

固定端末・モバイル端末混在環境における遅延を考慮した DHT アルゴリズム拡張手法の提案

中村 嘉隆^{1,a)} 舟橋 知論² 白石 陽¹ 高橋 修¹

受付日 2013年5月14日, 採録日 2013年10月9日

概要: 近年, 負荷分散やスケーラビリティ, 耐故障性に優れた P2P (Peer-to-Peer) 技術がさかんに研究されている. P2P 技術はこれまでファイル共有ソフトやストリーミング配信や IP 電話などさまざまなサービスに応用されている. モバイル端末自身の性能向上のほか, 近年の LTE や 3G アクセス, Wi-Fi などの無線通信技術の発展にともない, スマートフォンやタブレット端末などのモバイル端末においてもこれらのサービスが利用される機会が増加している. またこれらモバイル端末の普及率の向上にともなって, モバイル端末による P2P 利用の需要は今後, いっそう高まっていくことが予想されている. 本論文では固定・モバイル端末の混在 (Fixed Mobile Convergence, FMC) 環境下での P2P の利用を想定し, モバイル端末の存在を考慮した DHT (Distributed Hash Table) アルゴリズムの拡張手法の提案を行う. モバイル端末のバッテリー消費を抑えるため, FMC 環境における計算資源である固定端末へモバイル端末の通信処理を委譲する非対称型 DHT アルゴリズムを提案する. これにより, モバイル端末においてバッテリー消費量の大きな通信処理負荷を抑制することができ, かつ従来の探索性能を維持した DHT を構築することが可能となる. 提案手法について, シミュレーションによる性能評価を行った結果, 従来手法より探索遅延時間を改善できていることが分かった.

キーワード: P2P, DHT, FMC, 階層型 DHT, 非対称型 DHT

An Extension Method of DHT Algorithm Considering Delay in Fixed Mobile Convergence Environment

YOSHITAKA NAKAMURA^{1,a)} TOMONORI FUNAHASHI² YOH SHIRAISHI¹ OSAMU TAKAHASHI¹

Received: May 14, 2013, Accepted: October 9, 2013

Abstract: In recent years, P2P (Peer-to-Peer) technique superior in load balancing, scalability, and robustness has been studied actively. P2P has applied to some services, such as file sharing, streaming delivery, IP phone and so on. Currently, these services are also used by mobile terminals such as cellular phone, smart phones and tablets. This is due to the high performance of mobile devices and development of wireless communication technologies such as LTE, 3G, and Wi-Fi. In addition, with the improvement of the diffusion rate of mobile terminals, it is expected that the demand for P2P use with the mobile terminals will increase still more in the future. In this paper, we propose the extension method of heterogeneous DHT algorithm in consideration of the existence of mobile terminals assuming the use of P2P under the Fixed Mobile Convergence (FMC) environment. By this method, it is possible to construct DHT maintaining search performance of conventional method while suppressing the communication processing load of mobile terminals which has high battery consumption. We evaluated the performance of the proposed method by simulation. As a result, it found to be possible to improve the battery life of the mobile terminals compared with the existing method.

Keywords: P2P, DHT, FMC, hierarchical DHT, heterogeneous DHT

1. はじめに

ネットワーク通信技術の高性能化にともない、既存の物理ネットワーク上に仮想的なオーバーレイネットワークを構築し、エンドユーザのクライアント端末間で直接通信を行う P2P (Peer-to-Peer) 技術が注目されてきた。P2P は従来のクライアント・サーバ型のネットワークと比べ、サーバのような集中機構を持たないため、負荷分散やスケラビリティ、耐故障性に優れているといわれている。さらに、P2P 技術はこれまでファイル共有ソフトやストリーミング配信、IP 電話 (Skype, Einy, PeerCast など) など、多くのサービスに応用されてきている。一方、スマートフォンやタブレット端末など、モバイル端末自身の性能向上や、LTE (Long Term Evolution) や 3G, Wi-Fi に代表される無線通信技術の発達にともない、P2P のピアとなるノードが有線接続されたデスクトップ PC から、無線接続されたノート PC・モバイル端末へと移行していくことが見込まれる。また、モバイル端末の多様なセンサを用いるモバイルセンシングなどにおいても、センサ情報の集積時に P2P 技術が用いられるようになってきている。以上の点から、今後モバイル端末による P2P サービス利用の需要は高まっていくものと予想される。

しかし、P2P のモバイル端末による利用についてはいくつかの問題が指摘されている。P2P のオーバーレイネットワークを構成する各ピアの性能を考慮した場合、通信速度や安定性などの通信品質、また CPU の処理能力など、固定端末とモバイル端末間では大きな性能格差がある。さらに、オーバーレイネットワークの構築・維持処理やネットワーク上での探索処理は、モバイル端末にとって大きな負荷となり、バッテリー消費が激しくなるという問題がある [1]。このため、現状ではモバイル端末向け P2P サービスの展開は困難であることが指摘されている [2]。従来の P2P では、各端末が対等な立場でオーバーレイネットワークを構築し、それぞれがピアとしてすべて同様の処理を行っているが、この場合、P2P の性能は構成ピアの性能に大きく依存するために、端末性能の低いピアが探索処理などを行ってしまうことで、P2P 自体の性能低下を招いてしまう。そのため、すべてのピアが同様の処理を行うものから、各ピアの性能に合わせた処理を提供するために非対称構造や階層型構造を持たせる手法にシフトしている。

そこで、本論文では、固定・モバイル端末の混在 (Fixed Mobile Convergence, FMC) 環境における P2P の利用を

想定し、モバイル端末のバッテリー消費を抑えるために、計算資源が豊富な固定端末へモバイル端末の通信処理を委譲する非対称型 DHT アルゴリズムを提案する。この手法を適用することによって、モバイル端末においてバッテリー消費量の大きな通信処理負荷を抑制することができ、かつ従来の探索性能を維持した P2P ネットワークを構築するための DHT を構成することが可能となる。

2. 関連研究

DHT に関しては、モバイル端末の DHT 参加における負荷やバッテリー消費の分析研究がさかんに行われている。これらの研究は大別して以下の 3 つに分類できる。

2.1 モバイル端末のクライアント化

モバイル端末であるピアを Client-peer としてのみ動作させる手法は、モバイル端末のクライアント化と呼ばれる。Client-peer とは経路表を保持するものの、自身へ送られてくるメッセージに対する応答を返さないピアのことをいう。Hofstatter らは、モバイル端末にとって DHT に参加することは過負荷であり、また DHT の探索性能を著しく劣化させることを問題視し、モバイル端末を Client-peer としてのみ動作させることを提案している [19]。この研究では、固定端末は従来の DHT を構成 (すなわち Full-peer として動作) し、モバイル端末は Client-peer として任意の固定端末と接続している。たとえば、モバイル端末が DHT 内に value を put, get する場合は接続した固定端末を経由して行うことができる。この関係性を用い、Chord に対してモバイル端末クライアント化を行った Chordella を提案している。このようにすることで、モバイル端末の負荷は最小に抑えられ、かつ DHT の探索性能を劣化させることもない。また固定端末が複数のモバイル端末と接続することを許可することで、モバイル端末数が固定端末数を上回った場合でも稼働することが可能となる。

2.2 メッセージドロップ法

メッセージドロップ法は、すべての処理を行う Full-peer として稼働した際のバッテリー消費量を抑えるため、バッテリーに依存する値を用いて受信メッセージの返答を破棄する手法である [20]。この手法では Kademlia を用いて、モバイル端末を固定端末と同様に Full-peer として動作させ、モバイル端末側のみに端末のバッテリー残存量に依存した確率変数 P_{drop} を定義している。この P_{drop} によってモバイル端末が受信したメッセージを破棄するか、しないかを決定しており、 $P_{drop} = 1$ であれば Client-peer のような立場となり、 $P_{drop} = 0$ では Full-peer として動作することを意味する。 P_{drop} によって受信メッセージを破棄すると、そのメッセージを送信したピアは、メッセージ破棄したモバイル端末を故障ピアとして扱う。したがって、他ピ

¹ 公立はこだて未来大学システム情報科学部
School of Systems Information Science, Future University
Hakodate, Hakodate, Hokkaido 041-8655, Japan

² 公立はこだて未来大学大学院システム情報科学研究科
Graduate School of Systems Information Science, Future
University Hakodate, Hakodate, Hokkaido 041-8655, Japan

a) y-nakamr@fun.ac.jp

アの経路表からモバイル端末が削除されるため、モバイル端末へ送信されるメッセージが減少する。これにより受信メッセージ数だけではなく、中継や返答の際に発生する送信メッセージも削減できる。しかし、この手法は従来手法と比べて、探索遅延時間が若干増加することが評価実験によって明らかになっている。Kelenyiらは、モバイル端末がFull-peerとは異なる動作をすることでDHTの性能が劣化することは、不可避であると述べている[16]。また、これをふまえてすべてClient-peerとして動作したモバイル端末の台数が増加した場合に著しくDHTの性能を劣化させることを分析し、モバイル端末のクライアント化についてスケラビリティに問題があることを指摘している。

2.3 非対称型 DHT

田上らは、モバイル端末の問題として、大きな遅延と位置不安定性を取り上げ、固定端末からモバイル端末へ探索メッセージを中継させない非対称型 DHT を提案している[21]。位置不安定性とは、モバイル端末の移動によりアクセスポイントや IP アドレスが変化し、これによってオーバーレイネットワーク上での位置が変化すること意味している。この非対称型 DHT では、自分以外のモバイル端末に対してメッセージが届かないようなトポロジを形成する。したがって、すべてのピアは経路表内にモバイル端末を持たないこととなる。そこで、Chord の探索処理を探索処理と名前解決処理の 2 つに分けて、モバイル端末への到達性を確保している。まず、探索処理では通常の探索と同様にデータ名などから key を求め、key に対して探索を行う。この探索では結果として相手先の IP アドレスや value ではなく、value を持つピアの ID (以下、担当ピア ID) を得る。次に、名前解決処理では探索処理で得た担当ピア ID を用いて、 $\langle \text{key}, \text{value} \rangle$ の組合せを管理するピア (以下、担当ピア) の探索を行う。しかし、このような探索においても、経路表内にモバイル端末が存在しないため、モバイル端末への到達性を持たない。したがって、名前解決処理と探索処理によってモバイル端末の到達性を得るために、ピアの ID とピアの IP アドレスのペアからなるマッピングエントリというエントリを、そのピア自身がネットワークに参加、もしくは IP アドレスが変わるたびに作成する。そしてこのマッピングエントリを自らの先祖ピア、子孫ピアに複製し、このマッピングエントリを利用して到達性を得る。このときの先祖ピアと子孫ピアは非対称型 DHT のトポロジにより、固定端末であることが確保されているため、探索処理を固定端末のみで行うことが可能となる。この非対称型 DHT では、探索処理を固定端末で行う分、探索遅延時間を抑えることができ、モバイル端末の中継処理負荷を削減することができる。しかし、探索処理を 2 つのフェーズに分けて行うことで、通常の倍の探索トラフィックがかかってしまう点が問題となる。

3. 提案手法

3.1 アプローチ

既存研究では、単純にモバイル端末側の処理のみを削減する方針が多くみられる。これはモバイル端末が Full-peer として参加することが過負荷であることから、処理内容を減らす方針をとっているものと考えられる。本論文では P2P ネットワーク上における固定端末という豊富な計算資源に着目し、固定端末の計算資源を有効活用することでモバイル端末側の処理を減らして両端末が協調できるネットワークを構築するアプローチをとる。具体的には、モバイル端末が行う探索処理や更新処理を固定端末側に委託する手法、およびモバイル端末への到達性を保ちつつ、モバイル端末の中継転送処理を回避するための迂回手法を提案する。また、従来手法では特定の DHT アルゴリズム (たとえば Chord や Kademia) に限られているうえに、モバイル端末の数が非常に少ない環境における評価が主流であった。そこで、提案手法は複数の DHT アルゴリズムに適用できることを示し、固定端末に対するモバイル端末の台数比が大きくなった場合にも問題なく動作することを示す。

3.2 迂回処理

提案手法ではベアリストを用いて、モバイル端末を迂回する処理を行う。ベアリストとは各ピアが持つ経路表とは別に、モバイル端末と固定端末を 1 つのペアとして保持するテーブルである。このテーブル中には複数のモバイル端末の ID および IP アドレスと、そのモバイル端末の ID に最も近い ID を持つ固定端末の ID と IP アドレスが結び付けられて記載されている。図 1 に迂回処理の概要を示す。

DHT ではピアを hop-by-hop で探索するため、探索処理においてすべてのピアでこの迂回処理が行われる可能性がある。まず、メッセージを送信するピアは自らの経路表を利用して、次に転送するピアとして探索対象の key に近いピア候補 (次ホップピア候補) を選択する。次ホップピア候補としてモバイル端末が選択された場合はベアリストの参照処理を行う。ここでは作成したベアリストから次ホップピア候補となっているモバイル端末を参照し、そのペアである固定端末を次ホップピア候補として追加する。固定端末が次ホップピア候補として追加されている場合、モバイル端末ではなく固定端末の方にメッセージを転送する。一方、経路表から次ホップピア候補として固定端末が選択された場合、ベアリストの参照を行わずに送信処理を開始する。しかし、このようにした場合、担当ピアであるモバイル端末へのメッセージの到達性が失われてしまう可能性がある。よって、ベアリストからの参照処理を行う場合にペアの固定端末をすべて優先的に取り出すのではなく、固定端末の ID とモバイル端末の ID、そして key の ID (key ID) を比較することによってモバイル端末が担当ピアであ

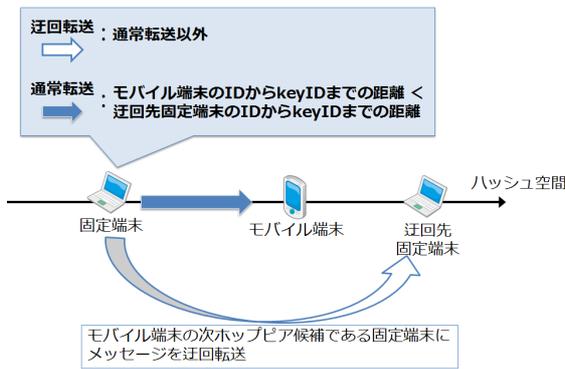


図 1 迂回処理

Fig. 1 By-pass process.

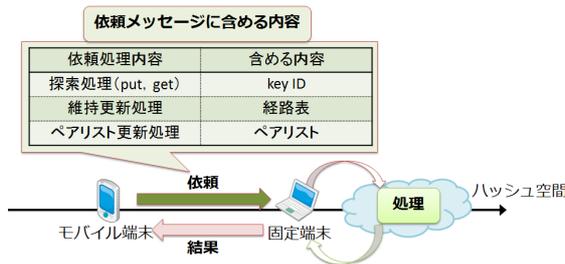


図 2 依頼処理

Fig. 2 Commission process.

る可能性があるかどうかの判断を行い、参照処理を行う。この担当ピアの判定は、利用する DHT アルゴリズムによって異なる。たとえば、Chord の場合、ある key の ID における子孫に相当するピアがこの key に対応する value を管理する。提案手法では、この判定を簡潔に行うため、探索対象である key ID を基点として固定端末の ID、モバイル端末の ID それぞれに対する距離 $distance(to, from)$ を計算し、key とモバイル端末との距離の方が短い場合は、モバイル端末が担当ピアである可能性が高いとしてモバイル端末にメッセージを送信する。この判定が誤っていた場合、担当ピア候補と見なされたモバイル端末はさらに迂回処理を実行し、新たな担当ピア候補のモバイル端末を見つけて通常転送を行うか、迂回先の固定端末に迂回転送を行う。これにより、モバイル端末が不必要に中継ピアとなることを回避しながら、モバイル端末への到達性を確保できる。

3.3 依頼処理

依頼処理ではモバイル端末が行う探索リクエスト (put, get) と経路表の維持更新処理を、固定端末に対して依頼し、モバイル端末の代わりに固定端末がその処理を行う。図 2 は依頼処理の概要である。提案手法において、モバイル端末が固定端末に依頼する処理は、探索処理と維持更新処理の 2 つである。

まず、探索処理の依頼方法について説明する。モバイル端末が探索リクエストを送信する場合、まず探索対象である key ID を記載した依頼メッセージを送信する。この依

頼メッセージを経路表内で自身に最も近い ID を持つ固定端末に送信する。これは、探索リクエストを送信したモバイル端末に最も近い ID を持つ固定端末に依頼メッセージを送信することで、探索対象である key ID を持つピアが該当モバイル端末の直近にあった場合でも、他の固定端末への経路など、探索にかかる処理コストを増加させずに依頼処理の効果を最大限に生かせるようにするためである。依頼メッセージを受信した固定端末が、key ID に対して探索処理を実行する。探索処理が完了した場合、処理が終了したことを依頼元のモバイル端末に返答する。この返答メッセージには value や担当ピアのアドレスが含まれる。

次に、維持更新処理の依頼方法について述べる。まず、更新処理を行うモバイル端末は自身の経路表の情報を依頼メッセージとして固定端末に送信する。このとき、送信先となる固定端末は先ほどと同様、自身に最も ID が近い固定端末が選ばれる。この理由としては ID の近いピアどうしの経路表の内容が類似しており、1 回の処理でモバイル端末と固定端末両方の経路表の内容を更新できる可能性が高いためである [21]。依頼メッセージを受信した固定端末は、受信した経路表のピアに対して更新メッセージを送信する。ここでピアの参加や離脱を検知した場合、離脱の検知であれば、そのピアを経路表から削除し、新規のピア参加であれば、そのピアを追加する。このように更新を終えた経路表を依頼元のモバイル端末に送信して、経路表更新の依頼処理は完了となる。

これによって、探索処理や維持更新処理のコストをすべて 1 回の送受信で済ませることが可能となる。また、提案手法ではピアの端末種別を示す識別子を用意する。この識別子を経路表に追加されるときに同時に保持させ、通信相手のピアが固定端末であるかモバイル端末であるかを瞬時に識別可能とする。

4. 提案手法の評価

4.1 実験環境

オーバーレイ構築ツールキットである Overlay Weaver を利用する [12], [13]。Overlay Weaver を構成するコンポーネント群を図 3 に示す。Overlay Weaver は Java 言語で構成されており、1 台のコンピュータ上に仮想的なノードを数十万ノードの規模でシミュレーションを行える。また対応する DHT アルゴリズムとして、Chord, Kademlia, Pastry, Tapestry, Koorde, FRT [14] がある。

本実験は Chord と Kademlia を比較対象としてシミュレーションを行った。また、Overlay Weaver ではモバイル端末のようなピアを想定していないため、バッテリーの消費量を表現することができない。文献 [17] では、バッテリー残存時間 $T_{battery}$ を、バッテリー容量と平均使用電力 P_{avg} を用いた数式で表現している。たとえばバッテリー容量が 0.950 Ah、動作電圧が V_{avg} である場合、以下の式のよう

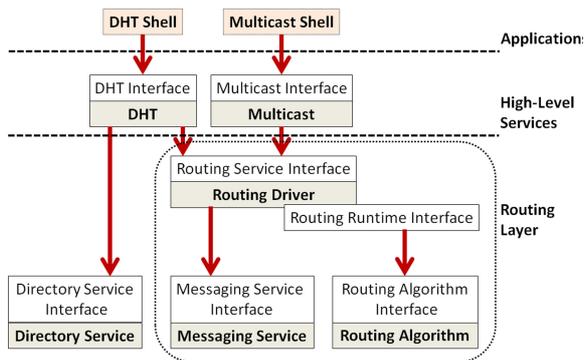


図 3 Overlay Weaver の構成コンポーネント群
Fig. 3 Components of Overlay Weaver.

表 1 シミュレーション環境
Table 1 Environment of simulation.

総ピア数 (N)	200, 2,000
固定端末・モバイル端末比	8:2, 5:5
固定端末の 1 ホップ通信遅延時間 (t_F)	1 msec
モバイル端末の 1 ホップ通信遅延時間 (t_M)	5 msec
バッテリー容量	0.950 Ah
動作電圧 (V_{avg})	3.7 V
送信処理の使用電力 (P_s)	1.1 W
受信処理の使用電力 (P_r)	0.2 W
ペアリスト更新間隔	100 sec

に表される.

$$T_{battery} = \frac{0.950 \text{ Ah} \times 60 \times \frac{\text{min}}{\text{h}} \times V_{avg}}{P_{avg}} [\text{min}] \quad (1)$$

そこで、式 (1) に従ってバッテリー残存時間を表現するクラスを実装した。シミュレーション実験の環境を、表 1 に示す。総ピア数やバッテリーに関する電圧、使用電力などは関連研究 [16], [17], [20], [21] を参考に設定している。また、実験で用いる Chord, Kademia のパラメータ設定は Overlay Weaver のデフォルト設定を用いる。一方、DHT 上で用いられる探索方式として Recursive 探索および Iterative 探索が知られている [22]。Recursive 探索はデータの要求元ノードが、他のノードに対して探索経路に沿って再帰的に探索を依頼していく方式であり、Iterative 探索は要求元ノードが探索経路上の各ノードに対して自ら問合せを行うことで、探索を実現する方式である。Chord がどちらの探索方式にも対応しているのに対し、Kademia は Iterative 探索を前提とした設計であることから、Chord においても条件を一致させて探索方式の違いによる影響を取り除くため、本実験では Chord, Kademia 双方とも Iterative 探索のみを用いることとする。

4.2 シミュレーション実験による評価

代表的な DHT アルゴリズムである Chord および Kademia について、提案手法を適用した場合と提案手法を適用しない場合とを比較することで、モバイル端末の大きな遅

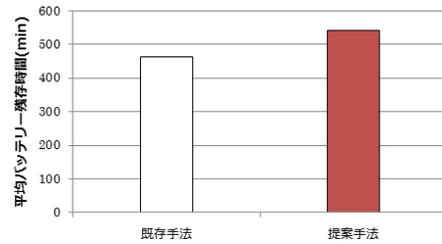


図 4 Chord での平均バッテリー残存時間 (ピア数 200, 固定・モバイル比 8:2)
Fig. 4 Average battery life time in Chord (200 nodes, fixed:mobile=8:2).

延を回避し、バッテリー消費量が抑えられていることを示すために、ピアごとの送受信回数からバッテリー残存時間を算出し、各端末の探索処理時の遅延時間を測定する。探索遅延時間は要求元ピアから担当ピアへメッセージが到達するまでの時間とし、提案手法ではモバイル端末が固定端末へ依頼メッセージを送信する処理も含む。また、提案手法のような拡張を行ったとき、DHT の探索性能への悪影響についても分析を行った。この DHT の性能を評価する項目として、経路長と生成メッセージ数を測定した。生成メッセージ数は実行する各処理によって生成されたメッセージで、一部の返答メッセージを除いた Overlay Weaver のメッセージカウンタで計測されたメッセージ数を示す。

シミュレーションのシナリオとしては、まず設定したピア数がオーバーレイネットワークに参加し、その後 1 分のインターバルを設けている。これは参加した直後の状態では経路表が正しい情報ではない可能性が高いため、Stabilize 処理などによってネットワークを正常に保つ状態を作るためである。その後、参加したすべてのピアが 15 秒間隔で put 処理と get 処理を交互に行う。この交互の探索処理を 10 回繰り返し、合計で約 7 分のシミュレーションを行った。

提案手法の有効性を検証するために、提案手法を適用した Chord および Kademia (以下、提案手法) と適用していない Chord, Kademia (以下、既存手法) で比較実験を行った。

総ピア数 200 における固定・モバイル端末の比率 8:2 の環境下での平均バッテリー残存時間、要求元ピアが固定端末、モバイル端末それぞれのときの平均探索遅延時間をそれぞれ図 4 と図 5 に示し、探索処理の平均経路長とシミュレーションにおける平均総生成メッセージ数を表 2 に示す。平均経路長は既存手法と提案手法ではほぼ同等の結果が得られている。

次に、ピア数を 200 で固定し、固定・モバイル比率を 5:5 とした場合のシミュレーションを行った。これは DHT 内にモバイル端末が占める割合が比較的多い場合の影響を分析するために実験を行った。この環境下におけるバッテリー残存時間と探索遅延時間を図 6 と図 7 に示し、探索処理の平均経路長とシミュレーションにおける平均総生成メッ

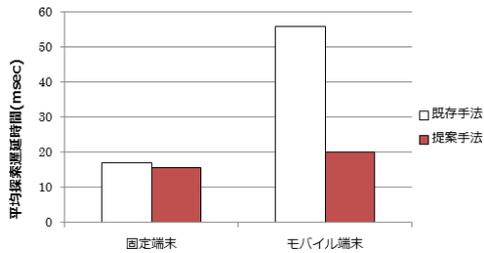


図 5 Chord での端末ごとの平均探索遅延時間 (ピア数 200, 固定・モバイル比 8:2)

Fig. 5 Average latency of each device in Chord (200 nodes, fixed : mobile=8 : 2).

表 2 Chord での平均経路長と平均総生成メッセージ数 (ピア数 200, 固定・モバイル比 8:2)

Table 2 Average path length and total created messages in Chord (200 nodes, fixed : mobile=8 : 2).

	既存手法	提案手法
平均経路長	3.79	4.29
平均総生成メッセージ数	32,261	41,327

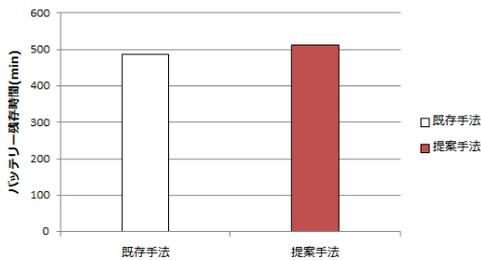


図 6 Chord での平均バッテリー残存時間 (ピア数 200, 固定・モバイル比 5:5)

Fig. 6 Average battery life time in Chord (200 nodes, fixed : mobile=5 : 5).

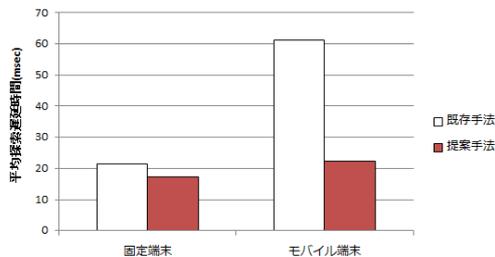


図 7 Chord での端末ごとの平均探索遅延時間 (ピア数 200, 固定・モバイル比 5:5)

Fig. 7 Average latency of each device in Chord (200 nodes, fixed : mobile=5 : 5).

メッセージ数を表 3 に示す。

さらに、固定・モバイル比率を 1 回目の実験と同様に 8:2 とし、ピア数を 2,000 とした場合のシミュレーションを行った。これは端末数が増加した場合の影響を分析することを目的としている。また、他の環境とはピア数が大きく異なるため、シミュレーション時間もピア数に合わせて延長している。この環境下におけるバッテリー残存時間と探

表 3 Chord での平均経路長と平均総生成メッセージ (ピア数 200, 固定・モバイル比 5:5)

Table 3 Average lookup length and total created messages in Chord (200 nodes, fixed : mobile=5 : 5).

	既存手法	提案手法
平均経路長	3.79	4.25
平均総生成メッセージ数	321,413	51,272

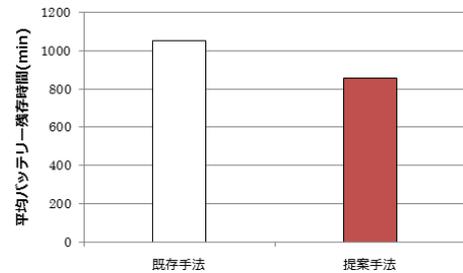


図 8 Chord での平均バッテリー残存時間 (ピア数 2,000, 固定・モバイル比 8:2)

Fig. 8 Average battery life time in Chord (2,000 nodes, fixed : mobile=8 : 2).

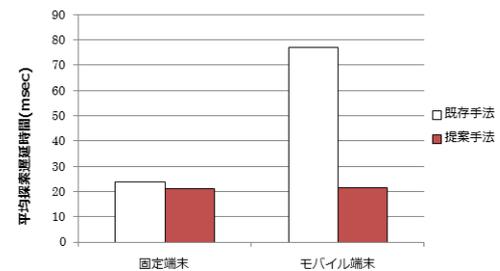


図 9 Chord での端末ごとの平均探索遅延時間 (ピア数 2,000, 固定・モバイル比 8:2)

Fig. 9 Average latency of each device in Chord (2,000 nodes, fixed : mobile=8 : 2).

表 4 Chord での平均経路長と平均総生成メッセージ (ピア数 2,000, 固定・モバイル比 8:2)

Table 4 Average lookup length and total created messages in Chord (2,000 nodes, fixed : mobile=8 : 2).

	既存手法	提案手法
平均経路長	5.59	6.08
平均総生成メッセージ数	871,029	1,084,513

索遅延時間を図 8 と図 9 に示し、探索処理の平均経路長とシミュレーションにおける平均総生成メッセージ数を表 4 に示す。

提案手法の拡張性を調べるために Kademia での動作確認を含めて簡単な比較実験を行った。シミュレーション環境は総ピア数 200, 固定・モバイル比率 8:2 の環境で、Kademia に提案手法を適用したものと既存の Kademia とで比較を行った。図 10 にバッテリー残存時間、図 11 に端末種ごとの探索遅延時間を示し、平均経路長や総生成メッセージ数、加えてモバイル端末の平均中継回数を表 5 に

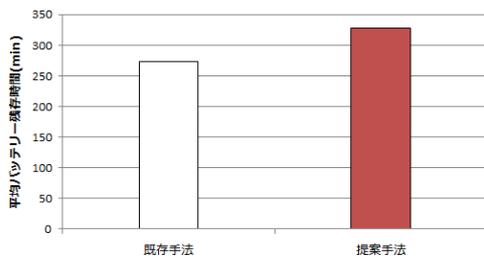


図 10 Kademia での平均バッテリー残存時間 (ピア数 200, 固定・モバイル比 8:2)

Fig. 10 Average battery life time in Kademia (200 nodes, fixed: mobile=8:2).

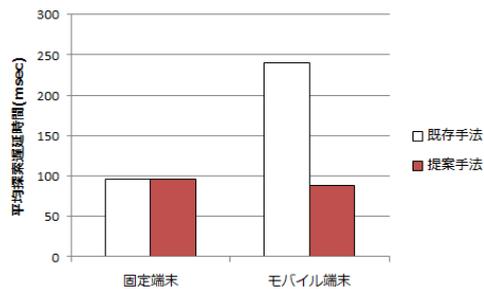


図 11 Kademia での端末ごとの平均探索遅延時間 (ピア数 200, 固定・モバイル比 8:2)

Fig. 11 Average latency of each device in Kademia (200 nodes, fixed: mobile=8:2).

表 5 Kademia での平均経路長と平均総生成メッセージ数 (ピア数 200, 固定・モバイル比 8:2)

Table 5 Average path length and total created messages in Kademia (200 nodes, fixed: mobile=8:2).

	既存手法	提案手法
平均経路長	12.31	13.33
平均総生成メッセージ数	88,388	120,114
平均中継回数	400	460

示す。

4.3 考察

4.3.1 総ピア数 200, 固定・モバイル端末の比率 8:2 における Chord の場合

図 4 と図 5 より, 提案手法において測定されたバッテリー残存時間は, 既存の Chord と比較して約 1.17 倍に向上し, 時間にして約 80 分長く持続している。これにより, モバイル端末の通信処理を削減し, バッテリー消費量を削減する目的は達成できているといえる。また探索遅延時間に関しては, 固定端末における結果は既存手法と提案手法で差異は少ないものの, 提案手法の方に若干の改善が見られた。一方, モバイル端末で測定された探索遅延時間は提案手法が既存手法の約 35% に削減できている。Iterative 探索の場合, 要求元ピアであるモバイル端末を必ず経由するために探索遅延が大きくなるが, 提案手法は依頼処理によってそれを防止している。

また, 表 2 より, 探索性能は従来の DHT の維持ができていることがいえる。さらに, 探索遅延時間の結果を併用すると, 提案手法は既存手法よりも FMC において効果的な探索を行っていることが分かる。しかし, 平均総生成メッセージ数では提案手法が約 9,000 増加している。これらのメッセージは, 提案手法で追加した依頼処理とペアリスト更新処理によるものである。

4.3.2 総ピア数 200, 固定・モバイル端末の比率 5:5 における Chord の場合

図 6 は図 4 の結果と比較すると, この環境ではバッテリー寿命の持続効果が減少していることが分かる。図 4 では既存手法よりも 80 分程度長く持続していたが, 30 分程度の持続に短縮されている。また図 7 より, 平均探索遅延時間は 8:2 の環境と比較した場合, 傾向は同等であるが, 全体的に探索遅延時間の増加が見られた。これはネットワーク内に存在するモバイル端末数が増加したことによる影響であると考えられる。

また, 表 3 より, 平均経路長は, 表 2 の結果とほぼ同等の結果を得た。これは経路長がピア数 N に依存する値であることから, 妥当な結果であると考えられる。一方で, 平均総生成メッセージ数は提案手法においての結果の倍以上増加している。したがって, モバイル端末の割合, もしくはモバイル端末数が増加することによって提案手法におけるメッセージ数は増加してしまい, 不利に働くことが考えられる。

4.3.3 総ピア数 2,000, 固定・モバイル端末の比率 8:2 における Chord の場合

図 8 と図 9 より, この環境では, 提案手法が既存手法のバッテリー残存時間を下回る結果となった。また平均探索遅延時間に関しては, 他の条件による実験結果と傾向は変化なく, 提案手法がモバイル端末の高い遅延を回避していることが分かる。

また, 表 4 より, この環境での平均経路長は, ピア数が増加したためにこれまでの実験結果と比べて大きくなっている。しかし, 既存手法と提案手法による差は小さく, 提案手法が従来の探索性能を維持できていることはこれまでと同様であることがいえる。また平均総生成メッセージ数は提案手法で 20 万メッセージ程度多く測定された。これは 200 ピア, 固定端末・モバイル端末の比率 5:5 での 2 万メッセージと同等のメッセージ増加比率となっている。この 2 つの環境の共通点はモバイル端末数が比較的多くなっていることであり, モバイル端末数が 100 を超えると問題が生じ始めることが分かる。モバイル端末数が増加するとペアリスト更新処理が増加するため, これがバッテリーの消費の大きな要因であると考えられる。これらのコストはペアリスト更新間隔を大きくすることで削減可能であるが, ピアの参加離脱を考慮した場合にペアリストの不整合が発生するなど, 問題が生じる可能性がある。

4.3.4 総ピア数 200, 固定・モバイル端末の比率 8:2 における Kademia の場合

図 10, 図 11 より, Kademia での測定実験においても, Chord の環境と同等の効果が得られていることが分かる. したがって, 提案手法の拡張方式を複数の DHT アルゴリズムに適用することは可能であり, かつ, 有効性を保っていることがいえる.

また, 表 5 より, Kademia の実験では, Chord と比較して経路長が長くなった. ここで, 中継回数が提案手法の方が多くなってしまっているにもかかわらず, 既存手法より提案手法の方がバッテリー残存時間は長いことにも注目する. この理由としては, 中継処理による処理数より, 経路長が長くなることによる探索処理の負荷が大きいためである. 通常の探索では, 経路長が長くなると, 要求元の負荷はそれだけ大きくなってしまふ. しかし, 提案手法では依頼処理によってこの負荷は必ず 1 回の通信処理分となるため, 経路長の長い Kademia においても小さい負荷で探索を行うことができた. また, 中継回数が増加してしまつた理由としては, 迂回処理における迂回の判断がスムーズに行うことができなかったことがあげられる. この原因は, Kademia の距離が XOR によって定義されている点にあると考えられる. Kademia では, 次ホップ候補は目的 ID と候補との距離を計算して決定する. たとえば, 要求元ピアの ID が ID_o であり, 次ホップ候補 x と y の ID がそれぞれ $ID_x \leq ID_o$, $ID_y \geq ID_o$ であった場合, 目的 ID が自分の ID より大きい (正方向) か, 自分の ID より小さい (負方向) といった判断で次ホップが決まる. このように Kademia において次ホップ候補は自ノードから見た目的 ID への方向によって決まる. 同様に, 迂回処理でもこのようなことが起きていて考えられる. 迂回処理においてペアとなる固定端末は最も ID が近いノードが選ばれる. このとき, 対象のモバイル端末の ID を ID_m , ペアの ID を ID_p とし, 仮に $ID_m \geq ID_p$ とする. ペアのピアが迂回先となる判断は, 図 1 により目的 ID と ID_p 間の距離が最も近い場合である. これが成立するときは, 少なくとも目的 ID が ID_o と ID_m の間に存在し, かつ ID_p 間との距離が最も近い必要があるため, 迂回先として選択される可能性が非常に低い. このため, 提案手法における迂回処理の判断設定やペアノードの選定は, 使用する DHT アルゴリズムに合わせて調整する必要があると考えられる.

4.3.5 考察のまとめ

これらの実験結果により, 提案手法は固定・モバイル端末混在環境において, モバイル端末の高遅延を回避し, どのような環境下でも遅延時間の短い探索処理を行うことができる手法であることがいえる. バッテリー消費量に関しては DHT の環境に依存し, 全体のモバイル端末数が増加した場合にはモバイル端末側の負荷も大きくしてしまっていることが考えられる.

5. おわりに

本論文では, P2P における検索技術である DHT においてモバイル端末の高遅延や通信の不安定性, バッテリーといった特性を考慮したモバイル端末に向けた DHT の応用を目的とし, バッテリー消費量の削減に対してモバイル端末が DHT において行う通信処理を固定端末に委譲する依頼処理とモバイル端末に余分なメッセージが送信されることを回避する迂回処理を行う DHT アルゴリズムの提案を行った. シミュレーション実験により, モバイル端末における通信処理の削減が確認でき, 探索処理にモバイル端末を極力参加させないことで, 探索遅延時間の改善が見られた.

今後の課題としては, オーバヘッドとなつてしまつたベアリスト更新処理の改善が考えられる. これについては定期的な更新処理ではなく, イベントトリガの処理やピアどうしによる情報交換の手法が考えられる. また, 迂回処理の判定基準の見直しも必要であると考えられる.

参考文献

- [1] Zoels, S., Schubert, S., Kellerer, W. and Despotovic, Z.: Hybrid DHT Design for Mobile Environments, *Agents and Peer-to-Peer Computing*, pp.19-30 (2008).
- [2] Junfeng, X., Zhenhua, L. and Guihai, C.: On Maximum Stability with Enhanced Scalability in High-Churn DHT Deployment, *Proc. 2009 International Conference on Parallel Processing (ICPP '09)*, pp.502-509 (2009).
- [3] 川田量久, 植田和憲: ノード性能を考慮した非対称 DHT の提案, 電子情報通信学会技術研究報告, Vol.108, No.481 (ICM2008 58-90), pp.69-74 (2009).
- [4] 新崎裕隆, 上田真太郎, 真下 洋, 重野 寛: 構造化オーバーレイにおける転送妨害攻撃の影響に関する一考察, 情報処理学会研究報告, Vol.2007, No.117 (DPS-133), pp.49-54 (2007).
- [5] 乾 広二, 鈴木朋子, 大下福仁, 角川裕次, 増澤利光: 構造化オーバーレイネットワークにおける故障耐性向上のための経路多重化法, 電子情報通信学会論文誌 D, Vol.J91-D, No.5, pp.1261-1274 (2008).
- [6] Garcés-Erice, L., Biersack, E.W., Felber, P.A., Ross, K.W. and Urvoy-Keller, G.: Hierarchical Peer-to-peer Systems, *Proc. 9th International Euro-Par Conference*, pp.1230-1239 (2003).
- [7] Zoels, S., Despotovic, Z. and Kellerer, W.: On Hierarchical DHT System - An Analytical Approach for Optimal Designs, *Computer Communications*, Vol.31, pp.576-590 (2008).
- [8] 羽場裕介, 松尾啓志: 物理ネットワークの状況を考慮した階層型分散ハッシュ法の提案, 情報処理学会研究報告, Vol.2006, No.14 (MBL-36), pp.43-48 (2006).
- [9] 金子 雄, 福村真哉, 春本 要, 下條真司, 西尾章治郎: ユビキタス環境における端末の位置情報に基づく P2P ネットワーク, 情報処理学会論文誌: データベース, Vol.46, No.SIG18 (TOD28), pp.1-15 (2005).
- [10] 金子 雄, 福村真哉, 春本 要, 下條真司, 西尾章治郎: モバイル環境における端末の位置情報に基づく P2P ネットワークの提案と評価, 電子情報通信学会第 15 回データ工学ワークショップ (DEWS2004) 論文集, 1-A-01 (2004).
- [11] 佐藤文明: 参加脱退率を考慮した階層的 DHT システムの

構成方式, 情報処理学会論文誌, Vol.51, No.2, pp.418-428 (2010).

- [12] 首藤一幸: Overlay Weaver: An Overlay Construction Toolkit, 入手先 (<http://overlayweaver.sourceforge.net/index-j.html>).
- [13] 首藤一幸, 田中良夫, 関口智嗣: オーバレイ構築ツールキット Overlay Weaver, 情報処理学会論文誌: コンピューティングシステム, Vol.47, No.SIG12 (ACS15) pp.358-367 (2006).
- [14] Nagao, H. and Shudo, K.: Flexible Routing Tables: Designing Algorithms for Overlays Based on a Total Order on a Routing Table Set, *Proc. 2011 IEEE International Conference on Peer-to-Peer Computing (IEEE P2P'11)*, pp.72-81 (2011).
- [15] Stutzbach, D. and Rejaje, R.: Improving Lookup Performance over a Widely-Deployed DHT, *Proc. 25th IEEE International Conference on Computer Communications (INFOCOM 2006)*, pp.1-12 (2006).
- [16] Kelenyi, I. and Nurminen, J.: Energy Aspects of Peer Cooperation - Measurements with a Mobile DHT System, *Proc. IEEE Cognitive and Cooperative Wireless Networks (CoCoNet) Workshop 2008 Collocated with the 43rd IEEE International Conference on Communications (IEEE ICC 2008)*, pp.164-168 (2008).
- [17] Kassinen, O., Harjula, E., Korhonen, J. and Ylianttila, M.: Battery Life of Mobile Peers with UMTS and WLAN in a Kademia-based P2P Overlay, *Proc. IEEE 20th International Symposium on Personal, Indoor and Mobile Radio Communications*, pp.662-665 (2009).
- [18] Zhonghong, O., Harjula, E., Kassinen, O. and Ylianttila, M.: Performance Evaluation of a Kademia-based Communication-oriented P2P System under Churn, *Computer Networks: The International Journal of Computer and Telecommunications Networking*, Vol.54, No.5, pp.689-705 (2010).
- [19] Hofstatter, Q., Zoels, S., Michel, M. and Despotovic, Z.: Chordella - A Hierarchical Peer-to-Peer Overlay Implementation for Heterogeneous, Mobile Environments, *Proc. 8th International Conference on Peer-to-Peer Computing (P2P'08)*, pp.75-76 (2008).
- [20] Imre, K. and Nurminen, J.: Optimizing Energy Consumption of Mobile Nodes in Heterogeneous Kademia-based Distributed Hash Tables, *Proc. 2nd International Conference on Next Generation Mobile Applications, Services (NGMAST'08)*, pp.70-75 (2008).
- [21] 田上敦士, 阿野茂浩: 固定・モバイル混在環境を考慮した非対称型 P2P ネットワークの設計と実装, マルチメディア, 分散, 協調とモバイルシンポジウム (DICOMO), Vol.2011, pp.121-126 (2011).
- [22] 首藤一幸, 加藤大志, 門林雄基, 土井裕介: 構造化オーバレイにおける反復探索と再帰探索の比較, 情報処理学会研究報告, Vol.2006, No.86 (OS-103), pp.9-16 (2006).



中村 嘉隆 (正会員)

2002年大阪大学基礎工学部情報科学科卒業。2007年同大学大学院情報科学研究科博士後期課程修了。同年奈良先端科学技術大学院大学情報科学研究科助教。2010年大阪大学大学院情報科学研究科特任助教。2011年より公立はこだて未来大学システム情報科学部助教。博士(情報科学)。センサネットワークセキュリティに関する研究に従事。IEEE 会員。



舟橋 知論

2011年公立はこだて未来大学システム情報科学部情報アーキテクチャ学科卒業。2013年同大学大学院システム情報科学研究科システム情報科学専攻博士前期課程修了。在学中はP2Pネットワーク、構造型オーバレイネットワーク、非対称型DHTとFMC環境に関する研究に従事。現在、KDDI株式会社勤務。



白石 陽 (正会員)

2002年慶應義塾大学大学院理工学研究科博士課程単位取得退学。東京大学空間情報科学研究センター研究機関研究員, 助手, 助教を経て, 2009年より公立はこだて未来大学助教, 2011年より同大学准教授。博士(工学)。データベース, 高度交通システム, 都市センシング, 地理情報システムに関する研究に従事。電子情報通信学会, 地理情報システム学会, ACM等会員。



高橋 修 (フェロー)

1975年北海道大学大学院工学研究科修士課程修了。同年電電公社(現, NTT)横須賀電気通信研究所入所。コンピュータネットワークの研究・開発・標準化に従事。NTTドコモを経て2004年より公立はこだて未来大学教授。博士(工学)。本会業績賞。本会フェロー。電子情報通信学会, IEEE各会員。