーパノードが自身の子ノードが所持するオブジェクトの情報を収集することは容易である．この 1 つのスーパノードに接続しているノード群の中での普及度がローカル普及度となる．そして，各スーパノード間で Gossip アルゴリズムを使うことで，ローカル普及度からグローバル普及度を推定する．

3.2 ローカル普及度の算出

スーパノードが自身の子ノードの所持オブジェクトの情報を集めることでローカル普及度を把握し，ローカルオブジェクトリストと普及度リストを作成する．ローカルオブジェクトリストとは，検索要求が来た場合に，検索要求に含まれるオブジェクトをどの子ノードが所持しているかを判断するためのリストである．普及度リストとは，オブジェクトに含まれる単語を，いくつの子ノードが所持しているかを表すリストである．普及度リストの値の初期値は，リストを所持するスーパノードの子ノード間でのローカル普及度であり，Gossip アルゴリズムによる更新を繰り返すことで最終的にグローバル普及度の近似値となる．また，ローカル普及度の収集は定期的に実行し，その周期を人気推定間隔と呼ぶ．

スーパノードはまず Object_Info_Request を自身の子ノードに送信する．Object_Info_Request を受信したノードは，自身が持つオブジェクト名を Object_Info_Reply に含め返信する．Object_Info_Reply を受信したスーパノードは，メッセージに含まれるオブジェクト名と相手のアドレスを，ローカルオブジェクトリストに記載する．さらに，オブジェクト名に含まれる単語を普及度リストに記録する．異なる Object_Info_Reply のオブジェクト名に同一単語が含まれる場合，重複数をカウントして普及度リストに記録する．

普及度リスト作成時に，あるノードが Query の対象に含まれるかどうかは，オブジェクトを持っているかどうかで決まるべきで，いくつ持っているかは関係ないという点に注意する必要がある．そのため，1 つの Object_Info_Reply に同じ単語が複数含まれていても，1 つしか含まれていない単語と同様に扱う．1 つのノードだけが持っているオブジェクトがあった場合に，もしそのノードが同じオブジェクトを無限に所持していたとしても，ネットワーク中でそのオブジェクトを発見できる確率は，結局 1 つのノードを発見できる確率と同じである．そのため，そのようなオブジェクトを Flooding 検索で探しても見つかる可能性は低い．

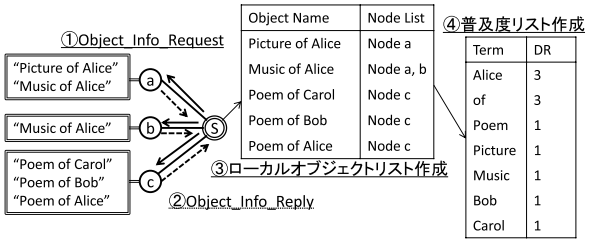

図 2　普及度リストの作成
Fig. 2　A making of diffusion rate list.

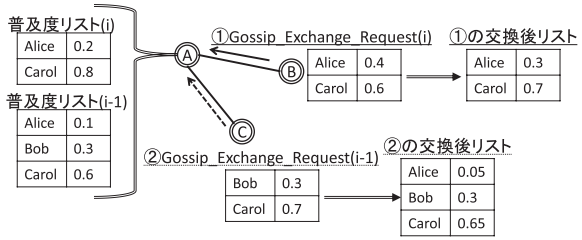

図 3　シーケンス番号ごとの普及度リストの交換
Fig. 3　A exchange of diffusion rate list every sequence number.

図 2 にノード a，b，c を子ノードとして持つスーパノード S の普及度リスト作成手順を示す．スーパノード S が子ノードに Object_Info_Request を送信し，子ノードは Object_Info_Reply を返信する．次に，スーパノード S はローカルオブジェクトリストと普及度リストを作成する．このとき，ノード C からの Object_Info_Reply には単語 Poem が 3 つ含まれ，また，全 Object_Info_Reply 中に単語 Picture は 1 つだけある．しかし両単語を所持するノード数は同じため，スーパノード S の普及度リストでは同じ普及度となる．

3.3　グローバル普及度推定

HyDiff は，既存手法と同様に Gossip アルゴリズムを使用して，普及度リストの値交換を繰り返してグローバル普及度を推定する．ただし HyDiff では，各ノードが周期的に Gossip アルゴリズムを実行するため，各回の Gossip アルゴリズムを Gossip シーケンス番号によって区別する点で異なる．また，要求された Gossip シーケンス番号に対応するローカル普及度を提供するため，ローカル普及度の履歴を保持する．新規参加ノードは Gossip シーケンス番号を知らないが，その場合は，他のスーパノードから最初に受信した Gossip メッセージに含まれる Gossip シーケンス番号を初期値として Gossip アルゴリズムに参加する．シーケンス番号によって区別し，古い履歴を保持する理由は，普及度推定アルゴリズムが各ノードで非同期的に動作することが原因で，一部のノードの普及度推定に時間がかかるなどの影響を抑えるためである．仮に，普及度が変化する状況で普及度の新旧をまったく区別しなかった場合，大きく異なる時間帯の普及度が入り混じって交換が行われる．Gossip アルゴリズムでは，交換前と交換後の値の差分が，事前に設定された Gossip 閾値より小さくなった時点で算出値が平均値に十分近づいたと判定する．その性質上，ほとんど収束していない新しい値が途中で混じることを許すと，値が不正確になるだけでなく，最悪な場合は収束しない．そのため，計算が遅くなったノードの普及度リストの交換に，新しすぎる普及度リストの混入を防ぐために，ローカル普及度の履歴を保持する．

グローバル普及度推定は，ローカル普及度の収集の終了後に実行する．まずスーパノードは自身の普及度リストを，Gossip シーケンス番号と関連づけ，既知のスーパノードからランダムに選んだ相手に Gossip_Exchange_Request として送信する．Gossip_Exchange_Request は普及度リストとシーケンス番号が含まれたメッセージである．Gossip_Exchange_Request を受信したスーパノードは，Gossip シーケンス番号と一致する自身の普及度リストを Gossip_Exchange_Reply に含めて返信する．そして，両方のスーパノードが普及度リストの値を，自身のリストと交換したリストの値の平均値に更新する．以上の動作を既存手法と同様に終了条件を満たすまで繰り返す．

また，グローバル普及度の推定が終了した後でも，Gossip_Exchange_Request を受信した場合に，メッセージ含まれる Gossip シーケンス番号と一致する普及度リストを所持していた場合，上記と同様の交換手順を実行する．

図 3 はスーパノード A，B，C による普及度リストの交換を示す．また，図では Gossip シーケンス番号 i に対応するメッセージ，普及度リストを，それぞれ gossip_Exchange_Request(i)，普及度リスト (i) として表現している．スーパノード A，B はともに普及度リスト (i) の交換を行っている最中である．そして，スーパノード B から A に対して Gossip_Exchange_Request(i) が送信された場合，スーパノード A は自身の最新のリストである普及度リスト (i) を Gossip_Exchange_Reply に含めて返信する．結果として，スーパノード A，B の普及度リスト (i) は Alice = 0.3，Carol = 0.7 という同じ内容のリストになる．一方，スーパノード C はまだ i-1 番目の普及度リスト交換が終了していないため，スーパノード A に対して Gossip_Exchange_Request(i-1) を送信している．スーパノード A は普及度リスト (i-1) は古いため使用しないが，スーパノード C のために普及度リスト (i-1) を

含んだ Gossip_Exchange_Reply を返信することになる．

3.4　検索閾値の決定

HyDiff は DHT 検索と，Flooding 検索を使い分ける複合型検索であり，検索方法を選択するための検索閾値を決定する必要がある．検索閾値の決定は既存研究と同様に，式 (1) で示される効用関数を用いた方法を使用する．ただし，HyDiff では要求度ではなく普及度を使用する点と，グローバル普及度推定と検索閾値決定がセットで繰り返される点で異なる．

閾値の決定は，各検索の，応答数，最短応答時間，消費帯域を計測し，それらの値から効用関数を用いて効用値を算出し，どちらの検索が効率的かを判断して行う．また，検索閾値の計算の実行は，グローバル普及度推定が終了した時点で開始される．まずスーパノードは 2 つの単語を選択し，それぞれの単語について，Search_Query_Request に転送回数を記録する特殊なフィールドを含めて，DHT 検索と Flooding 検索の両方を行う．2 つの単語の選択方法は，初回実行時は普及度リストからランダムに選択し，1 度検索閾値を求めた以降は，現在の検索閾値より普及度が大きい単語のうち最も検索閾値に近い単語と，普及度が小さい単語のうち最も検索閾値に近い単語を選択する．この際，各単語の普及度として最新の Gossip シーケンス番号に対応する普及度リストの値を使用する．そして，スーパノードは返信された Search_Query_Reply の情報から式 (1) を用いて効用値を算出し，既存手法と同様の方法を用いて検索閾値を決定する．

3.5　検索の実行

検索はすべてスーパノードを通して行われ，検索対象の普及度により，DHT 検索と Flooding 検索を使い分ける．まず検索を行うノードはスーパノードに対して，Search_Query_Request を送信する．HyDiff は，どのノードも必ずスーパノードの子ノードとなるため，必ずスーパノードのアドレスを知っている．Search_Query_Request を受信したスーパノードは，最新の普及度リストを参照し検索対象の普及度を計算する．検索対象オブジェクトの普及度は，そのオブジェクト名が含む各単語の普及度の最小値である．求めた普及度が検索閾値より大きければ Flooding 検索，閾値と同じか，閾値以下ならば DHT 検索をスーパノードが代理で行う．

4. 性 能 評 価

4.1　シミュレーション環境

HyDiff の有用性評価のため，ファイル共有ネットワークを想定したシミュレーションを実装し，あるオブジェクトの要求度を時間経過により変化させ，既存手法に周期的な人気度の再計算を組み込んだ方式である，P-GAB（Periodic GAB）と比較した．

表 1　シミュレーションパラメータ
Table 1　Simulation parameter.

シミュレーション時間	96 時間
参加ノード数	10,000 ノード
平均参加・離脱率	0.75 ノード/秒
ウルトラピア	5%
平均所持オブジェクト数	10 個
Gossip 閾値	0.01
最大応答数（R_{max}）	25
要求度変化間隔	10–40 時間
平均ファイル消去時間	4 時間
人気推定間隔	40–3,600 秒
スーパノード生成確率	0.1

表 1 にシミュレーションで用いたパラメータを示す．参加・離脱率は秒間何ノードが離脱し参加するかを表す．シミュレーション上ではノードの離脱と参加を一組で扱い，必ず同数となる．したがって，システム内のノード数はつねに一定である．また各ノードは平均 10 個，最低 1 つ以上のオブジェクトを所持している．ウルトラピアは，P2P システムに特に貢献するノードであり，ファイル共有ネットワークにおいては，積極的にファイルを収集し，長期間ファイルを削除しないで他ノードに配布するノードである．要求度変化間隔は，要求度変化の度合いを表し，要求度変化間隔 10 時間の場合，10 時間の間にネットワークに参加するノードの半分が検索要求を行うことを表す．つまり，要求度変化間隔が短いほど，要求度の変化が激しい．人気推定間隔は，既存手法においては要求度，HyDiff においては普及度を推定する間隔を表す．スーパノード生成確率はノードがネットワークに参加する際に，スーパノードになる確率である．また，Gossip 閾値，最大応答数，スーパノード生成確率はどちらの手法でも使用するパラメータで，既存研究におけるパラメータを参考にし，ウルトラピアなどは一般的なファイル共有アプリケーションで用いられる値を使っている[10]．

4.2　評 価 項 目

効用関数を使用した効用損失率，検索時に必要な消費帯域と検索の応答時間，検索機能の維持に必要な帯域コストについて評価を行った．効用損失率は，最適効用値理論値からどれだけ損失しているかを表す値である．最適効用値理論値は，つねに適切な検索モデルの選択に成功する最適複合型検索が使用できた場合の効用値であり，すなわち複合型検索で達成できる最大の効用値を表す．シミュレーション上であるオブジェクトに対するすべての検索の

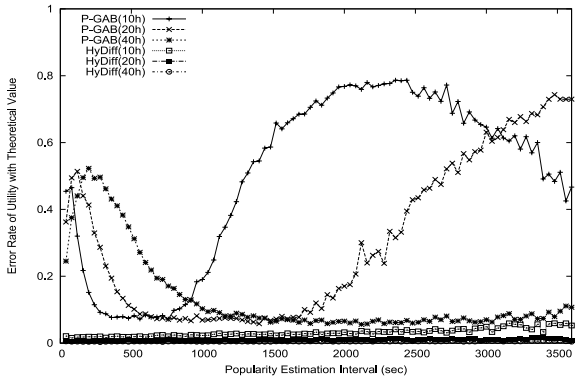

図 4　人気推定間隔の変化による効用損失率への影響
Fig. 4　Error rate from utility theoretical value versus popularity estimation interval.

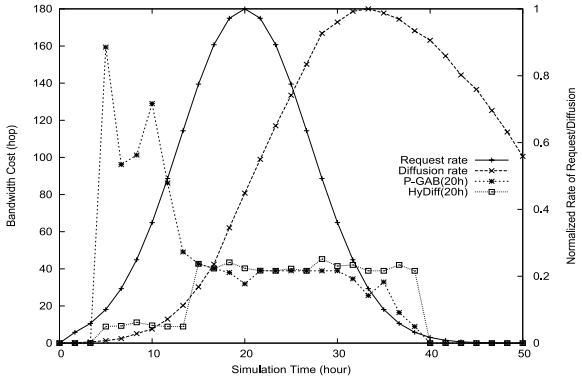

図 5　消費帯域の比較
Fig. 5　Bandwidth cost versus simulation time for HyDiff and P-GAB.

集合を X とし，そのオブジェクトに対して行われたある 1 回の検索 x，$x \in X$ を評価対象の検索手法で行った場合の効用値を $U\_search(x)$，最適複合型検索で行った場合の効用値を $U\_theoretical(x)$ とした場合，効用損失率 E は次式，

$$\mathrm{E}=\frac{\sum_{x\in X} U\_theoretical(x)-\sum_{x\in X} U\_search(x)}{\sum_{x\in X} U\_theoretical(x)}, \tag{2}$$

によって算出され，0 も近いほど理想的な状態に近いことを表す．

4.3　効用関数による評価

図 4 に人気推定間隔を変化させ，オブジェクトの要求度変化間隔を 10 時間，20 時間，40 時間とした場合の効用損失率を示す．また，検索はシミュレーション時間中に各ノードが要求度変化間隔に従った確率で 1 回行い，その際の計測値の合計を図に示した．

HyDiff は，どの要求度変化間隔の場合でも，既存手法である P-GAB に対して安定して理論差が小さく，また，いずれの場合も人気推定間隔を短くするほど効用損失率は下がる．要求度変化間隔が 20 時間の場合は人気推定間隔 1,200 秒以下で，要求度変化間隔が 40 時間の場合は人気推定間隔 5,800 秒以下で，それぞれ効用損失率が 1%以下になり，それ以上人気推定間隔を短くしてもほとんど変化をしなくなる．実際に起こりうる要求度の変化は，激しいオブジェクトの場合で 1，2 日の間にほとんどの検索が集中する程度であり，すなわち要求度変化間隔が 20 時間の場合が，実際に起こりうる中でもかなり激しいケースである[13]．

一方，既存手法において要求度変化間隔が 20 時間の場合，人気推定間隔が 400〜1,800 秒の間で効用損失率が最低となっていて，その範囲より人気推定間隔を大きくしても小さくしても損失率は大きくなる．この傾向は，どんな要求度変化間隔でも同様で，どのような場合でも最も小さい効用損失率は 8.3%程度である．既存手法は人気推定間隔中に検索キーワードをカウントして人気度を求めるが，推定間隔ごとにカウントをリセットしているため，検索要求が行われる間隔以上に人気推定間隔を短くなると，間隔ごとの人気度のばらつきが大きくなり，かえって性能が低下するためである．また，要求度変化間隔が大きく，つまり要求度の変化が緩やかになるほど，効用損失率が最低となる間隔が多くなる．これは，要求度の変化が緩やかになるほど，要求度と普及度との差が小さくなるためだと考えられる．

4.4　消 費 帯 域

図 5 に，要求度変化間隔 20 時間，普及度推定間隔 1,200 秒とした場合の，1 検索あたりの平均消費帯域，普及度，要求度の時間変化を示す．要求度変化間隔と普及度推定間隔の値は，図 4 において両手法とも，ともに最も低い効用損失率であった範囲から選んだ．また，要求度，普及度は，最大値で正規化した．普及度に対して要求度が高い 13 時間までに，既存手法は大きく帯域を消費した．検索回数が多いため既存手法は Flooding 検索を選択したが，要求度が上がった直後は普及度は低いため Flooding 検索での発見に失敗し，余計な帯域を消費した．一方，HyDiff では DHT 検索を行うため消費帯域が少ない．13 時間以降はしばらく，両手法とも Flooding 検索を選択し，また普及度が十分高いため安定している．

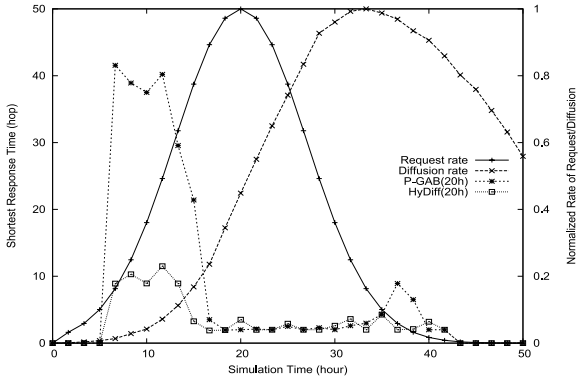

図 6　最短応答時間の比較
Fig. 6　Shortest response time versus simulation time for HyDiff and P-GAB.

30 時間から 40 時間にかけては HyDiff の方が消費帯域がわずかに大きい．これは，既存手法では検索を行うノードが減少したため，人気度が低下したと判断し DHT 検索に切り替わっているためである．検索手法の選択に使用する効用関数には消費帯域以外のパラメータも含まれているため，HyDiff では Flooding 検索を続けて使用しているため，消費帯域の面では既存手法よりもわずかに多い．また，シミュレーション全体での平均消費帯域を比較すると，既存手法が 49.2 hop であるのに対し，HyDiff では 32.7 hop と，33%近く削減できている．

4.5　応 答 時 間

図 6 に，要求度変化間隔 20 時間，普及度推定間隔 1,200 秒とした場合の，1 検索あたりの平均最短応答時間，普及度，要求度の時間変化を示す．応答時間は P2P が使用する物理ネットワークの構造に依存するため，ここでは P2P ネットワーク上での hop 数で時間を表す．まず，15 時間以下の状態では既存手法では，応答が非常に遅い．これは図 5 の場合と同じく，既存手法が Flooding 検索を選択するためである．普及度が低い状態で Flooding 検索を行った場合，結局ファイルを見つけることができず，Flooding 検索の後に DHT 検索を行うことになり，最初から DHT 検索を選択する HyDiff に比べて 4 倍近い時間が必要となる．35 時間から 40 時間にかけては，図 5 と同様に，既存手法は DHT 検索を選択する．しかし検索されなくなっても普及度自体はまだ非常に高く，Flooding 検索を行う HyDiff の方が応答時間が短い．図 5 で示したように，30 時間から 40 時間にかけて，HyDiff の方が

表 2　要求度一定時における各方式の比較
Table 2　A efficiency of P-GAB and proposal technique when request rate is static.

	応答範囲	平均最短応答時間	平均検索コスト	維持コスト
HyDiff	99.9	0.99	1.00	1.01
P-GAB	99.9	1.00	1.00	1.00

帯域を多く消費していたが，代わりに応答時間は HyDiff の方が早い．以上より，要求度が大きく低下し，さらに普及度はまだ高いような状態では，応答時間と帯域コストにトレードオフの関係がある．また，シミュレーション全体での平均最短応答時間を比較すると，既存手法が 11.1 hop であるのに対し，HyDiff では 3.49 hop と，70%近く削減できている．

4.6　要求度一定時における比較

HyDiff は要求度の変化に対応するため，要求度ではなく普及度を使用しているが，普及度の収集は要求度の収集よりも多くの帯域を維持コストとして必要とする．また，現実の環境では要求度の変化が激しいオブジェクトでけでなく，要求度一定のオブジェクトが混在している場合も当然考えられる．要求度の変化がない場合，要求度を使用する既存手法でも検索モデルの選択にほとんど失敗しないため，普及度を算出するために所持オブジェクト情報を収集している HyDiff は，帯域コストの面で悪化する可能性がある．そのため，そのような HyDiff にとって不利な，オブジェクトの要求度が一定の場合で両手法の比較を行った．

表 2 に要求度が変化しない場合の応答範囲，平均最短応答時間，平均検索コスト，維持コストを示す．応答範囲はネットワーク中に存在するオブジェクトの発見成功率である．また，平均最短応答時間は各検索で最も早い発見応答が返るまでにかかった時間の平均，平均検索コストは 1 回の検索を実行するのに必要な帯域コストの平均を，それぞれ P-GAB を 1 とした場合の比で表している．表より，要求度が一定の場合は，両手法ともほとんど同性能となることが分かる．さらに，HyDiff においてローカルオブジェクト情報の収集によって増加している維持コストは，わずか 1%であり，ほとんど悪化していない．これは，DHT 構造の維持と Gossip アルゴリズムに必要なコストが，ローカルオブジェクト情報の収集にかかるコストに比べて非常に大きいためであると考えられる．

5. お わ り に

本論文では，要求度が急変するオブジェクトに対応するため，普及度ベース複合型検索手法 HyDiff を提案した．HyDiff は，2 種の検索モデルを併用する複合型検索の一種であり，要求度ではなく普及度を用いることで，要求度が変化する状況における効率低下を抑える．

HyDiff をオブジェクトの要求度変化を考慮したシミュレーションにより評価し，効用損失率を安定して低く抑えられることを示した．特に 20 時間の間に要求度が急変するようなオブジェクトに対し，人気推定間隔を 1,200 秒にすることで安定して効用損失率を 1%程度に抑えられた．また，特に普及度より要求度が高い状況で消費帯域を抑え，平均で約 33%の改善ができること，最短応答時間を短くでき，平均で約 70%の改善ができることを示した．

以上より，要求度が急変化する環境下で，HyDiff は適切な検索手法の選択失敗率を下げ，選択失敗が原因の効用値低下を抑えられることが確認でき，本提案の有用性を示した．

謝辞　本研究はグローバル COE プログラム「アクセス空間支援基盤記述の高度国際連携」（C12）の助成を受けたものである．

参 考 文 献

1) Chen, A. and Muntz, R.R.: Peer clustering: A hybrid approach to distributed virtual environments, *NetGames '06: Proc. 5th ACM SIGCOMM Workshop on Network and System Support for Games*, New York, NY, USA, ACM, p.11 (2006).
2) Liu, Z., Shen, Y., Ross, K.W., Panwar, S.S., Member, S. and Wang, Y.: LayerP2P: Using Layered Video Chunks in P2P Live Streaming, *IEEE Trans. Multimedia*, Vol.11, No.7, pp.1340–1352 (2009).
3) Liu, Y., Ni, L.M., Xiao, L. and Esfahanian, A.-H.: Approaching Optimal Peer-to-Peer Overlays, *International Symposium on Modeling, Analysis, and Simulation of Computer Systems*, Vol.0, pp.407–414 (2005).
4) Ripeanu, M., Iamnitchi, A. and Foster, I.: Mapping the Gnutella network, *IEEE Internet Computing*, Vol.6, pp.50–57 (2002).
5) Chawathe, Y., Ratnasamy, S., Breslau, L., Lanham, N. and Shenker, S.: Making gnutella-like P2P systems scalable, *Proc. 2003 Conference on SIGCOMM*, New York, NY, USA, ACM, pp.407–418 (2003).
6) Chen, Y.-Y., Jan, J.-K., Chi, Y.-Y. and Tsai, M.-L.: A Feasible DRM Mechanism for BT-Like P2P System, *Proc. 2009 International Symposium on IEEC*, Washington, DC, USA, IEEE Computer Society, pp.323–327 (2009).
7) Stoica, I., Morris, R., Karger, D., Kaashoek, M.F. and Balakrishnan, H.: Chord: A scalable peer-to-peer lookup service for internet applications, *Proc. 2001 Conference on SIGCOMM*, New York, USA, ACM, pp.149–160 (2001).
8) Chen, H., Jin, H., Wang, J., Chen, L., Liu, Y. and Ni, L.M.: Efficient multi-keyword search over 2 web, *Proc. 17th International Conference on World Wide Web*, New York, NY, USA, ACM, pp.989–998 (2008).
9) Loo, B.T., Hellerstein, J.M., Huebsch, R., Shenker, S. and Stoica, I.: Enhancing P2P file-sharing with an internet-scale query processor, *Proc. 30th International Conference on VLDB*, VLDB Endowment, pp.432–443 (2004).
10) Zaharia, M. and Keshav, S.: Gossip-based search selection in hybrid peer-to-peer networks: Research Articles, *Concurr. Comput.: Pract. Exper.*, Vol.20, No.2, pp.139–153 (2008).
11) Shi, X. and Han, J.: Popularity Biased Hybrid Search in P2P Systems, *International Conference on Grid and Cooperative Computing*, Los Alamitos, CA, USA, IEEE Computer Society, pp.173–176 (2006).
12) Boyd, S., Ghosh, A., Prabhakar, B. and Shah, D.: Randomized Gossip Algorithms, *IEEE/ACM Trans. Netw.*, Vol.14, No.SI, pp.2508–2530 (2006).
13) Luo, J.-G., Zhang, Q., Tang, Y. and Yang, S.-Q.: A Trace-Driven Approach to Evaluate the Scalability of P2P-Based Video-on-Demand Service, *IEEE Trans. Parallel Distrib. Syst.*, Vol.20, No.1, pp.59–70 (2009).

(平成 22 年 5 月 31 日受付)
(平成 22 年 11 月 5 日採録)

遠藤　伶（学生会員）

2008 年慶應義塾大学理工学部情報工学科卒業．2010 年同大学大学院理工学研究科博士前期課程修了．現在，同大学院博士後期課程在学中．P2P ネットワークの研究に従事．

重野　寛（正会員）

1990 年慶應義塾大学理工学部計測工学科卒業．1997 年同大学大学院理工学研究科博士課程修了．現在，同大学理工学部准教授．博士（工学）．情報処理学会学会誌編集委員，同論文誌編集委員，同マルチメディアと分散処理研究会幹事，同モバイルコンピューティングとワイアレス通信研究会運営委員等を歴任．ネットワーク・プロトコル，モバイルコンピューティング，ITS 等の研究に従事．著書『コンピュータネットワーク』（オーム社），『ユビキタスコンピューティング』（オーム社）等．電子情報通信学会，IEEE，ACM 各会員．