

# 分散ハッシュによる名前管理と モバイルチャットシステムへの応用

大石忠広<sup>1</sup>

眞壁浩二<sup>2</sup>

佐藤文明<sup>2</sup>

高速で常時接続可能なインターネット環境の普及とともに、P2P による様々なアプリケーションが登場した。また携帯電話もパケットの定額制やネットワークの高帯域化が進み、携帯電話をつかった様々なサービスが提供されるようになってきた。携帯電話は常に利用者に携帯されているという特徴から、様々なアプリケーションに利用できる可能性がある。しかし携帯電話にかかる制約上、携帯電話上のアプリケーション間で直接通信できるものは少ない。本稿では分散ハッシュによる名前管理を利用して携帯電話ユーザによる P2P ネットワークを構築する。さらに携帯電話で提供される HTTP 通信上で擬似双方向通信を行うことにより、エンドユーザ間の擬似直接通信を実現する。この通信方式により、ネットワークに参加する任意のユーザを検索し、オンデマンドでのアクセスが可能となる。

## Node Lookup Based On Distributed Hash Method and Its Application to the Mobile Chat System

Tadahiro Oishi<sup>1</sup>

Kouji Makabe<sup>2</sup>

Fumiaki Sato<sup>2</sup>

Various applications based on the P2P technology have appeared by the spread of the high-speed Internet environment. Moreover, various services which use the cellular phone have come to be provided because of the fixed charge system and high-speed wireless communication of the cellular phone. The cellular phone would be possible to be used for various applications because the cellular phone is always being carried by the users. However, because of the restriction of the cellular phone, few applications communicate with each other directly. In this paper, we develop the P2P network for cellular phone based on the name service using the distributed hash technology. Furthermore, we realize the pseudo direct communication between end-users by using pseudo bi-directional communication over the HTTP protocol of the cellular phone. By using these technologies, users can look for the other users in the system and communicate between them on demand.

### 1 はじめに

近年、ADSL や FTTH などの高速で常時接続可能なインターネット通信環境が広く普及してきた。これに伴い、Napster[1]、Gnutella[2] に代表されるピア・ツー・ピア (P2P) アプリケーションの開発・利用が活発におこなわれてきた。また、skype[3] や SIONet[4] のように、ファイル共有に留まらず幅広いサービスやアプリケーションで P2P が利用されており、今後も P2P 技術をベースとしたサービスやアプリケーションの普及が進むといわれている。P2P は従来のインターネットにおいて主流だったクライアント/サーバモデルのように要求を集中管理するサーバが不要であり、構築が比較的容易である。さらに処理を分散できるため個々のコンピュー

ターの性能や回線の容量がそれほど要求されず、耐故障性や拡張性に優れているという性質をもつ。

一方、携帯電話からのインターネットアクセスも近年急増している。特に第 3 世代携帯の登場による通信速度の高速化や機体性能の向上、さらにはパケット通信料の定額制導入により携帯電話からインターネットアクセスを利用したアプリケーションの利用機会がいっそう加速することが予想される。

しかしながら、携帯電話用アプリケーションの開発には依然として制約が多いのが問題である。特にアプリケーションからインターネットにアクセスする際、利用できる通信プロトコルが HTTP (HTTPS) に制限されているため、アプリケーションからは限定された形でインターネットに接続される事が多い。そのため、現在 PC 等で実現されているサービスやアプリケーションを携帯電話上で実現することが困難となっている。

本研究の目的は、携帯電話からのインターネット

<sup>1</sup>静岡大学情報学院情報学研究科

Graduate School of Information, Shizuoka University

<sup>2</sup>静岡大学情報学部

Faculty of Information, Shizuoka University

アクセスに関する制約を隠蔽し、エンドユーザ間での直接的な通信を可能とするモバイル向け P2P ネットワークシステムの実現である。P2P 型通信に基づくシステムの研究開発は広く行われているが、モバイル機器を対象とした事例はまだそれほど多くないのが現状である。その中で、携帯電話用の P2P ミドルウェアとして JXME(JXTA for J2ME)[5] を紹介する。JXME は MIDP/CLDC 対応のデバイスを JXTA P2P ネットワークに参加可能にするミドルウェアであり、リレーピアと呼ばれる Proxy がリソースの発見やバイナリ XML 変換、プロトコル変換等を代行することによって携帯電話に実装する機能を最小限にしている。JXME はネットワーク上のリソースの検索が Gnutella ベースとなっているため、リソース検索にかかる負荷が大きく、またスケーラビリティの面で問題がある。さらに、JXME では P2P ネットワークからのメッセージを受信するために各携帯電話が一定間隔で Proxy にポーリングする方式をとっているため、エンドユーザ間でのリアルタイムな通信が困難という問題もある。一方、無線 LAN ベースによるワイヤレス機器向け P2P アプリケーションのミドルウェアとして、DECENTRA[6] などがある。

本稿で提案するシステムは、広く普及し人々が常に持ち歩く携帯電話をターゲットとした P2P ネットワークである。そのため、検索にかかる負荷を極力下げつつ、いかに効率良く通信したいユーザを検索するかが重要となってくる。

## 2 分散ハッシュによる名前検索

### 2.1 分散ハッシュテーブル

オーバーレイネットワークの基礎技術として注目されているのが分散ハッシュテーブル (Distributed Hash table) である。分散ハッシュテーブル (以下 DHT) は、あるハッシュ空間を複数のノードで構成し、検索キーやメタ情報をハッシュにかけて得られた値 (=hash(key)) を ID とするノードに検索先やデータの保持を割り当てる方式である (図 1)。

DHT は Napster のような Hybrid 型 P2P や Gnutella のような Pure 型 P2P よりもスケーラビリティが良く、全てのノードに対して検索が可能、負荷の分散等の点で優れている。DHT の実装アル

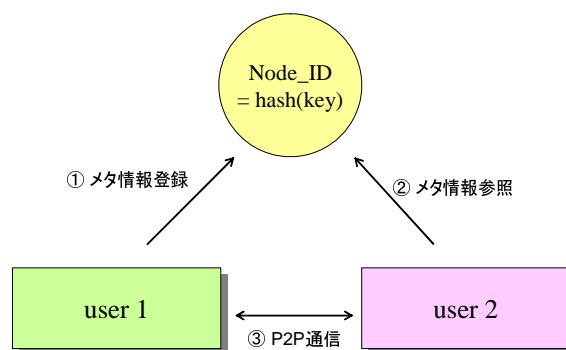


図 1: 分散ハッシュにおけるデータ登録・参照

ゴリズムとして Chord[8], CAN, Pastry などが提案されているが、本研究では実装が比較的容易でデータの検索が高速な Chord を分散ハッシュアルゴリズムとして利用する。なお Chord ではハッシュ関数として SHA-1 を想定しているが、本システムでは実装を簡単にするため簡易なハッシュ関数を独自に設定して利用する。以降、Chord のルーティングアルゴリズムの詳細を示す。

### 2.2 Chord のルーティングアルゴリズム

Chord ネットワークにおいて、ID を持った各ノードは仮想的なリングを形成し、時計方向に接続を行っている。このネットワークの中であるリソースを参照する場合、検索者が検索キーワードをハッシュ化したキー値を送信するとキー値に対応したデータを持つノードへルーティングされる。Chord では各ノードが skiplist と呼ばれるルーティングテーブルを常に保持することにより、次にどのノードに対して検索をしにいくかを決定している。m ビットハッシュ空間においてネットワークが n 個のノードで構成されている場合を考えると、skiplist は m 個のエントリを持ち、各エントリは 2 つのフィールド (キー値とそのキー値に対応したデータを持つノード) からなる (図 2)。このキー値は自ノードの ID を k とするとき  $key = k + 2^i \pmod{2^m}$  で表され、検索するノードを次のノードに渡すたびに検索すべきハッシュ空間を 1/2 に絞っていくことになる。これにより、 $\log m$  のルーティングテーブルを保持することにより、最大  $\log m$  回のホップ数での検索が可能である。

一般に、Chord ネットワークを構成するノードの頻繁な参加・脱退は負荷が大きい。これは、ノード

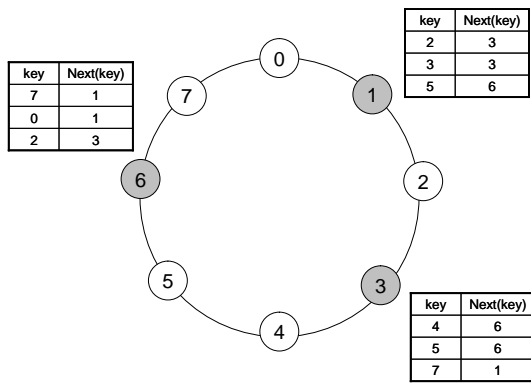


図 2: skip list によるルーティング

の参加・脱退の度にオーバーレイネットワークの構成が変化し、その都度制御パケットを送信してリングの再構築をすることに起因する。ユーザの参加・脱退頻度が高いシステムでは、ネットワーク構成の頻繁な変化によるシステムのパフォーマンス低下が予想される。

### 3 クライアント間擬似直接通信

筆者らはこれまでに、携帯電話で動作する Java アプリケーション上において、HTTP による擬似双方向通信の実装とその有効性を確認している [9][10]。本システムでもインフラ上の Proxy ノードとノードに直接通信する携帯電話間の通信に HTTP 擬似双方向通信を用いる事により、任意のタイミングで Proxy ノードから携帯電話へのアクセスを可能とする。これにより、ネットワークに参加している任意のユーザ間での擬似的な直接通信が実現できる。

なお、本稿で用いる持続的接続というフレーズは HTTP/1.1 における持続的接続 (persistent connection)[7] とは別のものであり、その挙動も異なる)。以下、HTTP の持続的接続の概要を簡潔に紹介する。

#### 3.1 HTTP 持続的接続

本システムで用いる HTTP 持続的接続とは、HTTP クライアントからのリクエストに対してサーバがレスポンスを返さず、保持している状態を指す。接続クライアントはこの状態を常に維持する事により、通信相手のユーザから発行されたメッセージがサーバに到着したタイミングで、保留されていたレ

スponseとしてこのメッセージをサーバから受け取ることができる (図 3)。従来、HTTP 上で双方向通信を実現する手法として各クライアントが一定間隔おきにサーバにポーリングする方式がとられてきたが、従来方式ではリアルタイム性と送信パケット量がトレードオフとなる。一方、持続的接続は送信パケット量を極力減らし、リアルタイム性も向上することからチャットやネットワークゲーム等のリアルタイム性が要求されるシステムに適していると考えられる。

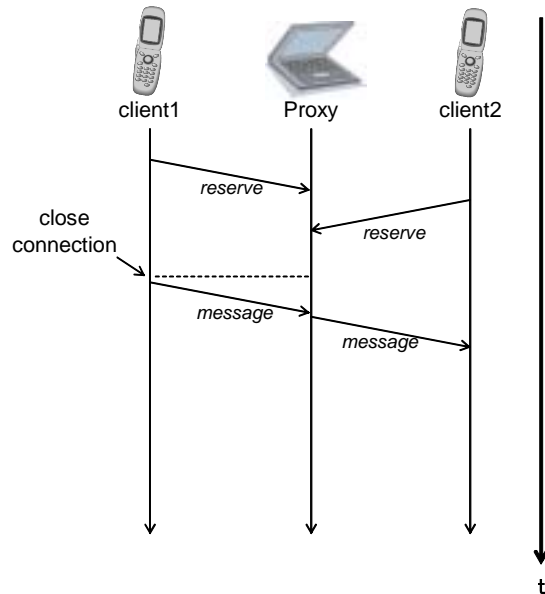


図 3: HTTP の持続的接続を用いたクライアント間通信

この通信方式により HTTP 上でリアルタイムな双方向通信が可能となる一方で、持続的接続にはメッセージを中継する Proxy の負担が大きいという問題がある。Proxy は常に接続クライアントからのリクエストを保持し、保持したレスポンスを適切なタイミングで該当クライアントに発行しなければならない。そのため、接続クライアントの数が増えるとその負荷が増大し、応答性の低下やネットワークパフォーマンスの低下につながる。

### 4 システム概要

本稿で提案するシステムでは Chord ネットワークを Proxy ノードによって構成し、Proxy ノードに

クライアントが接続する形態をとる．これにより、ノードの頻繁な参加・脱退による Chord ネットワークのパフォーマンス低下や、ユーザ間の持続的接続通信時に中継 Proxy にかかる負荷の軽減などが期待できる．

本システムでは、リソースの検索や登録といったサービスを実際に利用するのは構成ノードでなく、構成ノードに接続するクライアント端末である．つまり、ネットワークへの頻繁な参加・脱退を実際に発行するのはクライアント端末であり、Proxy ノードがクライアントの要求を代表して発行することによりネットワーク構成の頻繁な変化を吸収する役割をもつ．また HTTP 持続的接続によるエンドユーザ間通信を行う際、実際の通信は複数の Proxy ノードを経由して行われる事が多い．そのため、ユーザ間通信時に Proxy にかかる負荷を抑え、またネットワーク全体の負荷分散に有効であるといえる．

#### 4.1 基本アーキテクチャ

本稿で提案するシステムは、携帯電話 (クライアント) とインフラ上の Proxy ノード、リストサーバの 3 つから構成される．(図 4)

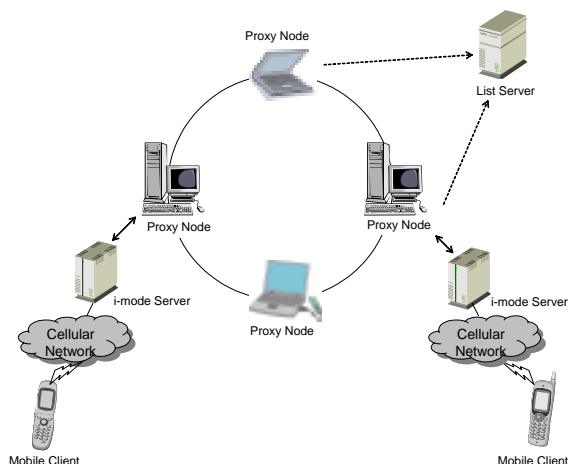


図 4: システム構成

#### Proxy ノード

Proxy ノードは、Chord と呼ばれる分散ハッシュアルゴリズムを用いてインフラ上にオーバーレイネットワークを構築する．Chord は  $n$  ビットのハッシュ空間を保持しており、各 Proxy ノードにはハッシュ空間内から一意なノード ID

が割り当てられる．このハッシュ空間はリング状につながっており、ノード ID をもった各ノードが仮想的なリングを形成する．Chord では、データの検索やネットワークへの参加/脱退に特定のサーバを必要とせず、各ノードが完全に分散して処理を行う．

#### 携帯電話 (クライアント)

ネットワークに参加する携帯電話ユーザはクライアントアプリケーション上からリングを構築している Proxy ノードのいずれかに接続し、以後ネットワークへのアクセスは接続した Proxy ノードを介して行う．ユーザはリクエスト (リソース検索、エンドユーザ間通信等) を接続した Proxy ノードに送信し、Proxy ノードが Chord ネットワーク内での処理を代行する．ユーザは得られた結果を Proxy ノードから受け取ることで様々なサービスを利用することができる．なお、携帯電話はインターネットにアクセスする際、無線通信網を通じて i モードサーバと呼ばれるゲートウェイを中継しているため、インフラ上で Proxy ノードと直接的に通信するのは i モードサーバである．

#### リストサーバ

リストサーバは、現在 Chord ネットワークを形成している実在ノードのノード ID と IP アドレスの対応表を公開するサーバである．リストサーバは Proxy ノード内に処理として組み込むことも可能であるが、将来的に携帯電話から直接リストを参照することを考慮して今回は Proxy ノードとは独立してリストの処理のみを行うリストサーバを設定した．各 Proxy ノードは新規参加時やノードリングの構造が変化した際に必要に応じてリストサーバにアクセスすることで現在のノードリングの状態を知ることができる．

## 5 実装

### 5.1 クライアントアプリケーションの実装

まずユーザと P2P ネットワークとのインターフェースになる Java アプリケーションを実装する．今

回実装したプロトタイプでは、ユーザがクライアントアプリケーションを通して利用できるサービスは以下の4つである。

- ・ P2P ネットワークへの参加要求
- ・ P2P ネットワークからの脱退要求
- ・ P2P ネットワーク上のリソース (ユーザ) 参照
- ・ P2P ネットワークに参加する任意のユーザとの双方向通信 (チャット)

ネットワークへの参加要求はクライアントアプリケーションの起動後に発行される。その際、ネットワーク内で使用するユーザ名 (ニックネーム) を指定する。ネットワーク上からあるユーザの情報を参照するときはこのユーザ名をキーとして検索が実行される。

ネットワーク上のリソース参照サービスでは P2P ネットワーク上に分散されたリソースの検索を Proxy ノードに依頼し、その検索結果を得る。今回実装したプロトタイプでは、検索するリソースのキーとしてユーザ名のみ指定が可能であり、検索結果は対象ユーザの所在 (接続している Proxy ノードの IP) とする。

任意のユーザとの双方向通信サービスはネットワーク上のユーザ検索と一緒に利用され、検索したユーザと直接通信する際に発行される。なお、今回はユーザ間双方向通信機能のとして簡易なチャットアプリケーションを実装する。実装の際に考慮すべき携帯電話固有の制約については次節で述べる。

## 5.2 携帯電話の制約

本システムのクライアント端末が対象とする携帯電話は J2ME/CLDC/Doja Profile をサポートした第 2, 3 世代携帯電話 (i-mode 携帯電話) である。この i-mode 携帯電話には、同時に複数のアプリケーションを起動できない、クラスロード機能がない等の制約がある他、インターネットアクセスやアプリケーションサイズに関する制約がある。(表 5) に Doja Profile 2.x, 3.0 及び 3.5 の主な仕様をまとめた。

## 5.3 プロトタイプの実装

インフラ上の PC に Proxy ノードモジュールとリストサーバモジュールを、現行の携帯電話機にク

	Doja 2.x Profile	Doja 3.0 Profile	Doja 3.5 Profile
HTTPバージョン	HTTP /1.0	HTTP /1.1	HTTP /1.1
同時に開可能なコネクション数	1	1	1
アプリケーションからアクセスできるサーバ	ダウンロード元サーバのみ	ダウンロード元サーバのみ	ダウンロード元サーバのみ
アプリケーションファイルのサイズ制限	30K byte	30K byte	100K byte
最大通信速度 (下り)	28.8K bps	28.8K bps	384k bps
最大通信速度 (上り)	9.6K bps	9.6K bps	64K bps
HTTP受信データサイズ制限	10K byte	20K byte	20K byte
HTTP送信データサイズ制限	5K byte	10K byte	10K byte

図 5: Doja Profile による仕様

クライアントアプリケーションのプロトタイプを実装した。(図 6) にモバイルチャットシステムの機能構成を示す。

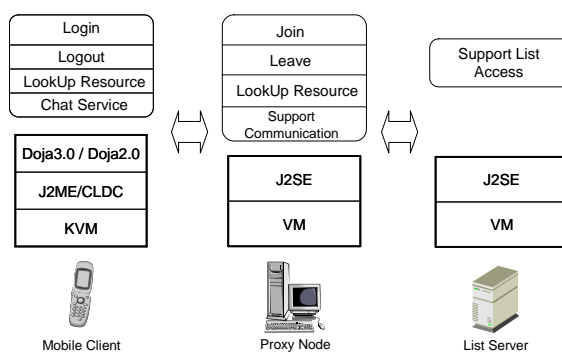


図 6: モバイルチャットシステムの機能構成

## 5.4 動作検証

実装したプロトタイプにおいて、各サービス・機能の動作を検証した。動作確認を行った環境は以下の通りである。

- ・ リストサーバ: デスクトップ PC×1
- ・ Proxy ノード: デスクトップ PC×1, ノート PC×3
- ・ クライアント端末: i-mode 対応携帯電話 × 4(504i, FOMA900i)
- ・ ハッシュ空間: 5 ビットハッシュ空間

動作検証を行った結果、分散ハッシュによる基本的な処理 (ノードの参加 / 脱退, リソースの探索) とエンドユーザ間での擬似双方向通信によるリアルタイムなチャットが実現できたことを確認した。

## 6 考察

### 6.1 本システムの適用例

本システムは、インフラ上のネットワークを通じて携帯電話上から利用可能な P2P ネットワークの構築を実現するものである。今回実装したプロトタイプでは、現行の携帯電話機から Proxy ノードを介して P2P ネットワーク上のリソース検索サービスを利用し、任意のユーザへのオンデマンドなアクセスを実現した。さらに、HTTP 擬似双方向通信を利用することでユーザ間での直接的な通信を確認できた。今回のプロトタイプでは設定可能な検索キーはユーザ名のみであったが、ユーザの状態や所属グループ、所在、プロフィールなどのメタ情報を検索キーとして指定することも考えられる。またプロトタイプではユーザ間の直接通信機能として簡易なチャットシステムを搭載しているが、文字列によるメッセージのみでなく、携帯電話に保存されている画像等のファイルの送信も考えられる。

### 6.2 課題

今回実装したプロトタイプにより、携帯電話上から P2P を利用した基本的な処理は動作するが、まだ多くの課題が残されている。以下に、今後解決すべき課題を列挙する。

- ・ファイアーウォールの内側にいる PC は Proxy ノードになることができない
- ・独自のハッシュ関数であるため、負荷の分散を考慮していない
- ・不意に Proxy ノードがネットワークから脱退した場合の処理が実装されていない
- ・メッセージの盗聴や改ざんの可能性がある

## 7 まとめ

本稿では、携帯電話上から利用できる P2P ネットワークシステムを実装し、Chord アルゴリズムによるハッシュ空間の構築と管理、また HTTP の擬似双方向通信を利用したユーザ間の end-to-end 通信を実現し、動作を確認した。プロトタイプではリソース検索のキーとしてユーザ名のみ利用できたが、扱うキーを追加することでユーザのニーズに

応じた柔軟なリソース検索が可能になる。今後の課題は、扱うキー情報の追加とユーザ間通信機能の充実、より大規模な環境での稼働テストなどである。

## 参考文献

- [1] Napster. <http://www.napster.com/>
- [2] Gnutella - LimeWire Pro.  
<http://www.limewire.com/>
- [3] skype. <http://web.skype.com/home.ja.html>
- [4] SIONet. <http://www.ntt.co.jp/news/news01/0104/010427.htm>
- [5] JXTA J2ME Implementation Project, 2003.  
<http://jxme.jxta.org/>
- [6] <http://www.skyley.com/products/decentra.html>
- [7] <http://www.w3.org/Protocols/HTTP/Issues/http-persist.html>
- [8] Ion Stoica, Robert Morris, David Karger, M. Frans Kaashoek, and Hari Balakrishnan, "Chord: A Scalable Peer-to-peer Lookup Service for Internet Applications", In the Proceedings of ACM SIGCOMM 2001, San Deigo, CA, August (2001).
- [9] 大石忠広, 佐藤文明, "携帯電話における HTTP 擬似相互通信によるリアルタイムユーザ共有空間" DICOMO2003, June 2003.
- [10] 大石忠広, 佐藤文明, "携帯電話での擬似双方向通信に基づくユビキタスアプリケーションの提案" 情報処理学会マルチメディア通信と分散処理研究会研究報告 116-11, 2003