

平成26年度 卒業論文



琉球大学工学部情報工学科

115702H 新里亮太

指導教員 宮里智樹

目次

第1章	はじめに	1
1.1	背景	1
1.2	研究目的	1
1.3	本論文の構成	2
第2章	基礎概念	3
2.1	アドホックネットワーク	3
2.2	AllJoyn	3
2.2.1	AllJoyn アーキテクチャ	4
2.3	Bluetooth Low Energy	6
2.4	iBeacon	6
第3章	電子見守りシステム	8
3.1	電子見守りシステムの概要	8
3.2	見守りエリアからの逸脱管理	8
3.3	ビーコン情報の共有	8
第4章	設計と実装	10
4.1	提案手法	10
4.2	開発環境	10
4.3	AllJoynApp の概要	11
4.4	設計	11
4.4.1	AllJoyn セッション	11
4.4.2	iBeacon の検知	12
4.4.3	プログラムファイル	13
4.5	実装	14
4.5.1	アプリケーションの起動	14
4.5.2	FIND ボタン押下	16
4.5.3	SCAN ボタン押下	17
4.6	検証	19
4.7	考察	20

第 5 章	結論	21
5.1	結論	21
5.2	今後の課題	21

目 次

2.1	アドホックネットワークの例	3
2.2	AllJoyn を用いたアプリケーションの例	4
2.3	AllJoyn ネットワークの構成	5
3.1	見守りエリアからの逸脱管理	9
3.2	ビーコン情報の共有	9
4.1	AllJoyn セッションの例	11
4.2	AllJoynClient クラス図	13
4.3	AllJoynService クラス図	13
4.4	ユースケース図	14
4.5	アプリケーション起動画面	15
4.6	アプリケーション起動画面	15
4.7	FIND ボタン押下	16
4.8	FIND ボタン押下	17
4.9	SCAN ボタン押下	18
4.10	SCAN ボタン押下	18

表 目 次

4.1	スペック:TORQUE G01	10
4.2	スペック:ARROWS NX F-05F	11
4.3	iBeacon データ配列	12

第1章 はじめに

1.1 背景

日本では、高齢者人口が増加し少子高齢化社会が加速している。それに伴い、増加しているのが認知症患者数である。厚生労働省の調査 [2] によると 2010 年時点で 200 万人程度といわれ、専門家の間では、すでに 65 歳以上人口の 10%(242 万人程度) に達しているという意見もある。

年々増加している認知症患者であるが、介護現場では、徘徊による事故を防止するために、家族や介護者が多大な負担を強いられているのが現状である。認知症患者の介護者への調査 [3] によると、見守りが常に必要な割合が 4 割にものぼり、徘徊などで行方不明になったとして警察に届けられている数は 9607 人に及ぶ。愛知県大府市で 2007 年 12 月、徘徊症状がある認知症の男性が電車にはねられるという事故が発生し、介護者である男性の遺族への賠償判決が出された。

これらの現状から、介護者の負担を軽減し、認知症患者の徘徊を防止する仕組みの構築は急務だといえる。介護者への負担軽減を目的とした電子的・機械的なセンサーシステムはすでにあるが、既存施設への電気設備の敷設が困難、外出先での対応ができないなどの課題がある。そのため、スマートフォンを受信機とし、低価格の発信機 (iBeacon) を利用した電子見守りシステムの構築を株式会社国建システムと共同で行う。

電子見守りシステムでは、発信機と受信用スマートフォンが、1 対 1, 1 対多, 多対多の組み合わせでの見守りが可能なしくみを開発する。特に受信用スマートフォンが複数ある場合、スマートフォン同士でアドホックネットワークを構築し、発信機情報を共有することで見守りエリアの拡大を図る。

1.2 研究目的

本研究では、上述の電子見守りシステムのスマートフォン同士のアドホックネットワークの構築と、情報共有手法の実装を行い、それらの検討を行う。電子見守りシステムでは介護者それぞれのスマートフォンを用いて見守りを行うことを想定しており、異なる機器や OS での開発が予想される。そのため、クロスプラットフォームな P2P(peer-to-peer) 型端末通信フレームワークである AllJoyn を用いてアドホックネットワークの構築を行う。

1.3 本論文の構成

本論文の構成について解説する。

本章では、「はじめに」として本研究の背景と目的について述べた。

第二章では、本研究に関する概念や技術について述べる。

第三章では、本研究の背景としてある電子見守りシステムについて述べる。

第四章では、本研究で作成したアプリケーションの設計や仕様、考察について述べる。

第五章では、結論と今後の課題について述べる。

第2章 基礎概念

2.1 アドホックネットワーク

アドホックネットワーク [4] とは、アクセスポイントを必要としない無線で接続できる端末のみで構成されたネットワークである。直接電波が届き通信可能である場合には、始点端末から終点端末へ1 ホップでパケット伝達が行われる。電波が直接届かない場合には経由 (マルチホップ) することで終点端末までパケットを伝達する。端末の中継機能を利用してネットワークを構成しているので、アドホックネットワークには基地局インフラが不要である。そのため、限られた領域内の簡易なネットワークの構築の手段として有効である。

以下の図では始点端末から終点端末まで直接繋がっていないが、他の端末を経由し4 ホップで通信を行っている。

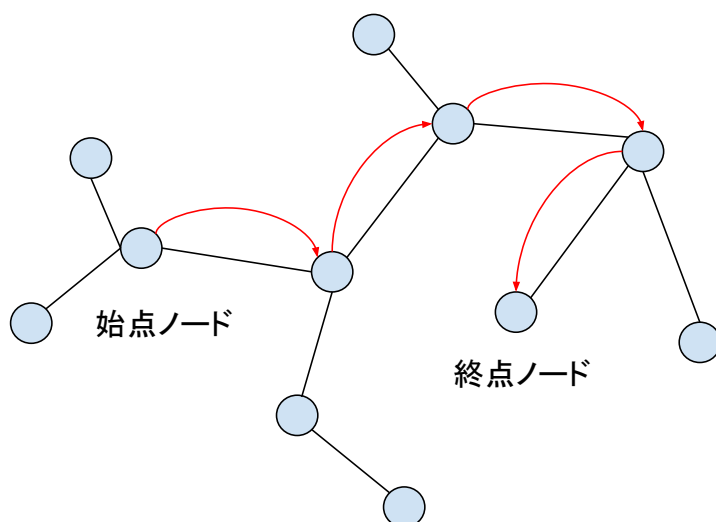


図 2.1: アドホックネットワークの例

2.2 AllJoyn

AllJoyn[5][6] は Qualcomm が開発したオープンソースプロジェクトである。端末間通信を実現し、製品やアプリケーション間の相互運用を可能にするデバイス通信フレームワークを提供している。Qualcomm は IoT 標準化団体である AllSeen に参加しており、AllSeen

が推進する IoT 規格は AllJoyn をベースに設計されている。OSI 参照モデルのセッション層以上の規格であり、トランスポート層には依存しないというメリットがある。そのため、Wi-Fi や Wi-Fi Direct, Bluetooth といった様々な通信規格を用いて開発することができる。Android, iOS, OS X, Linux, Windows7 など様々な OS に対応し、Java, C++, C, JavaScript などの言語で利用できる。AllJoyn は近傍デバイスの探索や P2P ネットワークへの接続、セキュリティなどの要素を提供しており、機器や OS に依存せず開発することができる。

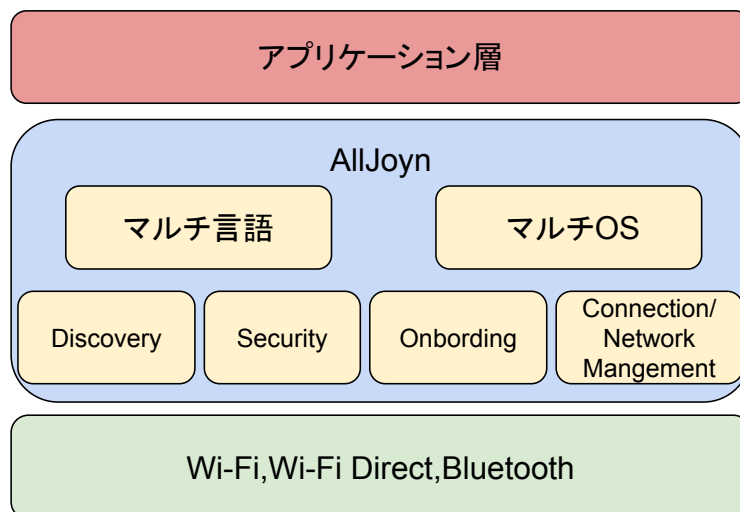


図 2.2: AllJoyn を用いたアプリケーションの例

2.2.1 AllJoyn アーキテクチャ

AllJoyn ネットワークは AllJoyn アプリケーションと AllJoyn ルーターから成る。AllJoyn アプリケーションは AllJoyn ルーターを通じて他の AllJoyn アプリケーションや AllJoyn ルーターと通信を行う。AllJoyn アプリケーションは以下の要素から成る。

- AllJoyn App Cods
- AllJoyn Service Frameworks Libraries
- AllJoyn Core Library

AllJoyn Core Library

AllJoyn Core Library は AllJoyn ネットワークと通信するための最低限の API を提供する。提供する API は以下の通りである。

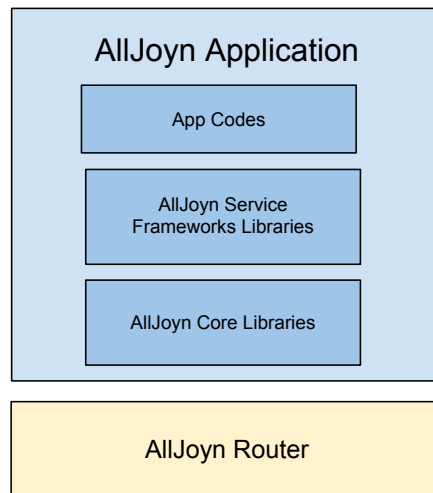


図 2.3: AllJoyn ネットワークの構成

- Advertisement UDP パケットを用いて AllJoyn サービスをブロードキャストする.
- Discovery Advertisement された AllJoyn サービスを検索する.
- Session creation セッションの確立
- Interface definition of methods, properties, and signals メソッド, プロパティ, シグナルのインターフェース定義
- Object creation and handling オブジェクトの生成とハンドリング

AllJoyn Service Framework Libraries

AllJoyn Service Framework Libraries は, onboarding, notificatino, control panel などのサービスを備えている. Base Services として以下の機能がある.

- Base Services
 - Onboarding
Wi-Fi ネットワークへの接続をサポート
 - Configuration
端末の構成
 - Notificatinos
デバイスからの通知
 - Control Panel
UI widget とリモート端末との通信

AllJoyn Service Framework Libraries を使うことで、アプリケーションや端末は互いにこれらのサービスを運用することができる。

AllJoyn App Code

AllJoyn App Code は AllJoyn アプリケーションのロジックである。高レベルの機能を提供する AllJoyn Service Frameworks か、直接 AllJoyn Core API にアクセスできる AllJoyn Core Library のどちらかを用いてプログラミングされる。

AllJoyn Router

AllJoyn Router は P2P Advertisement/Discovery, Connection Management といった AllJoyn システムの核となる機能を提供する。メッセージは AllJoyn Router を経由して他のアプリケーションに送信される。

2.3 Bluetooth Low Energy

Bluetooth Low Energy(BLE)[7][8] 近距離無線通信技術 Bluetooth の拡張仕様の 1 つであり、Bluetooth4.0 規格の一部として採用された。一般的な Bluetooth と同様に 2.4GHz 帯を使用し、データ転送速度は 1Mbps である。非常に少ない消費電力で動作することができ、ボタン電池で数年動作すると言われている。

2.4 iBeacon

iBeacon[9][10] は、2013 年に Apple が発表した Bluetooth Low Energy(BLE) を利用した技術であり、発信機から発せられるビーコンを受信し、発信機の位置を特定・確認できる。iOS7 以降の端末 (iPhone, iPad, および iPod touch) や Android4.3 以降の端末などで受信することができる。iBeacon では領域の出入りをチェックするリージョン監視やビーコンの情報を受信するレーシングを用いて位置や近接具合などを検知する。ビーコンを受信したときに得られる情報には以下の 6 つがある。

- proximity UUID : ビーコン識別子
- major : ビーコン識別子
- minor : ビーコン識別子
- proximity : ビーコンとの距離
- accuracy : 距離の精度

- rssi : 受信強度

また,proximity は正確な距離を表すものではなく, 以下の大まかな 4 つの値で得ることができる.

- Unknown : 検出不可
- Immediate : 至近距離
- Near : 近距離
- Far : 遠距離

第3章 電子見守りシステム

3.1 電子見守りシステムの概要

株式会社国建システムと共同で行っている電子見守りシステムでは、見守りエリアからの逸脱管理、ビーコン情報の共有という二つの機能がある。介護対象者が自身から離れた際に警告を発し、周囲の介護者にビーコン情報を送信する。大規模、広範囲の見守りになる場合には、見守り対象者を適宜グループ分けし、複数の介護者で分担して見守りを行う。以下の環境を想定してシステムの開発を行っている。

- 場所
 - － 屋内（一般家家庭、介護施設）
 - － 屋外
- 規模
 - － 1～10 名程度

3.2 見守りエリアからの逸脱管理

介護対象者に小型の発信機（iBeacon）を一個、もしくは複数個装着し、そこから発せられる電波（ビーコン）をスマートフォンで受信する。発信機から発せられる電波が介護者に届く範囲を見守りエリアとし、受信ビーコンが一定時間受信できない場合、見守りエリアから外れたとみなしてスマートフォンから警告メッセージを発する。発信機を複数個装着している場合には、「発信機の過半数が電波外になった場合に警告を発する」などといった設定も可能とする。また、電波を受信した際には見守りエリアに入ったとし、メッセージを発する。介護対象者・発信機が複数の場合も同様に逸脱管理を行う。

3.3 ビーコン情報の共有

受信用スマートフォンが複数ある場合は、スマートフォン同士で発信機情報を共有し、見守りエリアを拡大させる。自らの見守りエリアから逸脱した発信機が、他のスマートフォンで検知されていれば見守りは成功していると判断できる。スマートフォン同士で共有する発信機情報は、

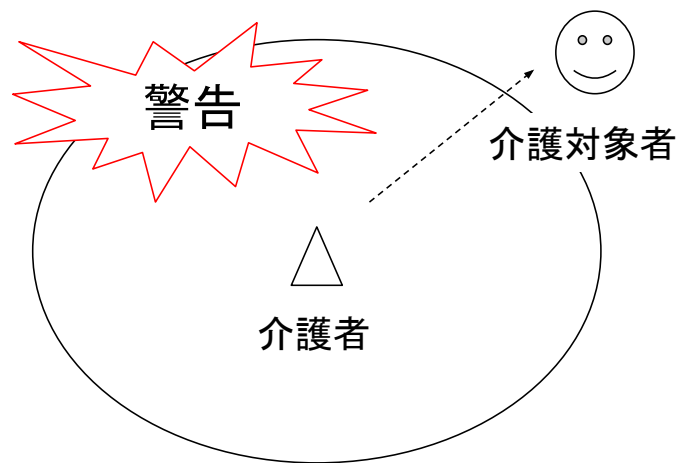


図 3.1: 見守りエリアからの逸脱管理

- タイムスタンプ
- 各自の見守りエリアにある発信機 ID
- 見守りエリアから逸脱した発信機 ID

以上の三つである。これらの情報を共有することで、同じアドホックネットワーク内にある他のスマートフォンが検知している発信機を知ることができ、見守りエリアを拡大することができる。

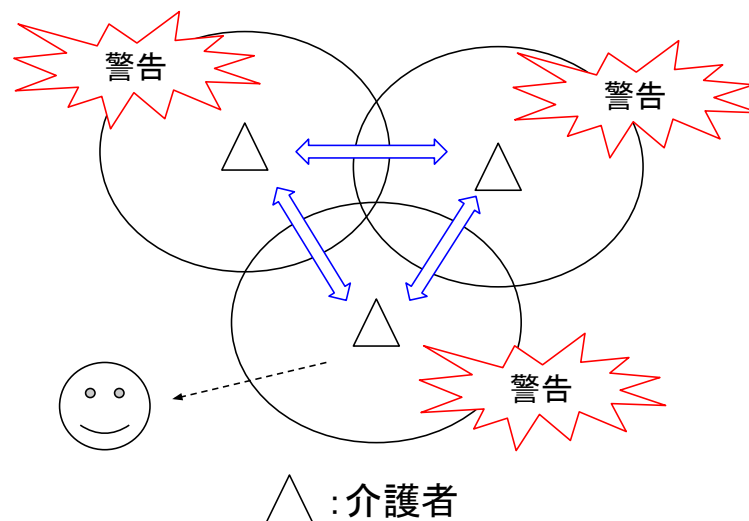


図 3.2: ビーコン情報の共有

第4章 設計と実装

4.1 提案手法

本研究では,P2P 型端末通信フレームワークである AllJoyn を用いて端末間のビーコン情報共有を行う。具体的には, 端末 A で周囲のビーコンを検知し, 端末 B にビーコンの UUID,RSSI などの情報を送信する。また, 自身で検知したビーコンの情報と, 他端末から送信されたビーコンの情報を保持する。今回の設計では, 同一アクセスポイント内に全ての端末があることを想定している。

4.2 開発環境

本研究では, 対象 OS を Android とし,Android OS で動作するアプリケーションを作成する。なお,Android 4.3 以上かつ Bluetooth4.0 以上に対応した端末を想定している。

- 開発環境
 - Eclipse 3.8 for Mac OS X
 - Android SDK Tools 23.0.5
 - AllJoyn Android v14.02
- 使用端末
 - TORQUEG01
 - ARROWS NX F-05F

使用した端末のスペックは以下の通りである。

OS	4.4.2
CPU	MSM8928 1.4GHz クアッドコア
ROM	16GB
RAM	2GB
Bluetooth	Ver.4.0
Wi-Fi	IEEE802.11a/b/g/n/ac

表 4.1: スペック:TORQUE G01

OS	4.4.2
CPU	MSM8974 2.3GHz クアッドコア
ROM	32GB
RAM	2GB
Bluetooth	Ver.4.0
Wi-Fi	IEEE802.11a/b/g/n/ac

表 4.2: スペック:ARROWS NX F-05F

4.3 AllJoynApp の概要

iBeacon から発せられるビーコンを受信し, AllJoyn を用いてビーコン情報を送受信する Android アプリである AllJoynApp を作成する.

4.4 設計

今回作成する AllJoynApp には, ビーコンを検知し, AllJoyn を用いてビーコン情報を送信する AllJoynClient と, ビーコン情報を受信する AllJoynService, インターフェースを記述した SimpleInterface という三つのプログラムがある. AllJoynService はバックグラウンドで動作するプログラムであり, AllJoynClient から呼び出されて起動する. 更に, FIND ボタンと SCAN ボタン, ListView といった UI 部品がある.

4.4.1 AllJoyn セッション

今回設計した AllJoynApp のセッションの流れは以下の通りである.

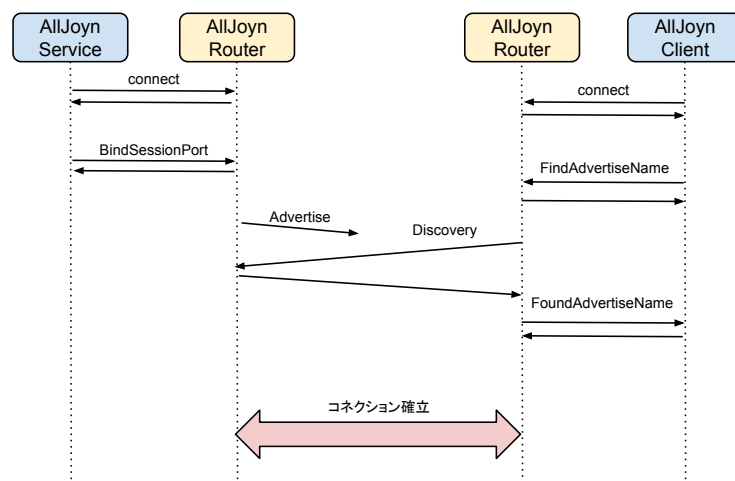


図 4.1: AllJoyn セッションの例

1. AllClient と AllJoynService は、それぞれの AllJoyn Router に接続する。
2. AllJoynService では、BindSessionPort API を用いてセッションポートやセッションオプションなどを決定する。
3. AllJoynService では、AllJoyn Router を通して任意の名前を Advertise してコネクションを待つ。
4. AllJoynClient では、AllJoyn Router を通して任意の名前を Discover する。
5. Advertise されている任意の名前と一致した場合、FoundAdvertiseName シグナルが AllJoynClient に送られる。
6. AllJoynClient が JoinSession API を用いて AllJoynClient と AllJoynService のコネクションが確立する。

4.4.2 iBeacon の検知

iBeacon は iOS 特有の機能であるが、Bluetooth Low Energy(BLE) を利用した技術であるため、BLE を受信することができる Android4.3 以上かつ Bluetooth4.0 に対応した端末なら iBeacon を検知することができる。iBeacon がブロードキャストしているバイトデータは 6~9 バイト目が固定されており、この値を参照して iBeacon かどうかを判断する。

データ領域	説明
1	1 ブロック目のバイト数
2,3	フラグ
4	2 ブロック目のバイト数
5	メーカー固有の AD type データ
6,7	会社コード (Apple のコードは 0x004C)
8	データタイプ (iBeacon は 0x02)
9	連なる iBeacon データのバイト数
10 ~ 25	UUID
26,27	major
28,29	minor
30	電波強度を求めるための 2 の補数

表 4.3: iBeacon データ配列

Android アプリ内でバイトデータを処理し、UUID や RSSI といったビーコン情報を抽出する。

4.4.3 プログラムファイル

AllJoynApp は,

- AllJoynClient.java
- AllJoynService.java
- SimpleInterface.java

の三つのプログラムファイルから成る。

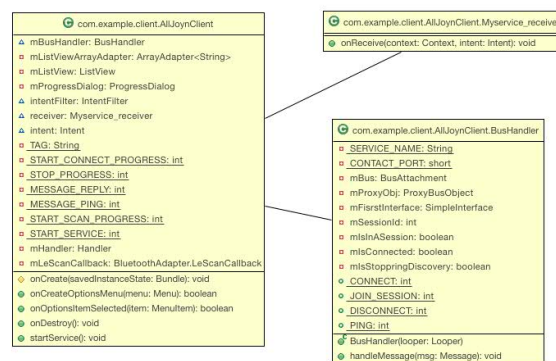


図 4.2: AllJoynClient クラス図

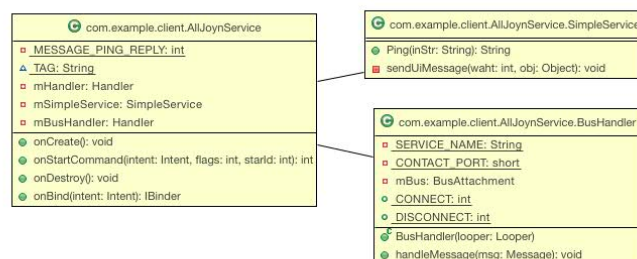


図 4.3: AllJoynService クラス図

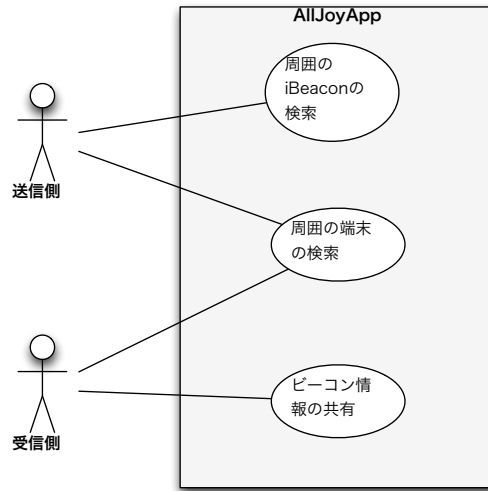


図 4.4: ユースケース図

4.5 実装

今回作成した AllJoynApp の処理を解説する。操作の手順としては

1. アプリケーションの起動
2. FIND ボタン押下
3. SCAN ボタン押下

となっている。

4.5.1 アプリケーションの起動

アプリケーションを起動した際の処理は以下の通りである。

1. AllJoynClient を起動する。
2. AllJoynClient は AllJoynService を起動する。AllJoynService はバックグラウンドで動作する。
3. 画面上に FIND,SCAN という二つのボタンを表示する。
4. AllJoynService は AllJoyn Router と接続し,Advertise する。

以下の図は、端末 A、端末 B で AllJoynApp を起動した様子を表している。

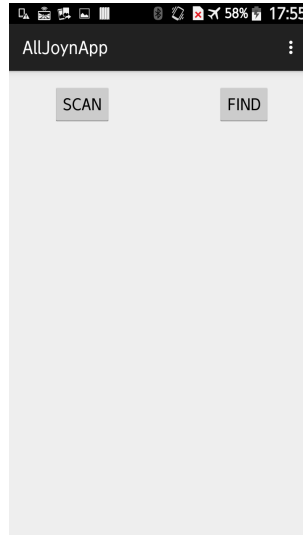


図 4.5: アプリケーション起動画面

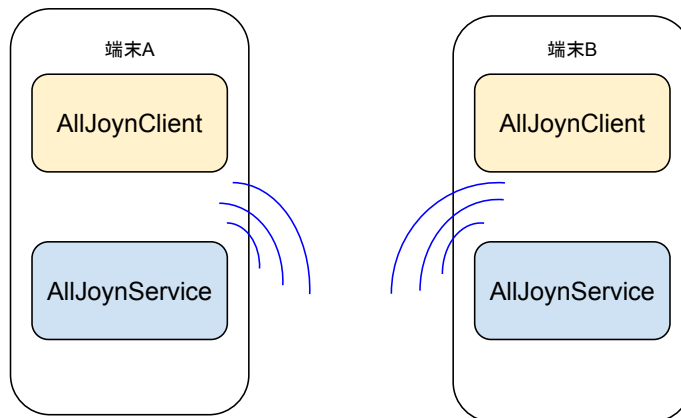


図 4.6: アプリケーション起動画面

4.5.2 FIND ボタン押下

FIND ボタンを押下した際の処理は以下の通りである.

1. 自身の AllJoynService を停止する.
2. AllJoynClient は自身の AllJoynRouter と接続する
3. 周囲の AllJoynService を Discovery する.
4. 検索している間, 画面上には'DISCOVERING' と表示される
5. AllJoynService を検知した場合, コネクションを確立し,'DISCOVERING' の表示をやめる.



図 4.7: FIND ボタン押下

以下の図では, 端末 A で FIND ボタンを押下した際の様子を表している. 端末 A では AllJoynService が停止し, 端末 A の AllJoynClient は端末 B の AllJoynService とコネクションを確立させる.

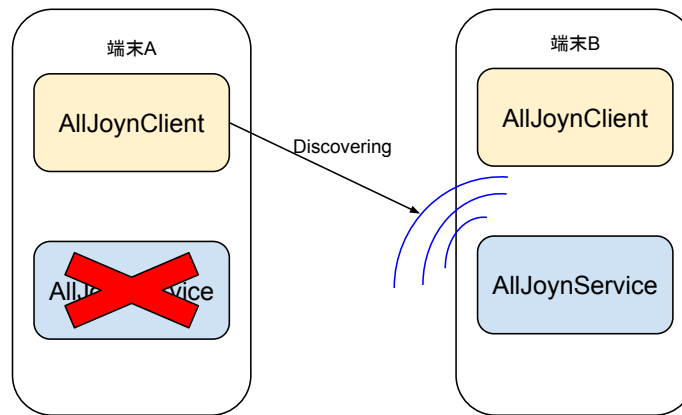


図 4.8: FIND ボタン押下

4.5.3 SCAN ボタン押下

SCAN ボタンが押下された際の処理は以下の通りである。

1. 二秒間周囲の iBeacon を検知し, ビーコン情報を画面上に表示する. iBeacon を検知しなかった場合, 画面上に 'iBeacon not found' と表示する.
2. AllJoynClient から, 接続している AllJoynService にビーコン情報を送信する.
3. ビーコン情報を受信した AllJoynService は, 自身の AllJoynClient にビーコン情報を送信する.
4. AllJoynClient は, 受信したビーコン情報を画面上に表示する.
5. 自身の AllJoynService を起動する.

以下の図は端末 A で SCAN ボタンを押下した際の様子を表している. 端末 A の AllJoynClient でビーコンを検知し, 端末 B の AllJoynService にビーコン情報を送信する. 端末 B の AllJoynService は自身の AllJoynClient にビーコン情報を送信し, 画面上に表示する.

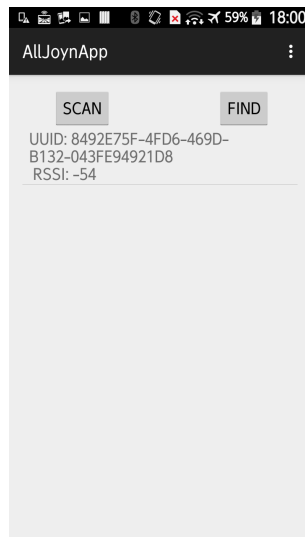


図 4.9: SCAN ボタン押下

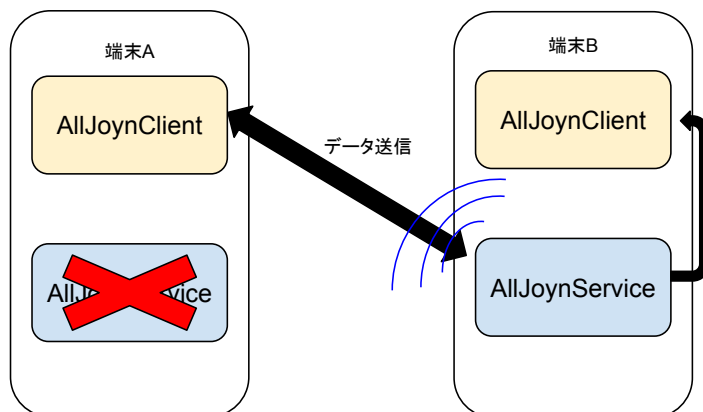


図 4.10: SCAN ボタン押下

4.6 検証

今回設計・実装を行った AllJoynApp の検証を行う。検証する項目は以下の通りである。

項目 1 端末で iBeacon を検知し, 他端末にビーコン情報を送信できる。

項目 2 端末で iBeacon の離脱を検知し, 他端末に離脱を通知する。

項目 3 複数の iBeacon 存在下でも検知し, それぞれのビーコン情報を他端末に送信できる。

これらの項目の検証を, 琉球大学工学部棟一号館 607 教室にて行った。使用した機材は, AllJoynApp の検証機器として ARROWS NX F-05F(以下端末 A)TORQUE G01(以下端末 B), ビーコン発信機として iOS 端末である iPhone5S 二台 iPhone6 一台 (以下発信機) の計三台を用いた。

- 項目 1

項目 1 の検証手順は以下の通りである。

1. 端末 A 及び端末 B を無線アクセスポイント 'ie-ryukyu' に接続する。
2. 端末 A 及び端末 B で AllJoynApp を起動する。
3. 発信機一台を起動する。
4. 端末 A で FIND ボタンを押下し, 'DISCOVERY' の表示が消えたことを確認し SCAN ボタンを押下する。

端末 A で取得したビーコン情報が端末 B の画面上に表示された。(画像は後で撮って貼ります。)

- 項目 2

項目 2 の検証は項目 1 の手順に以下の手順を加える。

1. 発信機を停止させる。
2. 端末 A で FIND ボタンを押下し, 'DISCOVERY' の表示が消えたことを確認し, SCAN ボタンを押下する。

端末 A 及び B の画面上に 'iBeacon not found' と表示され, iBeacon の離脱を検知し, 他端末に通知することができた。

- 項目 3

項目 3 の検証は項目 1 の手順に以下の手順を加える。

1. 発信機 3 台起動させる。
2. 端末 A で FIND ボタンを押下し, 'DISCOVERY' の表示が消えたことを確認し, SCAN ボタンを押下する。

端末 A 及び B の画面上に iBeacon 三台分のビーコン情報が表示された。

4.7 考察

今回の設計・実装で,Android 端末で iBeacon を検知し,ビーコン情報を他端末に送信し,ビーコン情報を共有することができた. 機材の関係上, 端末 2 台,iBeacon 発信機 3 台までしか検証できなかったが, この検証で一般家庭や小規模の介護施設などでの見守りできることが示された. また, アクセスポイントを経由して通信を行っているので同一アクセスポイント内なら距離を選ばず通信することができる.

FIND ボタンを押下して SCAN ボタンを押下するまで自身の AllJoynService を停止しているため, その間はビーコン情報の受信が出来なくなってしまうという問題点がある. また, 今回の実装は端末が同一アクセスポイント内にあるという想定で行ったため, 屋外での見守りには対応していない. そのため, 現状では見守り可能な環境が限定されている.

「検知出来なくなった iBeacon はどれか」といった表示をしていないため, 現状では大規模の見守りには不向きであるといえる.

第5章 結論

5.1 結論

本研究では、スマートフォン同士のアドホックネットワークの構築と、情報共有手法の実装を目的とした。検証の結果から、情報共有手法の実装という目的は達成できたといえるが、アドホックネットワークの構築は達成することができなかった。

5.2 今後の課題

検証で得られたデータから、現在のアプリケーションに追加すべき機能や改善点を検討する。

- アドホックネットワーク

スマートフォン同士でのアドホックネットワークを構築し、屋外での見守りにも対応する必要がある。そのためには、AllJoyn 使用環境の検証や AllJoyn の拡張をしていく必要があると考えられる。

また、今回の検証では機材の関係上端末二台での通信だったが、端末の数を増やした場合の検証をしていく必要がある。

- 情報共有手法

今回の設計では、端末が二台しかないという関係上、二台間での情報共有に留まっている。そのため今後、端末が三台以上での情報共有を想定し設計・実装をしなければならない。

- アプリケーション全般

周囲にある iBeacon を全て検知するという設計で開発を行ったが、実際の利用を想定した場合、アプリケーションの画面から検知する iBeacon を指定することができるようにする必要がある。

また、Discovery や iBeacon のスキャンの処理をボタンを押下することによって開始しているが、ユーザーが操作することなく自動で処理を開始するよう設計する必要がある。

ビーコン情報の共有に注力してアプリケーションを作成したため、UI 部分が洗練されていない。今後、より検証を重ねて UI を洗練させていく必要がある。

- 検証

端末や iBeacon 発信機の数をもっと増やして、検知できる iBeacon の数などの検証を進めていく必要がある。また、実際の利用環境での検証や長期間の運用実験を行い、データを収集するべきである。

参考文献

- [1] LaTeX コマンド集
<http://www.latex-cmd.com/>
- [2] 厚生労働省
http://www.mhlw.go.jp/kokoro/speciality/detail_recog.html
- [3] 認知症高齢者の在宅介護における負担の現状
http://www.yafo.or.jp/letter/pdf_new/vol186_2.pdf
- [4] アドホックネットワーク
<http://e-words.jp/w/E382A2E38389E3839BE38383E382AFE3838DE38383E38388E383AFE383Bhtml>
- [5] ALLSEEN ALLIANCE
<https://allseenalliance.org/>
- [6] AllSeen Alliance wiki
<https://wiki.allseenalliance.org/>
- [7] BLE とは
<http://e-words.jp/w/BLE.html>
- [8] Bluetooth Low Energy とは
http://k-tai.impress.co.jp/docs/column/keyword/20110412_438953.html
- [9] iOS:iBeacon について
<http://support.apple.com/ja-jp/HT202880>
- [10] iOS7 で iBeacon を使用してみよう!
<http://www.gaprot.jp/pickup/ios7/vol2/>
- [11] aBeacon~iBeacon を Android で受信する ~
<http://www.gaprot.jp/pickup/ibeacon/abeacon/>
- [12] ARROWS F-05F
<http://www.fmworld.net/product/phone/f-05f/>
- [13] TORQUE G01
<http://www.kyocera.co.jp/prdct/telecom/consumer/g01/>

謝辞

本研究の遂行，また本論文の作成にあたり，御多忙にも関わらず終始懇切なる御指導と御教授を賜りました宮里智樹助教に深く感謝いたします。

また，本研究の遂行及び本論文の作成にあたり，御教授と御指導を賜りました宮里研究室の早乙女さん，桃原さん，花城さんに感謝します。

また一年間共に研究を行い、暖かな気遣いと励ましをもって支えてくれた宮里研究室の饒平名長太君、呉屋桂基君に感謝します。

最後に、有意義な時間を共に過ごした情報工学科の学友、並びに物心両面で支えてくれた両親に深く感謝致します。

2015 年 2 月 16 日

新里亮太