

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

田上 敦士^{†1} 阿野 茂浩^{†1}

近年、移動網の広帯域化やモバイル端末の高性能化により、固定網・移動網が混在した環境が広く利用されている。ユーザは、固定網・移動網を意識せずに利用する事を求めるが、混在環境において peer-to-peer (P2P) 技術を利用した場合いくつかの問題が生じる。これらの問題は主に、伝送遅延の増大や IP アドレスの不安定性等の移動網の特性に起因する。これらの問題に対して、検索をサーバで行うハイブリッド型 P2P 技術や、スーパーノードを用いた階層型 P2P 技術が提案されている。しかしながら、これらはサーバやスーパーノードなど負荷が集中するピアが存在するという課題がある。筆者らは、これらの問題を解決するため、非対称型 P2P ネットワークの設計と実装を行った。本アプリケーションは各ピアがそれぞれ異なる最適なトポロジを形成する事により、モバイル端末が混在したネットワークにおいても、効率的な検索を実現するものである。本稿ではその設計と実装について述べる。

Design and Implement of Asymmetric P2P Network with Fixed Mobile Convergence Network

ATSUSHI TAGAMI^{†1} and SHIGEHIRO ANO^{†1}

Recently, wireless bandwidth has grown and there has been an improvement in mobile device performance. This growth and improvement have facilitated a convergence between fixed and mobile networks. However, this environment has some problems in the application of peer-to-peer (P2P) architecture. These problems are caused by the mobile peers' disadvantages, such as the transmission latency and location instability. To solve these problems, we propose an asymmetric P2P architecture which is based on a distributed hash table (DHT). Each peer in the architecture has a different topology which does not include bottleneck peers. The architecture does not have a peer with specific features and this ensures scalability and robustness. This paper shows its design and implement.

1. はじめに

通信のユビキタス化に伴い、光ファイバーや CATV 網を使用した広帯域通信、Wi-Fi や WiMAX などの高速モバイル通信が発展してきた。また、PC やモバイル端末機器自体も性能向上しており、端末が単に情報を受信するだけでなく、情報の管理・配信を行う能動的な端末として動作できる状況にある。このため、近年ではクライアント端末間でオーバーレイネットワークを構築し、情報の共有・配信を行う P2P 技術が注目されている¹⁾。P2P 技術は、処理負荷や通信トラフィックが集中するノードを持たず、大規模な情報配信・共有に向いていると言われている。そのため、PC だけでなくモバイル端末までその活用範囲は広がっている²⁾。今後、スマートフォンなど高機能なモバイル端末が普及するに従い、固定網とモバイル網は融合が進み、固定・モバイル混在環境での P2P 技術の必要性は高まって行くと思われる。

モバイル端末に対して P2P 技術を適用する場合、2つの問題が考えられる。1つ目は検索処理など短い処理完了時間が求められる処理において、無線回線による伝送遅延がボトルネックとなる点である。P2P 技術ではピア間の hop-by-hop 通信により検索処理を実現するため、伝送遅延の大きいピアが中継ノードとなった場合、処理完了時間が大きくなる。2つ目は、移動による回線断などにより頻繁にモバイル端末の IP アドレスが変更される点である。P2P 技術では、ピア間で直接通信を行うためピアの IP アドレスの変更を頻繁に P2P ネットワーク上に通知する必要がある。これらの問題に対して、検索をサーバで行うハイブリッド型 P2P 技術³⁾ や、スーパーノードを用いた階層型 P2P 技術が提案されている⁴⁾。しかしながら、これらはサーバやスーパーノードなど特別な機能を持つノードを必要とするため、スケーラビリティや公平性に課題がある。

これらの問題を解決するために我々は、非対称型 P2P ネットワークを提案する。本ネットワークは DHT (Distributed Hash Table)⁵⁾ をベースとし、ピアから見える P2P ネットワークトポロジが、ピア毎に異なる事を特徴としている。モバイル端末などボトルネックとなるピアを、他ピアから見たトポロジから除外する事により、処理完了時間に敏感な処理の完了時間増大を回避する。また、スーパーノードの様に特定の機能を持つピアを作らないため、負荷の集中を抑制できる。

^{†1} KDDI 研究所
KDDI R&D Laboratories, Inc.

一方、サーバやスーパーノードなどの特定の機能を持つピアが存在しないため、IP アドレスの変更についても分散的に管理する必要がある。そのため、提案ネットワークでは、P2P ネットワーク自体でピアの ID と IP アドレスのペアを管理する。ネットワークトポロジの非対称性により、頻繁に IP アドレスが変化するピアは他ピアから見たネットワークトポロジには含まれない。しかしながら、ファイル取得処理などで必要な際には、ID をキーとして検索する事でいつでも IP アドレスを取得する事が可能となる。

筆者らは、写真の共有アプリケーションを一例として、非対称型 P2P ネットワークの設計と実装を行った。本アプリケーションはモバイル端末で撮影した写真を P2P ネットワーク上で共有する。ユーザは、キーワードにより写真を検索し、写真を所持しているモバイル端末から直接その写真を取得する事ができる。ベースとなる P2P 技術としては、DHT の 1 つである Chord⁶⁾ を用いた。本稿では、その設計の詳細と実機による評価について示す。

2. 既存技術

2.1 モバイル端末向け P2P 技術

モバイル端末を対象として P2P 技術は、地理的に近い端末間で直接通信を行う Ad-hoc 通信を利用するものと、利用しないものの 2 つに分類される。MANET (Mobile Ad-hoc Network)⁷⁾ は、Ad-hoc 通信によりマルチホップ無線ネットワークを構築する経路制御プロトコルの 1 つであり、その上で動作する P2P 技術が提案されている^{8),9)}。また、Luo ら¹⁰⁾ は、Ad-hoc ネットワークを利用して移動網のスループットを向上させる手法を提案している。これらに対して本稿では、既に存在する Ad-hoc 通信を利用していない P2P ネットワークに対して、移動網を用いた端末が混在した時の影響について考察する。

Ekler ら²⁾ は、BitTorrent をモバイル端末に実装している。BitTorrent はサーバの負荷を軽減するために、クライアント間通信を利用するファイルダウンロードプロトコルである。しかしながら、BitTorrent は検索のための手順を持たず、検索は他のプロトコルを利用する必要がある。一方本稿では、検索などの処理完了時間に敏感な処理について議論する。

吉田ら³⁾ は、モバイル端末に対するゲーム向け P2P 通信ミドルウェアを開発している。本技術は、マッチングサーバで接続相手を検索した後、直接端末同士が通信を行う。Peng ら⁴⁾ は、CAN (Content-Addressable Network)¹¹⁾ を移動網向けに拡張した。本技術は、ピアの中から安定し性能の高いスーパーノードを選定し、そのノードを用いて検索を行う。これらの技術は、サーバやスーパーノードといった特別な機能を持つノードが存在するため、スケラビリティや耐障害性の面で課題がある。そこで本稿では、特別な機能を持ったノード

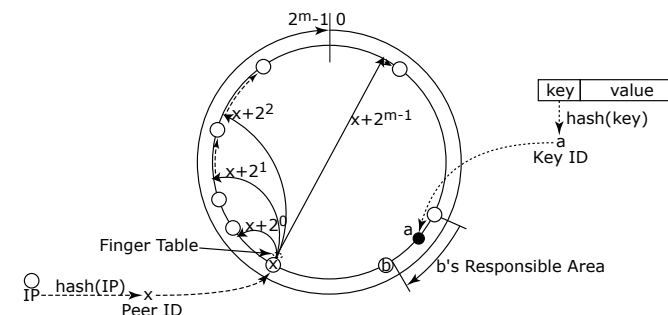


図 1 Chord の構造
Fig.1 Structure of Chord

が存在しない P2P ネットワークを提案する。

2.2 分散ハッシュテーブル

分散ハッシュテーブル (DHT, Distributed Hash Table)⁵⁾ は、膨大な数のピアに分散して蓄積されたエントリを、効率よく挿入・検索するプロトコルである。エントリとは DHT が保持するデータの基本要素であり、キーと値から構成される。ファイル共有アプリケーションの場合、キーにはファイルの内容を示すタイトル、値にはファイルを保持しているピアの IP アドレスが保持される。

Chord⁶⁾ は、広く利用されている DHT の実装の 1 つである。図 1 に Chord の構造を示す。Chord に参加するピアは 0 から $2^m - 1$ までの m ビットの ID (ピア ID) を持ち、1 つの円上に配置される。各エントリは、そのキーをハッシュ関数を用いて ID (キー ID) に変換し、ピアと同じ円上に配置される。Chord はハッシュ関数として SHA-1¹²⁾ を利用することが多く、この時 $m = 160$ となる。ある ID に対して、反時計回りに最も近い位置にあるピアを *predecessor*、時計回りに最も近い位置にあるピアを *successor* と呼ぶ。エントリはキー ID の *successor* に保持される。すなわち、ピアは自 ID の *predecessor* と自ピアとの間に配置されたエントリを保持する。さらに、ピアは耐障害性を確保するため、自 ID の *predecessor* と *successor* に対してエントリの複製を配置する。

Chord は Finger Table を用いる事により、 $O(\log n)$ のコストでエントリの挿入・検索を行う。Finger Table とは隣接ピアテーブルの事であり、自 ID から $2^0, 2^1, 2^2, \dots, 2^{m-1}$ を加えた値の *successor* の IP アドレスを保持している。また、Finger Table を保持するため、各ピアは自 ID の *successor* と *predecessor* に対して、定期的に Keep-Alive メッセージを投

げ、その生存を確認する。

3. 非対称型 P2P ネットワーク

3.1 設計方針

近年の無線アクセス回線の高速化やモバイル端末の高性能化は、固定網とモバイル網の垣根を低くしている。固定・モバイル網混在環境において、ユーザは固定網かモバイル網かを意識せずにアプリケーションの利用ができる事が望まれる。しかしながら、固定・モバイル混在環境下で P2P ネットワークを構築した場合、モバイル網に接続されたピア（モバイルピア）は下記のような問題を有している。

高伝送遅延 無線アクセス回線の伝送遅延は大きく、モバイルピアはメッセージの転送に数百 ms 必要とする。hop-by-hop 通信において中継にかかる時間が増加することにより、処理完了時間が長くなる。

位置不安定性 モバイルピアは、利用する無線アクセス回線種別の変更や、移動に伴うアクセスポイントの切り替えにより、IP アドレスが頻繁に変更される。このため、ピア間での通信時に IP アドレスが変化しており通信できない場合がある。

これらの問題を解決するために、我々は非対称型 P2P ネットワークを提案する。本ネットワークは以下の特徴を持つ。

トポロジの非対称性 各ピアから見たトポロジを、そのピアに最適なトポロジとする。これにより、検索処理などの処理完了時間がユーザビリティに大きく影響を与える処理において、モバイルピアの伝送遅延の影響を避ける事ができる。

ID/IP アドレスのマッピング ID と IP アドレスのマッピングを、P2P ネットワーク自体で管理する。これにより、モバイルピアの IP アドレス変更を P2P ネットワーク自体で追跡する事ができる。

本ネットワークは DHT⁵⁾ をベースとする事を想定しており、特に DHT の種類には依存しないが、本稿では簡単のため Chord⁶⁾ をベースとして説明する。次節より、非対称型 P2P ネットワークの構築処理と検索処理について詳細を述べる。

3.2 ネットワーク構築処理

図 2 に非対称型 P2P ネットワークの例を示す。通常、それぞれのピアは、P2P ネットワークに参加しているすべてのピアに対して到達性を持つ。しかしながら、非対称型 P2P ネットワークでは、それぞれのピアは自ピア以外のモバイルピアに対して到達性を持たない様にトポロジを構築する。これにより、検索処理など、処理完了時間の影響が大きい処理

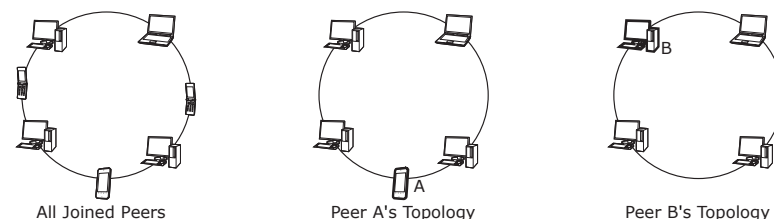


図 2 非対称型 P2P ネットワークの例
Fig. 2 Example of Asymmetric P2P Network

にモバイルピアを中継ノードとして参加させない事ができる。さらに、各トポロジにおいて、自ピア以外のピアは IP アドレスが頻繁に変更されず、隣接ピアテーブルの保守も容易となる。

非対称型ネットワークは、中継ノードとしてモバイルピアが選択されない。これはメッセージの次ホップとしてモバイルピアが選択されない事を意味する。すなわち、各ピアの隣接ピアテーブル (Finger Table) からモバイルピアを除外する事で、非対称型ネットワークを構築できる。Finger Table からモバイルピアを除外する方法は、隣接ピア間での品質計測など自律分散的に実施する方法も考えられるが、本稿ではピアの参加時に自ピアの参加を他ピアに広報しない自己申告による構築を用いる。

Chord において P2P ネットワークへの参加処理は大きく、

- (1) 自ピアの Chord の円上での位置を検索する。
- (2) 自ピアの参加を他ピアに通知する。
- (3) successor より Finger Table とエントリを複製する。

という処理からなる⁶⁾。この時、(2) の通知処理を省く事により、自ピアの参加を他ピアに通知することなしに、P2P ネットワークに参加する事ができる。

3.3 検索処理

図 3 に非対称型 P2P ネットワークにおけるファイル検索処理手順を示す。検索処理手順は、目的とするファイルを所持するピアを取得する探索処理と、ピアの IP アドレスを取得する名前解決処理から構成される。まず、ピアは通常の Chord と同様に目的とするファイルの名前をキーとして、エントリを検索する (探索処理)。この時、エントリの値としてはファイルを所持するピアの IP アドレスではなく、ピア ID が返される。その後、ピアは探索処理で得られたピア ID をキーとして、エントリを検索する (名前解決処理)。この時、

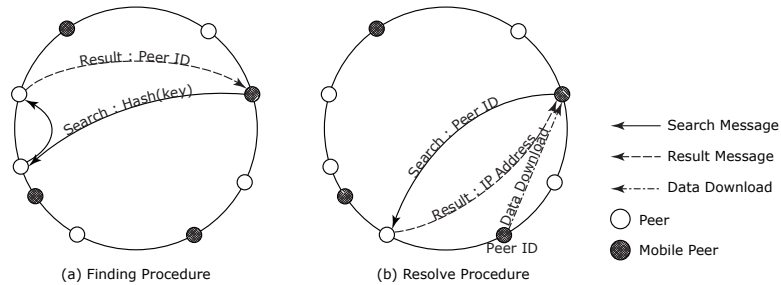


図3 ファイル検索処理手順
Fig. 3 File Search Procedure

エントリの値としては、ピア ID で示されるピアの IP アドレスが返される。ピアは取得した IP アドレスにより、直接ファイルを所持するピアと通信し、目的のファイルを取得する。

各ピアは、名前解決処理を実現するために、ピア ID と IP アドレスからなるマッピングエントリを定期的もしくは IP アドレスが変更される毎に P2P ネットワークに挿入・更新する必要がある。本更新処理を軽量化するために、マッピングエントリのエントリ ID を、キーであるピア ID のハッシュ値ではなく、ピア ID そのものとする。これにより、マッピングエントリはエントリを生成するピア自身で保持される。しかしながら、ネットワークの非対称性により他ピアからモバイルピアに対する到達性がないため、モバイルピアに保持されているマッピングエントリを、他ピアが取得する事はできない。これを解決するため、エントリの複製処理を利用する。すなわち、マッピングエントリが挿入・更新される毎に、ピアの successor と predecessor にエントリが複製される。前述のネットワーク構築手順により、隣接ピアである successor と predecessor はモバイルピアでない事が保証されているため、他ピアからの到達性が保証される。さらに、定期的なマッピングエントリを更新する場合、Keep-Alive 処理と兼用する事により、最小限のメッセージでマッピングエントリの保守が可能となる。

なお、モバイルピアはエントリの挿入・更新処理にも参加しないため、モバイルピアが保持するエントリは常に最新のものである保証はない。このため、定期的な複製処理でエントリを更新するか、自ピアが応答すべき検索メッセージを常に successor に転送するといった処理が必要となる。

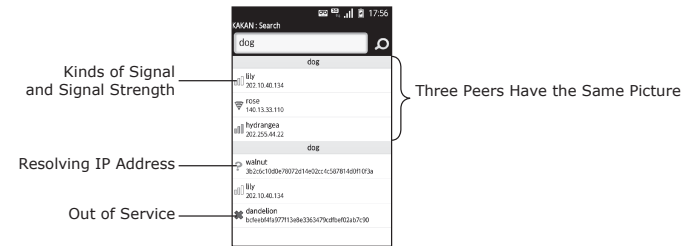


図4 検索結果の例
Fig. 4 Example of Search Results

4. 実 装

4.1 写真共有アプリケーション

非対称型 P2P ネットワークの応用例として、写真共有アプリケーションを作成した。本アプリケーションは、各端末で撮影した写真に付けられたタイトルをキーとして、非対称型 P2P ネットワークにより写真を管理する。ユーザはタイトルをキーとして写真を検索し、他ピアが保持する写真を取得する事ができる。

各ピアは定期的（例えば 1 分おき）にマッピングエントリを更新する。マッピングエントリの値としては IP アドレスだけでなく、現在のアクセス回線種別やその状態（電波強度など）を含める。これにより、検索を行ったピアは、ファイルを所持するピアが複数ある場合、回線速度などからファイルを取得する相手を選択する事ができる。また、各エントリには更新時刻を付与し、一定期間（例えば 5 分）更新されていないマッピングエントリに関しては、そのピアは既に電波圏外もしくは電源が切断されていると判断する。

図 4 に検索結果を例を示す。同じタイトルを持ったファイルが 2 つあり、それぞれを 3 つのピアが所持している事が分かる。この時各ピアのアクセス回線種別などの情報から、もっとも早くファイルを取得できるピアを選択し、そのピアからファイルをダウンロードする事が可能となる。

4.2 実装詳細

前節で述べた写真共有アプリケーションを、PC ならびに Android OS¹³⁾ 搭載のモバイル端末上に実装した。Android OS は、2007 年に米 Google 社が発表した Linux ベースのモバイル端末向け OS である。アプリケーション記述言語としては、Java Platform, Standard Edition¹⁴⁾ のサブセットを採用している。ソースの共有を考慮し、PC とモバイル端末共に

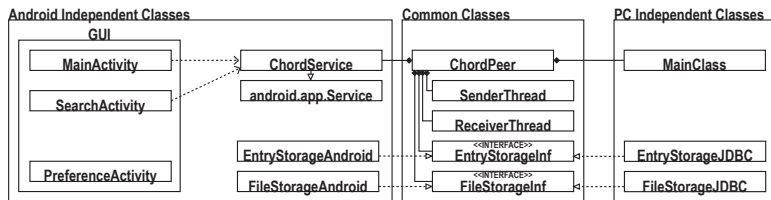


図 5 クラス図の一部
Fig. 5 A Part of Class Diagram

Java を利用して開発した。図 5 に、データ構造に関するクラスを除いたクラス図の一部を示す。双方の違いに留意しつつ実装する事により、プロトコル処理やネットワーク処理など、中心となる処理を行うクラスは共通化でき、Android OS と PC とで異なる箇所は、大きくデータベースと GUI の 2 つであった。

エントリーやファイル情報など、永続化が必要なデータを保持するデータベースには、Android OS/PC 共に SQLite¹⁵⁾ を用いた。Java において、データベースへのアクセスは、通常 JDBC (Java Database Connectivity)¹⁶⁾ が利用されるが、Android OS では SQLiteDatabase クラスの利用が推奨されている¹⁷⁾。このため、データベースへのアクセスを Interface を用いて抽象化し、それぞれの別のクラスを用いて具体化した。

今回、Android OS 向けにのみ GUI を実装し、PC 向けは CUI による操作を実装した。Android OS では Activity クラスを用いて画面を作成し、画面の遷移は Activity クラスの切り替えで実装する。しかしながら、P2P ネットワーク上のピアとしての機能は画面遷移に依存せず、常に動作している必要がある。そのため、常駐して処理を実行可能な Service クラスを継承した ChordService クラスを実装し、Chord のプロトコル処理を担当させた。

4.3 実機による評価

実装した写真共有アプリケーションを用いて、非対称型 P2P ネットワークの効果を評価した。PC と Android 端末および Wi-Fi アクセスポイント (AP) を 1 台ずつ用意した。表 1 にそれぞれの端末の仕様を示す。PC は有線で、Android 端末は Wi-Fi 経由で AP に接続されている。AP はブリッジモードで動作し、PC と Android 端末は同一サブネットに所属する環境とした。PC において、複数の写真共有アプリケーションをポート番号を変えて動作させる事で、30 個のピアを模擬させた。Android 端末は、PC ピアと同様に動作し既存の P2P ネットワークを構築する場合と、モバイルピアとして動作させ非対称型 P2P ネットワークを構築する場合を切り替えられるようにした。PC 上で動作する 30 ピアは、それぞ

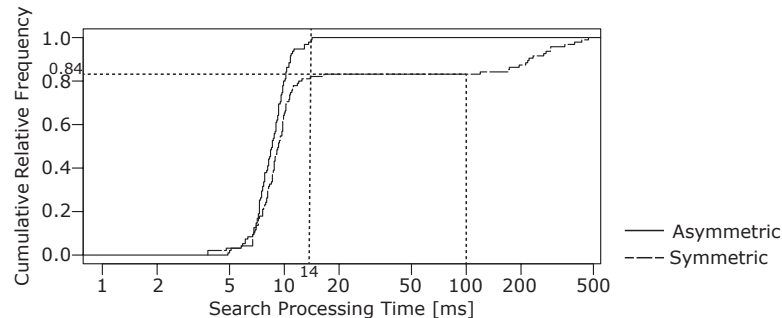


図 6 検索処理時間の累積相対度数分布
Fig. 6 Cumulative Relative Frequency of Search Processing Time

れ 20 枚の写真を持っており、P2P ネットワークに挿入されている。この時、PC 上で動作する 1 つのピアから検索メッセージを 5 秒おきに 100 個送信し、検索メッセージを送信してから、名前解決処理が完了するまでの時間 (検索処理時間) を計測した。データ複製による影響を除くため、すべての検索メッセージのキーは、異なるキーを用いた。

図 6 に検索処理時間の累積相対度数分布を示す。非対称型 P2P ネットワークにおいては 14ms 以下ですべての検索が完了しているのに対し、既存の P2P ネットワークでは、全体の 16% の検索において、検索処理時間が 100ms を超えている事が分かる。すなわち、30 ピアの中に 1 つのモバイルピアが参加しただけで、検索処理時間は大きく増加すると言える。

この検索処理時間の増加は、モバイルピアが検索メッセージが通過した検索パスに含まれるためであると考えられる。非対称型 P2P ネットワークにおける検索処理は、探索処理と名前解決処理の 2 フェーズで構成されるため、そのどちらかの検索パスがモバイルピアを通過した場合、検索処理時間が大きく増加する。 n ピアで構成される P2P ネットワークにおいて、モバイルピアが 1 ピア含まれている場合、2 つの検索パスの少なくともどちらか一方にモバイルピアが含まれる確率は、平均検索パス長を m とすると次式で示される。

表 1 PC とモバイル端末の仕様
Table 1 Specification of PC and Mobile Device

	Mobile Peer	PC
OS	Android OS 2.3.4	Fedora Linux 5
CPU	Qualcom Snapdragon 1GHz	Intel Pentium 4 3.4GHz
VM	Dalvik	Java Hotspot

$$P(X) = 2 \cdot \frac{1 \cdot n-1 C_{m-1}}{n C_m}$$

平均検索パス長 m を $\lceil 1/2 \cdot \log_2 n \rceil$ で近似すると、 $n = 31$ の時 $P(X) \simeq 0.194$ となる。これは 16% の検索で検索処理が増加するという実験結果とほぼ一致する。

5. おわりに

本稿では、固定網・モバイル網混在環境において P2P ネットワークを構築するため、非対称型 P2P ネットワークを提案した。本ネットワークは、各ピア毎に最適化された異なるトポロジを持つ事と、ピア ID と IP アドレスのペアを P2P ネットワーク自身で保持する事の特徴としている。これにより、固定網・モバイル網混在環境において P2P ネットワークを構築する時の 2 つの問題、モバイル端末の高伝送遅延ならびに IP アドレスの不安定性を解決する事ができる。

さらに、PC と Android OS 搭載のモバイル端末に対して提案した非対称型 P2P ネットワークを実装した。Android を含めて様々なプラットホーム動作可能な Java を開発言語用いる事により、多くのソースコードを共有した実装が可能であった。本実装を用いて 30 台の固定網に接続したピアで構築された P2P ネットワークに、1 台のモバイル端末が参加した時の影響を解析した。既存手法では、14% の確率で検索処理に 100ms 以上必要であったのに対し、提案手法ではすべての検索処理が 14ms 以下で完了した。これより、提案手法の有効性が示せたと言える。今後実際に 3G 回線を用いた時の評価を行う予定である。

参考文献

- 1) 亀井 聡：P2P 技術とインフラを巡る動向，電子情報通信学会技術研究報告. IN, 情報ネットワーク, Vol.109, No.79, pp.23–28 (2009).
- 2) Ekler, P., Nurminen, J.K. and Kiss, A.: Experiences of implementing BitTorrent on Java ME platform, *IEEE Workshop on Peer-to-Peer for Handheld Devices (P2PHD)*, Las Vegas, NV, pp.1154 – 1158 (2008).
- 3) 吉田将人, 鎌ヶ迫正俊, 林雄一郎：ケータイ用 P2P フレームワーク:Spear & Spear Multi, 電子情報通信学会技術研究報告. MoMuC, モバイルマルチメディア通信, Vol.107, No.446, pp.95–96 (2008).
- 4) Peng, G., Li, S., Jin, H. and Ma, T.: M-CAN: a Lookup Protocol for Mobile Peer-to-Peer Environment, *International Symposium on Parallel Architectures, Algorithms and Networks (ISPAN)*, Hong Kong, China, IEEE, pp.544 – 549 (2004).
- 5) Ratnasamy, S., Francis, P., Handley, M., Karp, R. and Shenker, S.: A Scalable Content-Addressable Network, *Conference on the applications, technologies, architectures, and protocols for computer communication (SIGCOMM)*, San Diego, CA, ACM, pp.161–172 (2001).
- 6) Stoica, I., Morris, R., Liben-Nowell, D., Karger, D., Kaashoek, M., Dabek, F. and Balakrishnan, H.: Chord: a scalable peer-to-peer lookup protocol for internet applications, *Networking, IEEE/ACM Transactions on*, Vol.11, No.1, pp.17–32 (2003).
- 7) Corson, S. and Macker, J.: Mobile Ad hoc Networking (MANET) : Routing Protocol Performance Issues and Evaluation Considerations, *Network Working Group RFC 2501* (1999).
- 8) Rajagopalan, S. and Shen, C.C.: A Cross-layer Decentralized BitTorrent for Mobile Ad Hoc Networks, *Proc. of International Conference on Mobile and Ubiquitous Systems (BIQUITOUS)*, Los Alamitos, CA, pp.1–10 (2006).
- 9) Conti, M., Gregori, E. and Turi, G.: A Cross-Layer Optimization of Gnutella for Mobile Ad Hoc Networks, *Proc. of ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc)*, Urbana-Champaign, IL, pp.343–354 (2005).
- 10) Luo, H., Ramjee, R., Sinha, P., Li, L.E. and Lu, S.: UCAN: A Unified Cellular and Ad-Hoc Network Architecture, *Proc. of International Conference on Mobile Computing and Networking (MobiCom)*, New York, NY, pp.353–367 (2003).
- 11) Ratnasamy, S., Francis, P., Handley, M., Karp, R. and Shenker, S.: A Scalable Content-Addressable Network, *Proc. of Conference on the Applications, Technologies, Architectures, and Protocols for Computer Communication (SIGCOMM)*, San Diego, CA, pp.161–172 (2001).
- 12) Eastlake, D. and Jones, P.: US Secure Hash Algorithm 1 (SHA1), *Network Working Group RFC 3174* (2001).
- 13) Open Handset Alliance: Open Handset Alliance, <http://www.openhandsetalliance.com/>.
- 14) Oracle Corporation: Java SE Overview – at a Glance, <http://www.oracle.com/tech/network/java/javase/overview/index.html>.
- 15) Hipp, D.R.: SQLite Home Page, <http://www.sqlite.org/>.
- 16) Oracle Corporation: Java SE Technologies - Database, <http://www.oracle.com/tech/network/java/javase/tech/index-jsp-136101.html>.
- 17) Android Open Source Project: Android Open Source Project, <http://developer.android.com/>.