社団法人　電子情報通信学会
THE INSTITUTE OF ELECTRONICS,
INFORMATION AND COMMUNICATION ENGINEERS

信学技報
TECHNICAL REPORT OF IEICE.

すれちがいフレームワークのためのBLEを用いた近接検知機構の実装と評価

田畑　亮馬†　林　亜梨沙†　徳永　清輝†　佐伯　幸郎†　柗本　真佑††
中村匡秀 †††

† ††† 神戸大学　〒 657–8501 神戸市灘区六甲台町 1–1
†† 大阪大学　〒 565–0871 大阪府吹田市山田丘 1–5
E-mail: †tabata@ai.cs.kobe-u.ac.jp, †††masa-n@cs.kobe-u.ac.jp

あらまし　モバイル端末による近接検知 (すれちがい) を利用したアプリケーション開発における様々な依存性解消のために，データを標準化して取り扱うすれちがいフレームワークを我々は提案している．本研究では，すれちがいフレームワークの近接検知部分の実装方法の違いがすれちがい検知の性能に与える影響を調べるために，Bluetooth Low Energy を利用した実験用すれちがいアプリケーションを開発し，電波を用いた近接検知について，パラメータを変更した際のすれちがい検知の挙動の変化を確認するための評価実験を行う．
キーワード　BLE, Android, すれちがいフレームワーク, 近接検知機構

Implementation and Evaluation of BLE Proximity Detection Mechanism for Pass-by Framework

Ryoma TABATA†, Arisa HAYASHI†, Seiki TOKUNAGA†, Sachio SAIKI†, Shinsuke MATSUMOTO††, and Masahide NAKAMURA†††

† Faculty of Engineering, Kobe University　Rokko-dai-cho 1–1, Nada-ku, Kobe, Hyogo, 657–8501 Japan
†† Faculty of Engineering, Osaka University　Yamada-oka 1–5, Suita, Osaka, 565–0871 Japan
E-mail: †tabata@ai.cs.kobe-u.ac.jp, †††masa-n@cs.kobe-u.ac.jp

Abstract　To achieve about various dependencies of application development using pass-by detection by a mobile device, We propose Pass-by Framework that handles data with standardized. In this study, we evaluate effects on performance of pass-by detection by differences in ways of implementation the sonar of pass-by framework. Therefore, we develop pass-by application used Bluetooth Low Energy as a first effort. We then perform an evaluation experiment becuse of confirmation of change pass-by detection behavior depends on the difference of parameters.
Key words　BLE, Android, Pass-by Framework, Proximity Detecction Mechanism

1. はじめに

近年，スマートフォンやタブレットなどのモバイル端末が世界中で急速に普及し，生活の価値向上やエンターテイメントなど様々な目的のモバイル端末向けアプリケーションが提供されている．最近では，モバイル端末が備えている Bluetooth や Wi-Fi，GPS などのデバイスを利用した，端末同士の近接やビーコンなどへの接近をきっかけとし新たな付加価値サービスを提供する，モバイル端末という特性を活かしたアプリケーションの利活用が進められている．本稿では近接検知機構により検出されるような，対象同士が一定時間内に一定距離以内へ接近する事象を**すれちがい**と呼び，すれちがいを利用する任意のシステムを**すれちがいシステム**とする．すれちがいシステムの利用例として，東京国立博物館では作品に近づくと自動的に解説が表示される「トーハクなび」[1] が提供されている．任天堂株式会社が開発・販売しているニンテンドー DS シリーズのすれちがい通信 [2] では，現実世界でのすれちがいをゲーム内での価値に変換することができる．

すれちがいシステムによる新しい価値の創出が期待される一方で，すれちがいシステムは，すれちがい検知のふるまいやすれちがいデータの表現方法，データの管理方法が技術や実装方

法に強く依存するために，アプリケーション開発者の負担が大きく，結果的に開発コストが増加する．また，異なる通信技術を用いるすれちがいシステム間には互換性が無く，実装の再利用性が低いといった問題がある．そこで，先行研究ではアプリケーション開発者を支援するためのフレームワークである**すれちがいフレームワーク** [3] を提案している．すれちがいフレームワークでは，端末や技術に依存する近接検知と，アプリケーションが利用するすれちがいデータの管理を分離し抽象化することで，開発コストを軽減することができる．また，近接検知の技術に依存せずすれちがいデータを扱うことができるため，異なる近接検知の技術を利用した端末同士のすれちがいを実現することも可能となる．

これまでのすれちがいフレームワークに対する研究では，主にデータ管理を対象としていた．しかし，すれちがいシステムの実現には，近接検知機構も実装しなくてはならない．近接検知における端末の挙動は，通信頻度や通信距離など，技術や実行環境によって異なるあらゆるパラメータに依存しており，パラメータの設定を変えることですれちがい検知の精度や端末の消費電流などが変化する可能性がある．ユーザのニーズを満たすすれちがいシステムを実現するために適切なパラメータをアプリケーション開発者が設定しなくてはならないが，開発者が各々で適切なパラメータを調べるのは開発コストが増大する要因となる．様々な近接検知機構に対応することを目標としているすれちがいフレームワークとしては，すれちがいフレームワークを開発者が利用する際に，あらゆる技術や実行環境におけるパラメータ決定の参考にできるデータを提示することにより，すれちがいフレームワークの研究の目的である開発コストの軽減をより大きく達成することができる．そこで，本研究では，近接検知機構によるすれちがいについて，近接検知の技術として近年普及してきている Bluetooth Low Energy(以下，BLE と表記する) を対象とし，技術依存のパラメータ設定とすれちがい検知のふるまいの相関関係について考察する．そのために，実際にすれちがいを実現するためのアプリケーションを開発し，パラメータと消費電流の相関関係，端末間の距離による受信電波強度の変化，移動速度によるすれちがい検知の精度と頻度への影響を調べるための評価実験を行い，実験データを分析し考察する．

2. 準　　備

2.1 すれちがいシステム

すれちがいとは，複数の対象がある一定の範囲内にある一定の時間存在することとする．また，すれちがいを何らかの近接検知機構を用いて検出し，その事実を記録する，または，データを送受信するといった任意の動作をするためのシステムをすれちがいシステムとする．すれちがいを検出するためには，端末間の相対的な位置情報からすれちがいを検出する方法と，緯度・経度といった絶対的な位置情報からすれちがいにあてはまる端末の組み合わせを検出する方法がある．BLE や Wi-Fi のアドホック・モードによるすれちがいの検出は前者にあたる．また，絶対的な位置情報を利用する技術としては，GPS や Wi-Fi のインフラストラクチャー・モードなどが挙げられる．

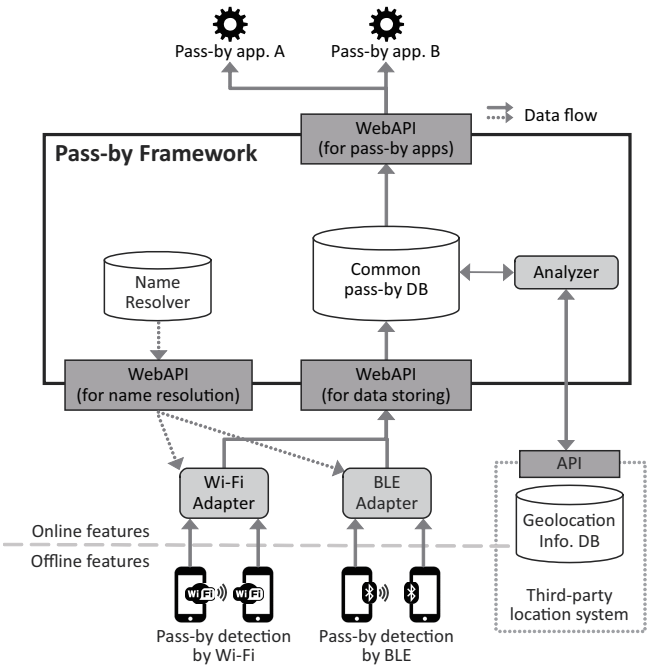

図 1　すれちがいフレームワーク

2.2 すれちがいフレームワーク

すれちがいシステムは，すれちがい検知のふるまいやすれちがいデータの表現方法，データの管理方法などが近接検知技術や実装方法に強く依存するために，アプリケーション開発者の負担が大きく，開発コストが増加する．我々の研究グループでは，開発コストを抑えるための手法としてすれちがいフレームワークというものを提案している [3]．このフレームワークでは，端末や通信技術に依存する近接検知機構と，アプリケーションから利用するすれちがいデータの管理を分離し抽象化することで，アプリケーション開発者はどのユーザーがどの端末を利用するかを意識せずに設計・開発することが可能となる．

すれちがいフレームワークのアーキテクチャを図 1 に示す．フレームワークは，近接検知機構 (以下，ソナーと表記する) とデータ管理サーバの 2 つで構成されている．すれちがいは端末のソナーで検知され，それぞれの近接検知技術ごとに作られた Adapter に送信され，そこで標準化されたすれちがいデータへ変換される．標準化されたデータは WebAPI を通し，Common Pass-by DB に記録される．また，GPS などによる位置情報は Geolocation Info. DB に送られる．Analyzer は，Common Pass-by DB から送られてくるすれちがいデータと，Geolocation Info. DB から送られてくる位置情報から，未知の新たなすれちがいを推論し，新たに生成されたすれちがいデータを Common Pass-by DB に送る．アプリケーションから Common Pass-by DB に記録されているすれちがいデータを利用する際は，フレームワークが提供している WebAPI を利用することでアプリケーション側がすれちがいデータを取得する．

2.3 BLE

BLE [4] は, Bluetooth4.0 規格の一部であり, 3.0 以前の Bluetooth(以下，クラシック Bluetooth と表記する) と比較して省電力での通信が可能である．BLE はクラシック Bluetooth と同じ 2.4GHz 帯の周波数を利用するが，クラシック Bluetooth との後方互換性を持たない．クラシック Bluetooth には同時接続が 7 台までという制約が存在するが，BLE には同時接続台数の制約は存在しない．規格において BLE の通信可能距離の最大と最小は規定されており，最大で 50m 程度，最小で 2.5m 程度とされている．

近年では，Bluetooth4.0 に対応している端末が市場に多く登場しており，すれちがいシステムが普及した時には，多くの端末がすれちがいシステムにおいて BLE を近接検知の技術として利用することが予想される．

2.4 本研究のスコープ

これまでのすれちがいフレームワークの研究は，主にデータ管理の部分が進められており，ソナーの部分の実装方法については未検討である．しかし，フレームワークを利用できるようにするには，ソナーを開発することが必須である．すれちがい検知の挙動は，ソナーの実装方法によって大きく変化する．電波を利用している BLE などの近接検知技術をアプリケーションで利用するためには，送信電波の強度や送信頻度など，すれちがいを検知する精度や電流消費量に関係するパラメータを設定する必要がある．パラメータを開発者が適切に設定しなければ，すれちがいをほとんど検知できないといったことや，バッテリーの消耗が早すぎるなどといった，ユーザがアプリケーションを利用する上での重大な欠点を生み出す可能性がある．

本研究では，すれちがい検知技術によって種類の異なるパラメータについて，その設定の違いによるすれちがい検知の挙動の差異を分析し，パラメータの設定がすれちがい検知に与える影響について研究する．本論文では第一歩として，BLE をすれちがい検知技術として利用する．最初に BLE を選択した理由としては，多くの端末が対応していて，すれちがい検知との相性が良く，将来的には代表的なすれちがい検知技術として利用される可能性が高いことが挙げられる．Android における BLE 通信のパラメータである「送信頻度」「送信強度」「受信頻度」について，バッテリー消費量や受信電波強度の観点から，すれちがい検知を実現するための適切なパラメータについて考察する．

3. BLE を用いた近接検知機構の実装

3.1 実験用すれちがいアプリケーションの必要要件

電波を利用した近接検知機構を実装する開発者は，実装方法の違いによる以下の E1〜E3 に示す検知能力への影響を考えてソフトウェアを作る必要がある．

E1：送受信に関係するパラメータが与える消費電流への影響
E2：対象間の距離と受信電波強度の関係
E3：対象の移動がすれちがい検知に与える影響

E1 は，開発者がアプリケーションによって消費される電流の大きさを予測できるようにするために調べる．E2 は，送信電波が発信源からどのように減衰して伝播するのかを開発者が知ることができるようにするために調べる．E3 は，速度が増加することによってすれちがい検知にどのような影響があるのかを開発者が知ることができるようにするために調べる．以上 3 つの要素を考慮に入れた上で，アプリケーションには以下の機能が必要になる．

R1：消費電流 (BLE 信号の送受信で消費する電流)
R2：距離測定機能 (端末間の距離と RSSI の依存関係)
R3：速度測定機能 (端末の速度と RSSI や受信回数の関係)

また，パラメータを変えながら BLE を送受信し，実験に利用できるようにするための最低限の機能として，以下の要素も必要になる．

R4：送受信の実行状態の切り替え
R5：電波の送受信に関係するパラメータの変更
R6：近接検知結果の記録

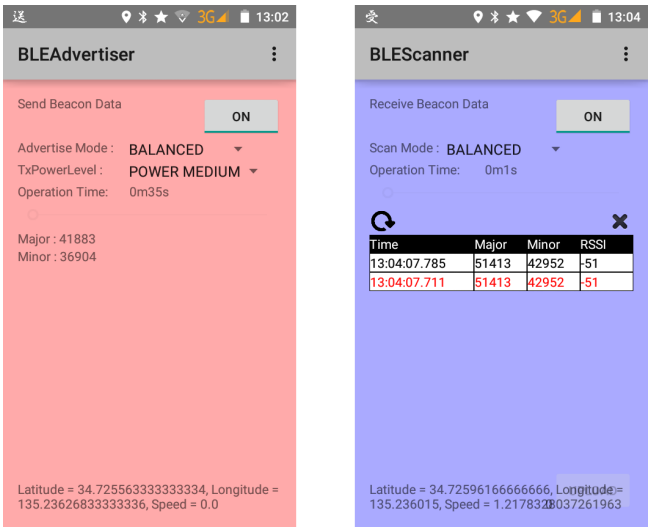

図 2　開発するアプリケーションの画面

3.2 開発するすれちがいアプリケーション

実験用すれちがいアプリケーションは以下の環境で開発する．

IDE：Android Studio 1.4
対象 OS：Android5.0 Lollipop 以上
対象端末：Nexus9 (HTC，2014)

送信アプリケーションでは，送信データのバイト列を iBeacon のデータフォーマット [5] に従って作成し，BLE で発信するように実装する．このフォーマットでは，利用アプリケーションを識別するための情報として UUID，端末それぞれを識別するための情報として major と minor と呼ばれる情報を持つ．受信側アプリケーションでは受け取ったバイト列を解析・選別し，UUID，major，minor を確認できるように設計する．

図 2 に開発するアプリケーションの画面を示す．左側が送信用アプリケーション (**BLEAdvertiser**)，右側は受信用アプリケーション (**BLEScanner**) である．送信・受信アプリ双方に共通する画面右上のボタンで信号の送信 (受信) の ON/OFF を切り替えることができる．両画面の左上には，パラメータの変更ができるコンボボックスがある．変更できるパラメータ

は，送信用アプリケーションでは AdvertiseMode(送信頻度) と TxPowerLevel(送信強度)，受信用アプリケーションには ScanMode（受信頻度）を変更できる．画面下部には，現在の位置情報である Latitude(緯度) と経度 Longitude(経度)，Speed(速度) が表示される．この数値は，端末の位置情報の変化を検知する度に更新され，端末の位置情報と速度が時間と紐づけられて端末内のローカルファイルに保存される．送信側と受信側で記録された同一時刻の位置情報から，端末と端末の距離を計算することが可能である．また，受信用アプリケーションのみの機能として，信号を受信する度にその信号を受信した時間 (HH:MM:ss.SSS)，端末の固有識別子として利用する major，minor(それぞれ 0〜65535)，受信時の電波強度 RSSI(dBm) を画面に出力する機能があり，リアルタイムに信号の受信状況が把握できる．

消費電流に関しては，Android アプリ開発用に提供されている直前の一定時間の平均消費電流を測定する機能 [6] を利用して，平均消費電流を測定するアプリケーションを上記のものとは別のアプリケーション (**PowerConsumptionMonitor**) として開発する．このアプリケーションは，一定時間ごとに平均消費電流を測定して端末の画面に表示する機能を持ち，端末がスリープ状態のときもバックグラウンドで動作可能である．

4. 評価実験

4.1 消費電流

開発するすれちがいアプリケーションによっては，消費電流を考慮に入れながら実装しなければならない場合が存在する．例として，災害時の安否確認システムにすれちがいを利用するシチュエーションを考えた場合，災害時は端末の充電が容易にはできない状況が想定できるため，充電をしなくてもできるだけ長時間稼働できるアプリケーションの設計が要求される．この場合，アプリケーション開発者はすれちがいアプリが連続で稼働できる時間を知っておく必要がある．BLE 通信のパフォーマンスの良さと端末のバッテリー消費量はトレードオフである可能性が高く，パラメータの違いによる消費電流の差も知ることが必要である．

4.1.1 実験方法

Android OS 環境下においては，送信側のパラメータとして，送信強度には U.LOW(最弱)，P.LOW(弱)，P.MIDUUM(中)，P.HIGH(強) の 4 段階，送信頻度には LOW POWER(低電力)，BALANCE(バランス)，LOW LATENCY(低遅延) の 3 段階のパラメータが設定できる．送信アプリケーション (BLEAdvertiser) においては，送信強度 4 パターンそれぞれについて送信頻度 3 パターン，計 12 パターンのパラメータの組み合わせで送信実行時についてそれぞれ 5 回測定し，平均を求めて比較する．受信側のパラメータとして，受信頻度には LOW POWER(低電力)， BALANCE(バランス)，LOW LATENCY(低遅延) の 3 段階のパラメータが設定できる．また，受信アプリケーション (BLEScanner) においては，送信用アプリケーションによって発信された信号だけではなく，周囲の機器から送信された BLE 信号を検知するたびコールバック関数を呼び出す仕様となっている．そのため，BLE 信号を受信する量とコールバック関数の内容によって消費電流に変化が起きる可能性がある．それを確かめるために，本実験では 3 つそれぞれ異なるのシチュエーションについて，3 段階の受信頻度を変更しながら消費電流をそれぞれ 5 回測定し，平均を求める．今回は以下の S1〜S3 のシチュエーションで実験する．

S1：BLE 信号が多く検知できる環境 (周辺に 3 個の BLE 発信機器) で，UUID を解析するコールバック関数を実装

S2：BLE 信号が多く検知できる環境 (周辺に 3 個の BLE 発信機器) で，コールバック関数に何も記述しない

S3：BLE 信号がほとんど検知できない環境 (周辺に BLE 発信機器がない) で，UUID を解析するコールバック関数を実装

消費電流の測定には **3.2** で開発した PowerConsumptionMonitor を利用する．このアプリを Nexus9 で利用すると，直前の 11.25 秒の平均消費電流を計算することができる．また，スリープ中でもバックグラウンドで消費電流を 12 秒ごとに測定して記録することができる．消費電力の測定時は，PowerConsumptionMonitor と，BLEAdvertiser または BLEScanner のどちらか以外は，全てのタスクを実行しないように設定し，ディスプレイをオフにした状態で，端末の CPU が安定した状態のときに消費電流を測定する．

表 1　BLE 送信機能の Nexus9 における消費電流

Advertise Mode	Tx Power Level	CC(μA)	Increment(%)
LOW POWER	U.LOW	125.2	0.4
	P.LOW	437.8	1.3
	P.MIDIUM	562.8	1.6
	P.HIGH	2187.6	6.3
BALANCE	U.LOW	1625.2	4.6
	P.LOW	1593.8	4.6
	P.MIDIUM	1594.2	4.6
	P.HIGH	2250.0	6.4
LOW LATENCY	U.LOW	2187.6	6.3
	P.LOW	2594.0	7.4
	P.MIDIUM	3062.6	8.8
	P.HIGH	3187.6	9.1

4.1.2 実験結果

BLE 送信機能の消費電流測定の結果を表 1 に示す．表中の CC は，何もしていないときに消費する消費電流 (34999.4μA，以下，自然消費電流と記述) との差分 (つまり，BLE の送受信のために消費された電流) であり，Increment は自然消費電流と比較した増加率である．実験結果から，BLE に費やす消費電流は，最大で 9%程度の大きさだった．

BLE 受信機能の消費電流測定の結果を表 2 に示す．検知した BLE 信号の UUID を解析する場合は，検知するだけで何もしない場合と比較して最大で 2 倍程度の消費電流を示している．また，信号の検知回数によっても消費電流は変化する．また，BLE 信号が多く検知できる場所では，送信機能と比較して，各頻度における消費電流は高い数値を示している．

表 2　BLE 受信機能の Nexus9 における消費電流

Situation	Scan Mode	CC(μA)	Increment(%)
S1	LOW POWER	10281.6	29.4
	BALANCE	36437.8	104.1
	LOW LATENCY	70281.4	200.8
S2	LOW POWER	6500.2	18.6
	BALANCE	18875.2	53.9
	LOW LATENCY	50844.2	145.3
S3	LOW POWER	2655.6	7.59
	BALANCE	4125.4	11.8
	LOW LATENCY	17625.2	50.4

4.2　対象間距離と RSSI の関係

Android で BLE 信号を送信する際の送信強度には 4 段階のパラメータがあるが，BLE 信号などの電波には，電波強度 (RSSI) と呼ばれるパラメータが存在する．電波は，発信源から距離が離れるにつれて RSSI が減衰していくものである．BLE 送信側は，送信 (電波) 強度の 4 段階のパラメータを変更することで，信号の送信範囲などをある程度制御することができる．受信側では，受信した信号がどれくらいの強度を持つのかを確認することができる．

本実験では実際に 2 つの端末を用い，端末間の距離を変化させながらすれちがいを行い，受信時の電波強度を確認し，どのように電波強度が減衰していくのかを確かめる．本実験で得られたデータをアプリケーション開発者に提示することで，開発者は各送信強度における，すれちがいに含める最大の距離であるすれちがい距離 L を知ることができる．受信強度でフィルターをかけることによって，任意の最大すれちがい距離 L でのすれちがいを実現することも可能となる．

4.2.1　実 験 方 法

実験は **A** と **B** の 2 人 (それぞれ端末 1 台所持) で行う．**A** は所持端末の BLE 送信アプリケーションで送信頻度を低遅延にして送信を開始し，端末を高さ 1m で持ったまま静止しておく．**B** は，**A** から一定距離離れた地点に立ち，受信アプリケーションで受信頻度を低遅延にして受信を 20 秒間行う．この試行を，**A** と **B** の対象間距離が 1m と 5m〜100m の 5m ごとの距離の場合についてそれぞれ試行する．同様の実験を，送信側アプリケーションの送信強度のパラメータ 4 つすべてについてもそれぞれ行う．

4.2.2　実 験 結 果

図 3 に，4 つの送信強度のパラメータについて，1m と 5m〜100m の 5m ごとの距離で電波強度を測定した実験結果を示す．4 つの曲線は，それぞれのパラメータについての実験結果を対数近似したものである．BLE で通信可能な最大距離は公称では 50m 程度とされているが，P.HIGH と P.MIDIUM において 50m を超える距離でのすれちがい検出ができた．最大すれちがい距離 L を 5m として設計した場合，送信強度 U.LOW(最弱) での平均 RSSI は約-82dBm となっており，アプリケーション開発者が送信強度を最弱で設定する場合に，受信時 RSSI が-82dBm 以上の場合をすれちがいとして判定するプログラムを実装すれば，最大すれちがい距離 L を 5m とする近接検知機構を構築することが可能となる．

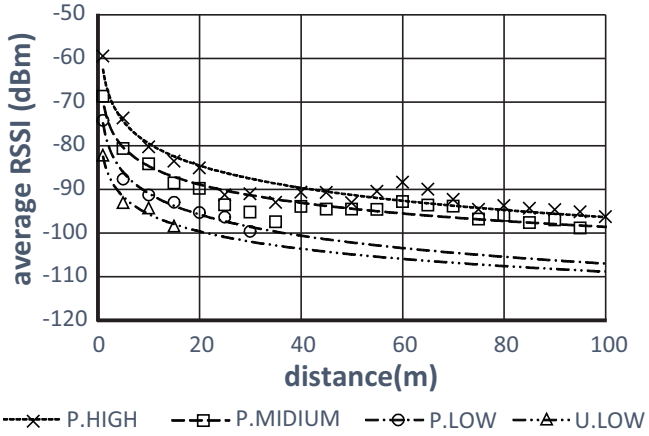

図 3　端末間の距離と平均受信電波強度 (RSSI)

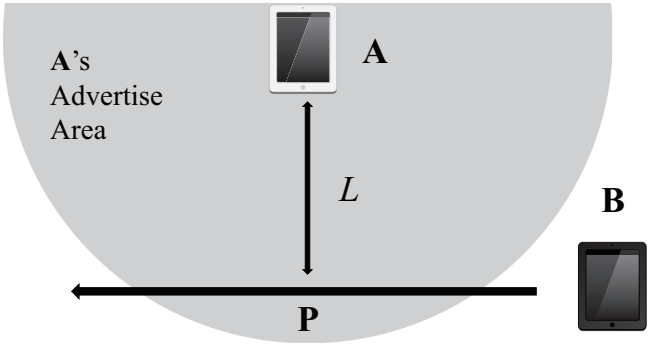

図 4　実験 **4.3** の測定方法

4.3　速度と検知精度・頻度の関係

すれちがいをする一般的なシチュエーションを想定した場合，端末は速度を持って移動している場合が考えられる．速度とすれちがいの際の受信電波強度や検知精度には依存関係がある可能性が考えられる．もし，自転車とのすれちがいを想定したアプリを開発するときに，徒歩でのすれちがいを想定したアプリと同様の設計で正しくすれちがいを検知できるとは限らない．開発者は，端末の移動速度がすれちがいにどう影響するかを知るべきである．

4.3.1　実 験 方 法

本実験では，最大すれちがい距離 L の範囲を 5m と設定する．実験は **A** と **B** の 2 人 (それぞれ端末 1 台所持) で行い，**A** の持つ端末では送信側アプリケーションを扱い，**B** の持つ端末では受信側アプリケーションを扱う．今回の実験では，双方の端末をリュックサックに入れたうえで行う．実験方法を図解したものを図 4 に示す．**A** は静止して，**B** はまず **A** の持つ端末の BLE 通信圏外まで移動し，**A** から 5m 離れた地点を **P** とすると，**B** は **AP** に垂直かつ地点 **P** を通る直線上を，BLE 通信圏外から逆側の通信圏外まで一定速度で移動する．実験に使う移動手段として，徒歩 (約 5km/h)，ジョギング (約 10km/h)，自転車 (約 20km/h) の 3 パターンで各 10 回の実験を繰り返す．計 30 回の試行の終了後，**A** と **B** の持っている端末を入れ替え，同様の実験を行う．なお，実験における送信強度は，**B** が

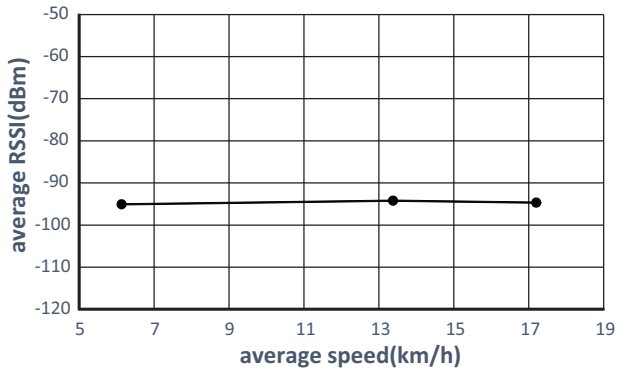

図 5　各移動速度の平均電波強度 (RSSI)

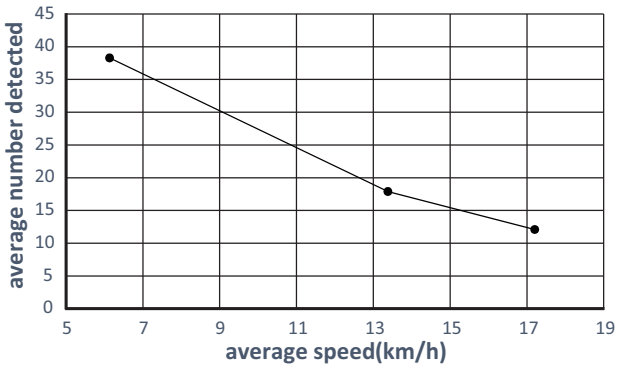

図 6　各移動速度の平均受信回数

地点 **P** で静止しているときにすれちがいを確実に検知できる中で最も小さい強度とする．

4.3.2　実 験 結 果

すれちがい検知の成功率は，3 つの移動手段すべてにおいて 90%を超える確率ですれちがいを検知することができた．以下，受信側が動いた場合の実験結果を示す．各速度において受信電波強度 (RSSI) を分析した結果を図 5，平均受信回数を図 6 に示す示す．RSSI の平均値はどの移動手段を利用した場合でも殆ど変化が無いということが分かった．結論としては，相対速度による電波強度への影響は無視できるほど小さい．また，受信回数は速度にほぼ比例する形で減少の傾向がみられる．また，送信側が動いた場合でも，同様の結果を示すことを確認した．

4.4　考　　察

BLE 送信の消費電流はどのパラメータの組み合わせにおいても，自然消費電流と比べて 10%以下となった．よって，できるだけ省電力のアプリケーションを開発する場合において，送信強度を低いパフォーマンスで抑えることは，アプリケーションの省電力化に大きな効果は得られないと考えられる．また，受信用アプリケーションでは BLE 信号を検知するたびにコールバック関数が実行されるため，コールバック関数の内容と BLE 信号の受信頻度の 2 つの要素によって消費電流が変わることが分かった．周囲に BLE を利用したすれちがいアプリケーションを利用している人が多い状況だと，それに伴って消費電流も増加すると考えられる．アプリケーションの開発時には，周囲に人の多い環境で利用することが想定されるかどうかまで考慮するのが良いと考えられる．

端末の移動速度に関しては，端末が速度を持って移動していても，受信電波の強度に大きな変化はないことが判明した．おそらく，同様の実験をさらに大きな速度で行ったとしても，受信電波の強度は大きく変化はしないと考えられる．しかし，受信回数は速度が上がるにつれて確実に減少している．これは，速度が速くなればなるほど，受信側が送信側のすれちがい可能領域にいる時間が短くなることが原因だと考えられるが，これによりすれちがいが可能な速度には限界があることが考察できる．

今回の実験はすべて Nexus9 を利用している．ここで示した実験は，Nexus9 以外の端末で実験する場合，同一の結果を得られない可能性が考えられる．今後，すれちがいフレームワークにおける参考データとして本研究で得られた実験結果を活用するならば，端末を変えて同様の実験を行い，実験結果の端末との依存関係を調査する必要がある．

5.　お わ り に

本研究では，アプリケーション開発者がすれちがいフレームワークにおけるソナーを実装するにあたり，各近接検知技術ごとにパラメータ設定による近接検知の挙動の差異について参考になるデータを開発者に提示するために，Android において BLE を利用する際の参考データを評価実験によって示した．

今後の課題として，他の Android 端末や iOS の端末を使用して同様の実験を行い，より汎用的なデータを示すことや，Wi-Fi のアドホック・モードなどの異なる近接検知機構の技術についても最適なパラメータを見つけ出すことが必要である．また，参考データをもとに割り出した適切なパラメータを用いて実際に災害時の安否確認システムなどに利用されるシチュエーションを考察し，シミュレーションを行うなどの方法で，実装後の汎用性などを実際に確かめる必要がある．

謝辞　この研究の一部は，科学技術研究費（基盤研究 B 26280115，15H02701，若手研究 B 26730155，萌芽研究 15K12020）の研究助成を受けて行われている．

文　　献

[1] “東京国立博物館 - アプリ 「トーハクなび」について,” http://www.tnm.jp/modules/r_free_page/index.php?id=1467.
[2] “すれちがい通信といつのまに通信,” https://www.nintendo.co.jp/3ds/hardware/features/network.html.
[3] 林亜梨沙，まつ本真佑，佐伯幸郎，中村匡秀，“すれちがいフレームワークにおける標準データ生成・蓄積機構の検討,” 電子情報通信学会技術研究報告，第 115 巻，pp.19–24，Dec. 2015.
[4] “Bluetooth low energy — bluetooth development portal,” https://developer.bluetooth.org/TechnologyOverview/Pages/BLE.aspx.
[5] 上原昭宏，“ビーコンのデータ構成,” iBeacon ハンドブック，pp.28–30，達人出版会，2014.
[6] “Measuring device power — android open source project,” https://source.android.com/devices/tech/power/device.html.