

近傍内の一時的な情報共有機能を有する P2P チャットアプリの開発 P2P chat application for information sharing with limited communication range

木谷 有生誠†
Yukimasa Kiya

高井 昌彰‡
Yoshiaki Takai

1. はじめに

SNS の普及により、自由な情報発信が可能となる一方で、プライバシーを考慮せずに他人の写真を投稿することや、実名や住所など現実世界において個人の特定に繋がる情報を発信するといったプライベートな情報の氾濫が近年ネット社会において問題となっており、対策が求められている。

そこで本研究では、物理的に限られた近傍内での一時的なコミュニケーション場の自律形成に注目し、あるユーザから見て一定距離の範囲内に存在する他ユーザとの間でのみ、直接的な P2P 通信によるリアルタイムの情報共有を可能にするチャットアプリ「cappi」を開発する。

本システムでは、端末間の近接・距離センシングを行う iBeacon[1]を利用した周囲のユーザに対する自動的な接続要求通知機能を実現し、アドホックなコミュニケーション場の自律形成を促進すると共に、チャット終了時にメッセージを自端末から全て消去することで情報の一過性を担保する。同様な P2P ベースのチャットアプリに Open Garden 社の FireChat[2]があるが、cappi では情報を一過性のものとして扱う点、チャット希望通知の実装及びメッセージ拡散機能に特徴がある。本論文では、cappi の iOS でのインプリメントとその動作の概要について述べる。

2. システムの概要

本システムの概要を図1に示す。各ユーザはアプリ起動時にユーザ名と開設するルーム ID を設定する。その後チャットルームを開設し、ルーム ID を通信可能範囲内でブロードキャストすると共に自身と同一のルーム ID を設定しているユーザを検索する。

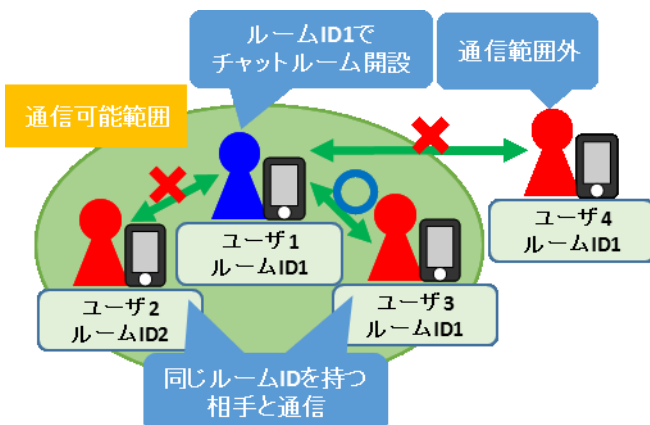


図1 P2P チャットアプリ cappi のシステム概要

†北海道大学大学院情報科学研究科, Graduate school of Information Science and Technology, Hokkaido University

‡北海道大学情報基盤センター, Information Initiative Center, Hokkaido University

検索で発見されたユーザは自分のチャットルームに登録され、チャットを行うことが可能となる。開設したチャットルームから抜けるとチャットが終了する。その際、自身の端末が保持しているメッセージはシステムによって全て消去される。

3. メッセージの構造

本システムで扱うメッセージは以下の要素からなる。

- メッセージ ID (UUID 型)
- 送信者のユーザ ID
- コンテンツ (文字列または画像データ)
- コンテンツタイプ
- 送信時刻
- 拡散希望者名 (拡散希望メッセージのみ)

メッセージ ID でメッセージを一意に識別し、重複を防ぐ。コンテンツタイプはコンテンツのデータ型を表す。拡散希望者名は後述するメッセージ拡散機能に用いる。

4. システムの実装

4.1 Multipeer Connectivity を用いた通信機能

本研究では、Wi-Fi または Bluetooth を介して近距離にある iOS 端末間の P2P 通信を実現するフレームワークである MultipeerConnectivity.framework[3]を用いてユーザ間の通信機能を以下のように実装した。

ユーザ間の接続処理

ユーザはチャットルームを開設後、自身のユーザ ID とルーム ID をブロードキャストする。そして、自身と同一のルーム ID を持つユーザのみと接続し、チャットルームに登録する。

チャット機能

送信側では、ヘッダデータをコンテンツデータの先頭に付加し送信データを作成する。ヘッダデータは3章で述べたメッセージ ID、コンテンツタイプ及び拡散希望者名に加えて、拡散希望者名のバイト列長からなる。次に送信データをバイト列データにエンコードし、チャットルーム内の全ユーザにブロードキャストする。

受信側では、送信データ及び送信者のユーザ ID を取得する。次に受信したデータからヘッダ部のみをデコードし、ヘッダデータを取得する。続いてコンテンツ部をヘッダデータに従ってデコードし、3章に示すメッセージ形式で取得する。

4.2 iBeacon を用いた自動的な接続要求機能

本研究では、Bluetooth LE を基盤とした端末間の近接・距離センシングを行う iBeacon を用いて周囲のユーザに自動的にチャット希望通知を送信する機能を実装した。

iBeacon は Proximity UUID、Major 及び Minor の3種類の ID 情報を持ち、端末間で送受信する仕組みを有している。本システムではチャット開始時に送信側を起動し、自身

のルーム ID を Minor 値に割り当てブロードキャストする。一方受信側は常時起動させておき、周囲の端末からの ID 情報を受け取りその Minor 値を読み取る。その際に、受信したデータの電波強度から送信元の端末とのおおよその距離を 4 段階で検知し、ルーム ID と合わせて図 2 のように通知する。



図 2 チャット希望通知受信の様子

4.3 メッセージ拡散機能

本研究ではチャット機能の充実のために、メッセージ拡散機能を以下のように実現した。

まず拡散したいメッセージを送信する際に、3 章で述べた拡散希望者名に本メッセージの送信者名を割り当てる。本研究ではこのメッセージを拡散希望メッセージと呼ぶ。拡散希望メッセージは、拡散を促すアイコンが付加された形でチャットビューに表示され、他ユーザが該当ビューをスワイプすることで、該当メッセージをコピーして自動送信する。

4.4 アプリケーションの構成

P2P チャットアプリ cappi は Xcode6.1.1 を用いて開発しており、iOS7.0 以上で動作する。cappi のビュー一覧を図 3 に示す。アプリ起動時には図 3(a)の設定ビューが表示される。ここでユーザ名及びルーム ID を設定し、チャットボタンから図 3(b)のチャットビューへと移行する。また、本アプリではチャットモードとして以下の 2 種類を用意した。これらのモードは図 3(a)上部のスイッチで切り替え可能である。

パブリックチャット

本モードはチャット開始時に iBeacon を利用し、4.2 節に示した方法で周囲の端末に対してルーム ID を通知する。このモードでチャットを行う際は、ルーム ID として 0~999 の整数値を割り当てることができる。これは、iBeacon で通知可能な ID が整数型を持つためである。

プライベートチャット

本モードでは iBeacon による周囲へのルーム ID の通知を行わずにチャットを行う。そのため、友人間やサークルの仲間内など、予め通信相手を限定したチャットを行いたい場合に有用である。このモードではルーム ID として半角英数の文字列を 14 文字以内で割り当てることができる。パブリックチャットに比べて表現できるルーム ID の幅を広げることで、周囲で開設している他のルームと ID が混同してしまう危険性を低減させている。

図 3(a)中央の表は周囲の端末から通知されてきたルーム ID を表示しており、セルをタップすることで該当するルーム ID を自身のルーム ID に設定することが出来る。この例では周囲の端末からルーム ID : 123 が通知されている。

拡散希望モードの変更は図 3(b)下部のスイッチで行う。図 3(b)中央部は拡散希望メッセージであり、この例では画像データ (北大ロゴマーク) を拡散させている。



(a)設定ビュー

(b)チャットビュー

図 3 実装した cappi のビュー

5. 通信性能の検証

本システムを実装した iPhone4s 二台を用いて通信性能を検証した結果、Bluetooth を介した接続可能距離は約 50m, その際のメッセージ送信-到達時間は 0.07 秒程度であった。また、iBeacon の通信可能距離は約 50m であった。この結果から、iBeacon による通知が届く範囲では Bluetooth によるチャットも可能であるといえる。

加えて、本システムを用いてチャットが可能である接続台数の上限を検証したところ、5 台までの同時通信であれば通信が安定し、動作が保障できることが判明した。

6. まとめと今後の課題

本論文では、物理的に限られた一定範囲内に存在するユーザ間でのみ、P2P 通信によるリアルタイムの情報共有を可能にする iOS チャットアプリの開発を行うと共に、iBeacon を用いた自動的な接続要求通知機能を実装した。加えて本システムの通信性能実験を行った。

今後は、UI の改善及び機能の充実に加え、評価実験による本システムの有用性の検証が課題となる。

参考文献

- [1]中道 理: “iOS7 の注目技術「iBeacon」距離測位の割り切りで実現”, 日経エレクトロニクス (1129), pp.20-23, 2014-03-03.
- [2]Open Garden | /faq
<https://opengarden.com/faq#faq-general-F1>
- [3] Multipeer Connectivity Framework Reference
<https://developer.apple.com/library/prerelease/ios/documentation/MultiPeerConnectivity/Reference/MultiPeerConnectivityFramework/index.html>