

BLE通信を用いる スマートフォンOS用マルチホップ通信技術の提案

吉崎 徹^{1,a)} 内藤 克浩²

概要: 本研究では、スマートフォン間の通信方式として、Bluetooth Low Energy (BLE) 通信を用いるマルチホップ通信技術の提案を行う。近年、イベント会場など多数の加入者が集中する場所、および災害時などにセルラー回線のサービス品質が著しく劣化することが報告されている。これは、セルラーシステムの1基地局が処理可能な通信性能には上限があり、特定の基地局に通信が集中したためである。そこで、基地局の負荷を軽減する手法として、Device to Device(D2D) と呼ばれるデバイス間通信方式を用いることで、セルラー回線に依存することなくデータ通信を可能とする方式が提案されている。一方、D2Dを利用したとしても、セルラーシステムによる制御が必要であり、基地局との通信が困難なほど混雑している状況および、基地局自体が破損した場合などには課題が残されている。本研究における提案方式では、メッセージ交換などの少量のデータ交換を想定し、省電力無線通信方式である BLE を用いたマルチホップ通信技術の通信手順を設計し、プロトタイプ実装を用いた基礎検討を行う。動作検証では、デバイス間通信技術(シングルホップ通信)と目的のデバイスへの経路制御技術(マルチホップ通信)の基礎動作を確認する。

キーワード: マルチホップ通信, D2D, BLE, スマートフォン, メッセージ交換

1. はじめに

スマートフォンの普及に伴い、様々なアプリケーションがスマートフォン常にインストールされるようになった。また、これらのアプリケーションは頻繁に通信を行うことから、セルラーシステムで処理されるトラフィックが急激に増加しており、より効率的な通信方式が望まれている。

スマートフォンで想定されるデータ通信では、セルラーシステムの基地局および、近隣のWi-Fi基地局と通信を行うことにより、インターネットとの通信を実現している。一般にスマートフォンで想定されるセルラーシステムでは、通信相手の位置には依存せず、常にセルラーシステムを経由する通信を行う。そのため、近隣に存在するスマートフォン同士の通信であっても、セルラーシステムが利用されている。しかし、このような場合、数百メートル離れた基地局にアクセスし、携帯電話事業者の交換機を経由するよりも、デバイス同士が直接通信することにより無線リ

ソースを節約することで、セルラーシステムの混雑の緩和が期待できる [1], [2], [3]。携帯電話業者の観点からも、近距離での通信であれば、デバイス間通信(D2D)を活用することで、通常のインフラによる通信回線の渋滞を避けることができる。また、災害時には基地局や電話局の物理的な破損、停電などの対策としても有効であると考えられる。このD2Dを活用した通信方式として、3GPP LTE Release 12では「LTE D2D」の標準化が進められている [4]。

LTE D2Dは基地局を介さずに端末同士が通信を行う、LTEによるデータ通信方式である。現在、実用化されているD2Dの仕組みはごく近い場合に使用するものである。一方、LTE D2Dは通信可能な距離が大きく異なり、約1キロメートルと非常に長い距離での通信が可能となる。本来、携帯電話と基地局の通信は数キロメートル離れていても通信が可能である。LTE D2Dは、LTEの仕組みをそのまま利用することで、非常に離れた場所同士の通信が可能になる。通信相手を見つける場合は、通信キャリアの基地局を活用する「ネットワークアシストD2D」が想定される [5], [6]。

インフラによる通信回線の渋滞回避や災害時などの基地局などの破損の対策として有効であると先に述べたLTE D2Dであるが、通信相手を見つける際にセルラー回線に依

¹ 愛知工業大学経営情報科学研究科
Faculty of Departments and Graduate Schools, Aichi Institute of Technology, Nagoya, Aichi 464-0807, Japan

² 愛知工業大学情報科学部
Faculty of Information Science, Aichi Institute of Technology, Toyota, Aichi 470-0392, Japan

a) ytoru@pluslab.org

存してしまうという問題がある。また、セルラー回線に依存しない LTE D2D の議論もされているが、あらかじめ設定を各端末が保持しておく必要があり、通信回線の遮断が起きてしまう状況が発生した後は、通信相手を見つけることさえ出来ない。スマートフォンのみで行えるデータ通信には、「Wi-Fi」と「Bluetooth」が挙げられる。Wi-Fi を用いる場合、多くの場合は Wi-Fi 基地局にスマートフォンが接続することによりデータ通信を行う。最近では、Wi-Fi Direct と呼ばれる、Wi-Fi を用いて対応機器同士を直接接続する技術が広がっている [7]。しかし、不特定多数の端末と接続することは意図されていないため、接続に手順が必要である。

Bluetooth とはデジタル機器の近距離間データ通信に用いられる無線通信技術である。Bluetooth では、対応した機器同士が無線通信でデータのやり取りができる。スマートフォンはもちろん、イヤホンやスピーカー、キーボードやマウスなど様々な製品が対応している [8]。Bluetooth は、Wi-Fi と比べ大量のデータを送るのに向いていないことや、通信可能距離が短いことがデメリットとして挙げられる。Bluetooth の通信可能距離は数メートルから 100 メートルと設定されており、送信電力により変化する。しかし、省電力設計であるスマートフォンにとって消費電力が少ないという大きなメリットも挙げられる [9]。また、Bluetooth には使用するチャンネル数を絞り、より消費電力を抑えた Bluetooth Low Energy(BLE) という技術が存在する。この技術は、従来の Bluetooth と比べ約 1/3 の消費電力に抑えられている。

省電力設計のスマートフォンに向けた Bluetooth であるが、通信可能距離が短いというデメリットを残している。この解決策としてマルチホップ通信が挙げられる [10], [11], [12], [13]。これは、バケツリレーのように情報を伝送していく方法である。先に述べたように、スマートフォンの所有率は年々増加している。そのため、Bluetooth とマルチホップ通信を用いることで、セルラー回線に依存することなく、長距離での D2D が実現できると考えられる。

本研究では、通信手段として BLE を使用する。BLE とは、Bluetooth から使用するチャンネルをさらに絞り、より消費電力を抑えた技術である。この技術は従来の Bluetooth と比べ 1/3 の消費電力に抑えられている。また、マルチホップ通信を実現するルーティングプロトコルとして Mobile Ad hoc NETwork(MANET) で利用されている Ad hoc On-Demand Distance Vector(AODV) を採用する。AODV は、Reactive 型と呼ばれるルーティングプロトコルであり、データ通信の直前に経路構築を行う。さらに、各ノードが経路表を保持する形になるため、スマートフォンにおける実装に有効であると考えられる。

本研究では、スマートフォンによるセルラー回線に依存

表 1 Classic Bluetooth & BLE の通信仕様

	Classic Bluetooth	BLE
Frequency Band	2.4GHz	2.4GHz
Number of Channels	79	40
Data Rate	1~3Mbps	1Mbps
Max Output Power	100mW(20dBm)	10mW(10dBm)
Max Packet Size	1021Byte	47Byte
Transmitted Power	35mA	15mA

することのない D2D を実現するため、BLE 通信を用いたマルチホップ通信技術の提案を行う。プロトタイプとして、iOS 向けアドホックチャットアプリケーションの開発・評価を行う。結果として、メッセージ交換などの小容量のデータ通信を実現可能であることを明らかにする。

2. Bluetooth Low Energy(BLE)

2.1 BLE の概要

Bluetooth Low Energy(BLE) とは、Bluetooth の拡張仕様の一つであり、極めて少ない電力のみで通信が可能な方式である。BLE は 2010 年 7 月に発表された Bluetooth 4.0 規格の一部として策定された。免許なく使える 2.4GHz 帯 (ISM バンド) の帯域を用い、最大 1Mbps の通信が可能である。対応チップは従来の 1/3 程度の電力で動作することができ、ボタン電池一つで数年稼働することができる [14]。従来の Bluetooth(Classic Bluetooth) と BLE の主な通信仕様の比較を表 1 に示す。

BLE には、他の Bluetooth の通信方式のように Class という概念はない。しかし、最大通信距離を 50m、最大出力を 10mW と定義している。

2.2 BLE の構造

BLE の仕組みとして Generic Attribute Profile(GATT) と呼ばれるプロファイルがある。GATT は BLE における標準的なデータフォーマットであり、BLE 通信を用いてデータを送受信する方法や形式を決めている。これにより、異なる企業の BLE も GATT のフォーマットに従ってデータを送信している。結果として、データのやり取りを共通で行うことができる。このように BLE デバイス間の通信制御を行うのが GAP(Generic Access Profile) である。BLE で何らかの通信を行う場合は、GAP によってその振る舞いを調整し周囲のデバイスとの通信が可能になる。デバイス間でのデータのやり取りを規定したものがプロトコルと呼ばれている。BLE では、ATT、SM、L2CAP、LL、PHY というように階層化されているが、一般的にアプリケーション側からプロトコルを直接扱うことはない。

2.3 BLE のネットワークポロジ

BLE 通信方式として、片方向によるブロードキャスト型と双方向による接続型に大きく分けられる。それぞれの概

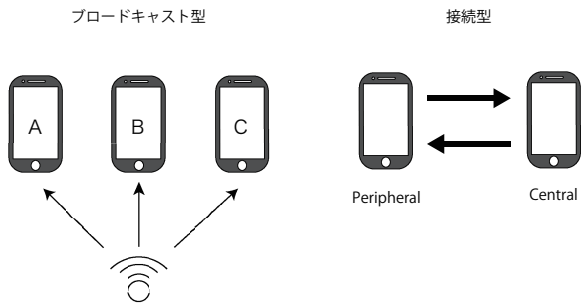


図 1 BLE のネットワークポロジ

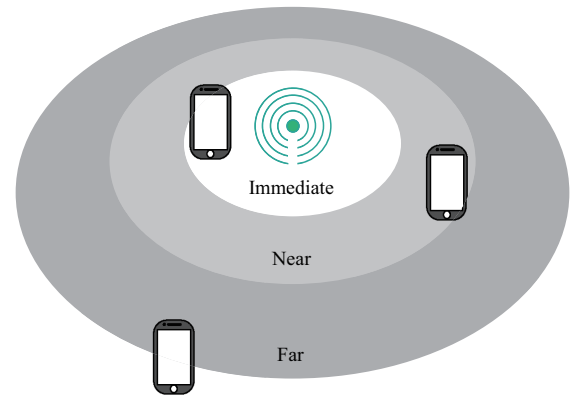


図 2 iBeacon の通信範囲

要図を図 1 に示す。

● ブロードキャスト型

ブロードキャスト型は、あるデバイスが周囲のデバイスに自身の存在や情報を伝えるために利用するものであり、片方向かつ対多数の形をとる。ブロードキャスト型の通信では、接続を確立させることなく通信が行われる。そのため、暗号化通信を行うことができない。

ブロードキャスト型では、BLEにより送信されるデータを Advertising Packet(または Advertising Data) と呼び、自身の情報を周囲に伝達する広告として機能する。なお、Advertising Packet を周囲に送信するイベントを BLE では Advertising、Advertising Packet を受信するイベントを Scanning と呼ぶ。

● 接続型

接続型は、通信する相手特定してから双方向での通信を行う。Bluetooth ver4.0 までは 1 対 1 と規定されていたが、Bluetooth ver4.1 以降では複数のデバイスと接続型トポロジーを持つことが可能になり、1 対多数の通信が可能となっている。

接続型トポロジーにおいて、他のデバイスから自身が発見されるように Advertising Packet を送信するデバイスを Peripheral と呼ぶ。Peripheral 側から送信された Advertising Packet を受信するデバイスを Central と呼ぶ。

2.4 iBeacon の概要

iBeacon とは、Apple が開発した BLE を利用したモバイル端末向け位置・近接検出技術である。iOS 7 から搭載されており、文字どおり「発信機(ビーコン)」として機能する。iBeacon が動作するための条件が 3 つ存在する。1 つ目はアプリケーションに実装することである。2 つ目は Bluetooth を ON にすることである。iBeacon は BLE を使用することが前提となっている。BLE は低消費電力であるため、常に Bluetooth を ON にしている場合でもバッテリーの消費を抑えることが可能である。3 つ目は OS が対応していることである。iOS 7 以降でなければ使用不可能

である。Android では、Android 4.3 以降であれば iBeacon と同様の技術を使用可能である。

iBeacon では、ビーコン信号に 4 種類のデータを格納し発信している。iOS 端末がビーコン発信端末に近づくとビーコンを受信し、その中に含まれるデータから位置、距離、近接などの情報を割り出す。iBeacon が発信するデータは UUID、Major、Minor、RSSI の 4 種類で構成されている。UUID は 128 ビットの識別子であり、MAC アドレスや日付などから生成されている。一意性が確保されているため、iBeacon によるサービスの識別に使用される。

Major と Minor は 16 ビットの識別子であり、サービス提供者が自由に設定することができる。RSSI は受信信号強度のことであり、受信した電波の強さにより、ある程度の距離を推定することができる。

iBeacon の通信範囲の区分を図 2 に示す。iBeacon の特徴として、建物内であっても位置情報が取得できることが挙げられる。iBeacon は通信距離を Immediate(近接)、Near(近距離)、Far(遠距離) の 3 種類に設定することができる。ビーコン端末により左右されるが Immediate が数 cm、Near が 1m ほどの通信が可能であり、それ以上は Far となっている。

iBeacon には、アプリケーションの起動に関わらずバックグラウンドで処理を行うことができる。この機能は iOS 7.1 以降に実装されたもので、アプリケーションが起動していない場合でも、iBeacon を受信した iOS デバイスは 10 秒前後であるが、処理を行うことができる。この状態においても通信は可能であるので、バックグラウンド処理を行うサービスにおいても iBeacon の有用性は高いと言える。

3. Ad hoc On-Demand Distance Vector(AODV)

3.1 AODV の概要

AODV は Reactive 型と呼ばれるルーティングプロトコルであり、ユニキャスト、マルチキャスト、ブロードキャストをサポートしている。Reactive 型とは、通信の要求が発

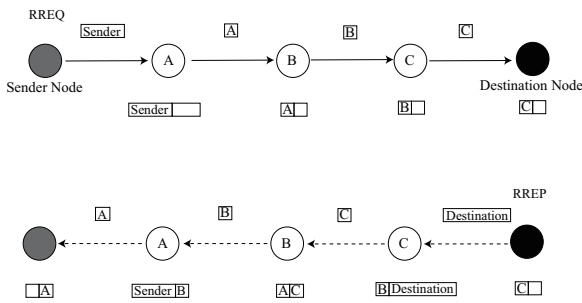


図 3 AODV の経路構築プロセス

生じた際に、周りのノードの存在を確かめるためメッセージを送り、ルーティングテーブルを作成するプロトコルである。通信の要求が発生し、ルーティングテーブルを作成後に実際の通信を開始するため、多少の時差が生じる。

AODV は各中継ノードがルータとして、終点アドレスに基づき次ホップを決定する経路制御方式を採用する。そのため、各ノードは経路表を作成し保持する。経路表は終点ノード、次ホップ IP アドレス、終点シーケンス番号、終点までのホップ数などから構成される。また、AODV ではシーケンス番号を用いることにより、中継ノードは最新のルーティング情報の探索を容易に行うことができる。

3.2 AODV の経路制御技術

AODV の経路構築プロセスを図 3 に示す。AODV は、大きく分けると Route Request(RREQ) と Route Reply(RREP) という 2 つの制御メッセージが存在する。RREQ は、送信要求が発生すると宛先ノードを検索するため周囲にフラッディングを行うパケットである。RREQ を受け取ったノードは、自身のノード情報を加えてさらにフラッディングを行う。もし受信履歴のある RREQ であれば破棄を行う。これを繰り返すことで宛先ノードへ RREQ が到達する。

RREP は、宛先ノードに RREQ が到達したことを知らせるためのパケットである。RREP は、RREQ が通ってきたルーティング情報を元に、RREQ の発信元へ戻っていく。RREP が発信元ノードへ到達することで、宛先ノードまでの情報を取得することができ、データを送信することができる。

AODV では、各ノードが経路表を保持することが特徴である。しかし、パケットには始点ノードから終点ノードまでの全てのルーティング情報を載せず、前ホップ、次ホップのみ情報を載せることで、どのノードへ送信すれば良いかを判断する。そのため、各ノードはルーティング情報の全体の構成はわからない。AODV はルーティングテーブルの有効期間が短期であるため、まず発信元ノードが保持するルーティングテーブルを参照する。発信元ノードに RREP が複数到達した場合は、より条件の良いルーティング情報であれば更新を行う。

4. 提案方式

4.1 提案方式の概要

本研究では、セルラー回線に依存しない通信技術として BLE を用いたスマートフォン OS 用マルチホップ通信技術の提案を行う。提案するマルチホップ通信における課題は大きく 2 つであり、隣接するスマートフォンへの通信技術と、目的スマートフォンまでの経路制御技術である。

1 つ目の課題は、いわゆるスマートフォン同士のシングルホップ通信を実現することである。スマートフォンでのシングルホップ通信を実現するためには、受信側がスキャンをしている状態で、送信側がデータの送信を行う必要がある。そのため、受信側のスキャンのタイミングをいかに実現するかが問題点となる。

2 つ目の課題は、シングルホップ通信で用いた通信技術を用いてマルチホップ通信を実現すればよい。マルチホップ通信では、複数の端末から送信される BLE を受信する側がどの様に処理をするかが問題点となる。これらを踏まえ、以下にシングルホップ通信技術とマルチホップ通信における経路制御のシステムモデルを説明する。

4.2 シングルホップ通信技術

本研究で提案するマルチホップ通信システムの各デバイス間の通信では、Bluetooth の BLE 技術を用いる。この通信では、送信側と受信側に分かれて処理が行われる。BLE 通信では、送信側が Advertising を行うタイミングと、受信側が Scanning を行うタイミングが一致することで通信が成立する。そのため、常に Advertising、Scanning を行うことが可能であれば問題がないが、iOS では常に Advertising、Scanning をさせ続けることは許可されていない。そのため、通信相手が検出できない場合、一定時間経過することで処理が中止となり、次の Advertising、Scanning までにインターバルを取るることとなる。また、一定間隔ごとであっても省電力の観点からは好ましくない。そのため送信側は、まずデータの送信とは別に BLE によるビーコンを発信する。このビーコンは Scanning をする必要なしに受信側が反応する iBeacon を採用する。iBeacon の送信後にデータの送信である Advertising を行う。受信側は、iBeacon を受け取ることをトリガーとして Scanning を開始する。この一連の動作により、シングルホップ通信を成立させる。

4.3 マルチホップ通信システムモデル

提案するマルチホップ通信方式のシステム概要を図 4 に示す。提案システムは MANET の一つである AODV に基づいて作られている。提案するマルチホップ通信方式は、始点ノード、中継ノード、終点ノードから構成されている。上記の図では、送信元である始点ノードを Sender Node、

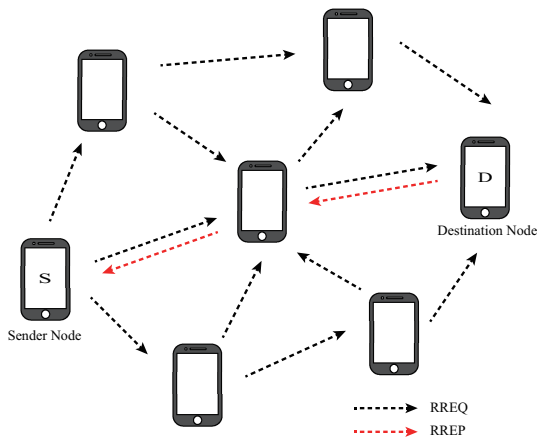


図 4 想定ネットワークモデル

送信先である終点ノードを Destination Node としている。

提案方式では、Reactive 型を採用しているためデータ送信の前に経路構築を行う。経路構築は、Sender Node が RREQ のフラッディングを行うことで開始される。RREQ を受け取ったノードは自身が Destination Node であるかを判断する。また、自身宛ではない場合は、通信相手を前ノードとして経路表に保存する。その後、そのノードから RREQ のフラッディングを行う。

受け取った RREQ が自分宛の場合は、経路探索が正しく動作したことを伝えるために、ルートリプライ (RREP) を Sender Node へ返す。RREP は、RREQ が辿った経路を戻すため、経路表の前ノードへ転送される。RREP を受け取ったノードは、自身に向けたものか判断する。自身に向けた RREP であった場合、通信相手を次ノードとして経路表へ保存する。その後、経路表に保存された前ノードへ向け、RREP を送信する。経路表に前ノードが保存されていない場合は、自身が RREQ の Sender Node であるため RREP の転送を終了する。Sender Node が RREP を受け取ることで、経路構築は完了となり、データ通信が開始される。

図 4 に想定するネットワークモデルを示す。想定するネットワークでは、同じ RREQ を重複して転送してしまう可能性がある。例えば、A-B-C と RREQ が転送される場合と、A-D-B-C という順に転送される場合があるとする。このような場合は、A からの同じ RREQ を B が二度転送する。この問題を解決するためリクエスト ID (RREQ ID) を用いる。RREQ の送信元は、毎回リクエスト ID を生成することにより、RREQ を識別することが出来る。また、同じ RREQ を二度転送してしまう問題に対する処理として、一度転送したリクエスト ID が再度転送されてくる場合、その時点でそのリクエスト ID を破棄する。

経路構築完了時のシグナリングを図 5 に示す。上記で記述したように Sender Node から RREQ を転送し、Desti-

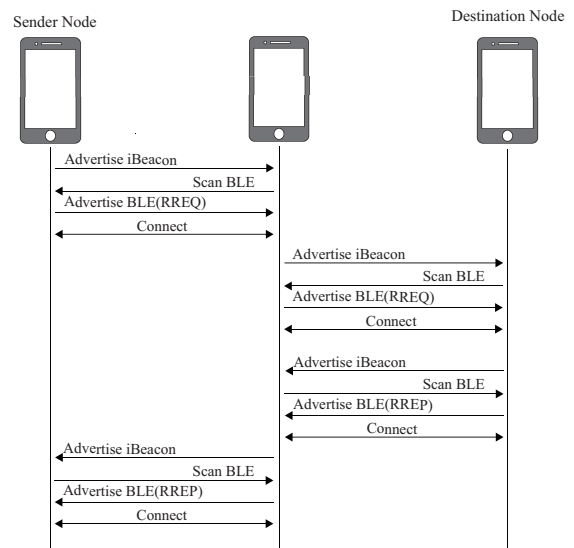


図 5 シグナリング

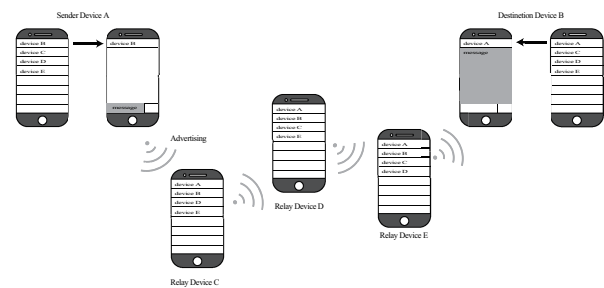


図 6 アプリケーション概要

nation Node が受信し、RREP を返信する。経路構築が完了した場合、Sender Node はその経路に沿ってメッセージを送信する。そのため、メッセージも図の様に転送され、同様に受信完了用のメッセージリプライが返信される。

各ノードには経路表として前ノード、次ノードが保存されている。この次ノードへメッセージを送信していく。さらにメッセージを受け取ったノードが次ノードへ転送することにより、Destination Node へメッセージが転送される。Destination Node からは、経路表の前ノードへメッセージリプライを送信する。メッセージリプライが Sender Node まで転送されることで、経路構築からメッセージ送信までの一連の流れが終了する。

4.4 アプリケーション実装

提案するマルチホップ通信方式を、アドホックチャットアプリケーションとして実装を行う。まず、アプリケーションの概要について説明を行う。

アプリケーションの概要を図 6 に示す。開発したアプリケーションは、BLE 通信により他のデバイスとメッセージのやり取りを行うものである。指定したデバイスが BLE 通信の範囲内に存在しない場合は、経路構築を行う。経路構築はメッセージを送る直前に毎回行う。経路構築完了

表 2 アプリケーションにおける使用パケット

パケットの種類	概要
0 (Message)	シングルホップ通信用メッセージ
1 (Route Request)	ルートリクエスト
2 (Route Reply)	ルートリプライ
3 (Multihop Message)	マルチホップ通信用メッセージ
4 (Message Reply)	マルチホップ通信用メッセージリプライ

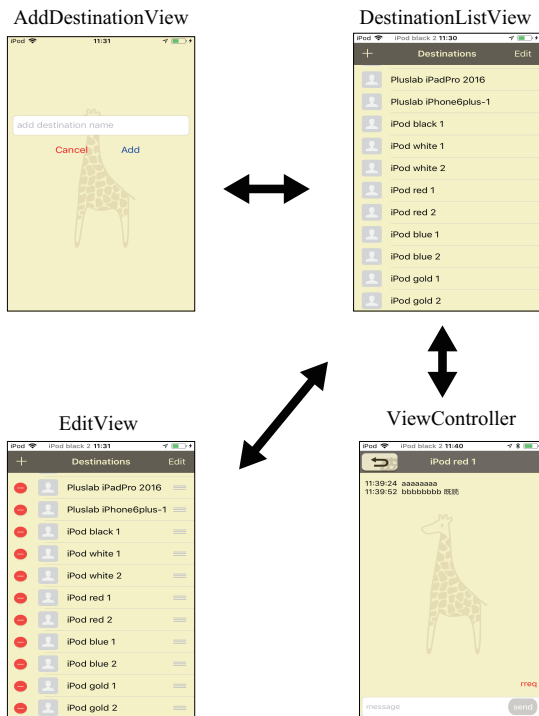


図 7 アプリケーションの画面遷移

後、図 6 の様にメッセージの転送を行う。

アプリケーションの画面遷移を図 7 に示す。始めにアプリケーションの起動を行うと DestinationListView が表示される。この画面は宛先一覧が表示されている。この画面から、宛先を選択することで ViewController へと画面遷移を行う。ここでは、選択した宛先との個別のチャット画面が表示される。送信するメッセージを入力し、送信ボタンをタップすることによりメッセージ送信、経路構築が行われる。DestinationListView からは、AddDestinationView、EditView へと画面遷移を行うことが可能である。AddDestinationView では、ラベルに名前を入力することで宛先の追加を行う。EditView では、現在登録されている宛先の削除を行うことが可能である。

続いて、シングルホップ通信の実装方法について説明を行う。まず、送信ボタンがタップされた場合に iBeacon が 1 秒送信される。iBeacon を検知したデバイスは、BLE の Scanning を行う。iBeacon の送信が終了すると BLE の Advertising が 5 秒間行われる。こうして Connection が確立しシングルホップ通信が実現する。

次に、マルチホップ通信の実装方法について説明を行う。マルチホップ通信では、AODV に基づいた経路構築方式を採用しているため、RREQ、RREP パケットを使用する。アプリケーション内では送信側 (Peripheral) と受信側 (Central) において処理が分かれている。実際のプログラムの処理について説明を行う。

● Peripheral

Peripheral では送信するパケットの制御を行う。こ

こでアプリケーション内にて使用するパケット種別を表 2 に示す。パケットのタイプ 0 は、シングルホップ通信におけるメッセージ用パケットである。このパケットは、経路構築を行う前に、送信する宛先がメッセージを受信可能であるのかを試す場合に用いる。タイプ 1 は、経路構築を行う RREQ 用パケットである。送信元がシングルホップ通信を行えないと判断した場合、周囲に向け送信する。タイプ 2 は、経路構築の完了を示す RREP 用パケットである。送信元が送信した RREQ を宛先が受信した際、宛先が送信元へ向けて送信する。タイプ 3 は、マルチホップ通信におけるメッセージ用パケットである。経路構築の完了後、送信元が再度メッセージを送るため送信する。タイプ 4 は、マルチホップ通信で送信したメッセージの返信用パケットである。送信元が送信したメッセージを宛先が受信した場合に、宛先から送信元へ向けて送信する。

● Central

Central では、Peripheral から送信されたパケットを受信した場合に、どのように振る舞うのか制御する。各パケットを受信した場合の処理を以下で説明する。まず、タイプ 0 を受信した場合は自身が宛先であるかを判断する。自身が宛先である場合は、パケットの送信元へ向け Write request を送信する。Write request とは、通信相手の Characteristic へ書き込みの許可を要請する処理である。Write request を受け取ることで、送信元は宛先がメッセージを受信したことを確認する。ここでは、Write request によって実際に相手の Characteristic へ書き込みを行うわけではなく、単に受信確認用に使用している。自身が宛先でない場合は受信したパケットを破棄する。

タイプ 1 を受信した場合は、自身が送信したルートリクエストをマルチホップ通信により受信している可能性がある。そのため、送信元が自身と一致する場合は、そのパケットを破棄する。次に、パケットの宛先が自身と一致しているのかを判断する。一致している場合は、タイプ 2 であるルートリプライを送信元へ向け、辿ってきた経路とは逆の経路で送信する。一致していない場合は、経路構築を進めるため、周囲のノードへルートリクエストを送信する。この際、ホップ数をインクリメントし、Time to live (TTL) をインクリメントする。TTL が 0 になってしまう場合はルートリ

クエストの送信を停止する。さらに、経路表を作成するために、送信してきたノードを前ノード (Previous Node) として保存しておく。ルートリクエストは、周囲のノードへ送信し転送されていくため、同じルートリクエストが重複する可能性がある。そのため、ID を用いてパケットの重複を防ぐ処理を行う。一度、受信した ID は破棄することで、ネットワークに存在するパケット数を減らすことが可能である。

タイプ2を受信した場合は、自身がこのパケットに対応するルートリクエストの送信元 (Sender) であるかを判断する。ルート自身が Sender である場合は、経路構築が完了しているため、経路に沿ってメッセージを送信する。自身が Sender でない場合、自身を中継するパケットであるかを判断する。中継する場合は、保存している経路表に基づき Previous Node ヘルトリプライを送信する。この時、送信してきたノードを次ノード (Next Node) として保存する。また、パケットの Next Node の値を自身に、Previous Node をタイプ1の際に保存した値へ置き換えた後、送信する。自身が中継しないメッセージの場合は、そのパケットを破棄する。

タイプ3を受信した場合は、自身が中継するメッセージであるのかを判断をする。タイプ3はマルチホップ通信によるメッセージであるため、この時点で経路構築は終了している。そのため、宛先が自身である場合は経路表に従い Next Node へメッセージを中継する。この時、Next Node が自身である場合は、経路の終点であると判断しメッセージの受信処理を行う。そして、メッセージ受信を知らせるメッセージリプライ (Message Reply) を逆経路で送信する。宛先が自身でない場合、そのパケットを破棄する。

タイプ4を受信した場合は、自身が Previous Node と一致するのかを判断する。次に、一致する場合は Sender であるのかを判断する。ここでの Sender とは、このパケットに対応するメッセージの送信元であるのかを意味する。自身が Sender と一致する場合は、メッセージが宛先へ受信されたことを示すため、受信完了の処理を行う。Sender と一致しない場合は、このパケットを中継する。その際は、パケットの Previous Node の値を自身が保持する経路表の値に更新した後、送信する。自身が Previous Node と一致しない場合、そのパケットを破棄する。

4.5 評価

本研究において作成したアプリケーションについて評価を行う。作成したアプリケーションは、BLE 通信をマルチホップさせることで、メッセージの転送を行うものである。実験では、ホップ数による時間の変化と、メッセージ

表 3 実験諸元

使用デバイス	iPod Touch
OS 環境	iOS 11.0
デバイス台数	最大 5 台
通信障害物	なし

表 4 実験結果 (経路構築)

使用デバイス数	経路構築時間 [sec]
3 台	11.9
4 台	17.6
5 台	22.8

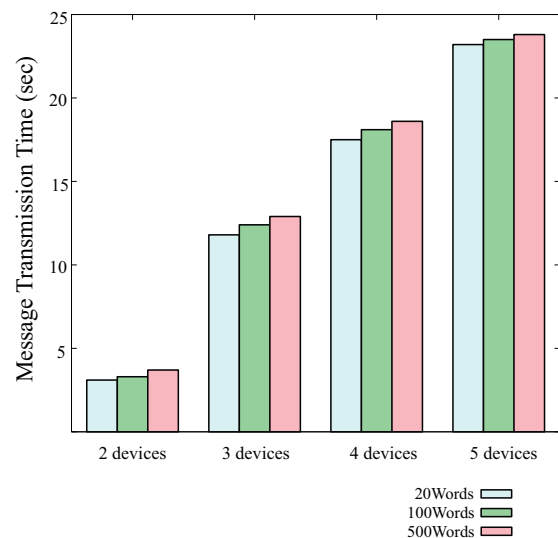


図 8 実験結果 (メッセージ転送)

の長さによる時間の変化の計測を行った。実験諸元を表 3 に示す。

実験では、送信元デバイスが経路構築を開始すると時間の計測が行われる。各デバイスが BLE を送信、受信するごとに時間経過を算出していく。こうして計測されたデータは、送信元が経路構築を開始した時間を元にどれだけ時間が経過したか計算を行う。

実験結果を表 4 と図 8 に示す。表 4 には、ホップ数の変化による経路構築時間を示す。使用するデバイスの数を増やし、ホップ数が増加すると 5 秒ずつ経路構築時間も増加していることが確認できる。これより、台数が増加すれば比例的に経路構築時間も増加していくことが考えられる。図 8 には、経路構築完了後に送信したメッセージの送受信時間を示す。デバイス 2 台での通信ではメッセージのやり取りに約 3 秒で終了している。デバイス 2 台の場合はシングルホップ通信になるため、受信完了用のパケットを必要としない。そのためデバイス 3 台の場合と比べても短い時間で終了している。また、メッセージの文字数を 20、100、500 のように変化させた場合にも、0.5 秒以下の変化しか見られなかった。デバイス 3 台、4 台、5 台の場合は、

経路構築時間と同様に約6秒ずつ増加していくことを確認した。

この実験から、BLE通信によるマルチホップ通信が実現可能であることを確認した。しかし、台数が増加するごとに時間が大きく増加するため、iBeaconやBLE通信の送信時間を可能な限り短くすることにより時間の短縮をはかる必要がある。

5. まとめ

本稿では、スマートフォンによるセルラー回線に依存することのないD2Dを実現するため、BLE通信を用いたマルチホップ通信技術の提案を行った。プロトタイプとして、iOS向けアドホックチャットアプリケーションの開発を行い、実際に複数台のデバイスにおける時間計測を行い評価した。また、BLE通信を用いることで、スマートフォン同士でのD2Dの実装とマルチホップ通信が実現可能であることを示した。

謝辞 本研究の一部は科研費(15H02697, 17K00142)、日比科学技術振興財団研究助成の助成を受けたものである。記して謝意を表する。

参考文献

- [1] Gao, Q., Zhao, R., Chen, W., Peng, Y., Guo, B. and Sun, S.: Radio resource management of D2D communication, *2014 IEEE International Conference on Communication Systems*, pp. 6–10 (online), DOI: 10.1109/ICCS.2014.7024755 (2014).
- [2] Yin, R., Yu, G., Zhang, H., Zhang, Z. and Li, G. Y.: Pricing-Based Interference Coordination for D2D Communications in Cellular Networks, *IEEE Transactions on Wireless Communications*, Vol. 14, No. 3, pp. 1519–1532 (online), DOI: 10.1109/TWC.2014.2368151 (2015).
- [3] Lee, N., Lin, X., Andrews, J. G. and Heath, R. W.: Power Control for D2D Underlaid Cellular Networks: Modeling, Algorithms, and Analysis, *IEEE Journal on Selected Areas in Communications*, Vol. 33, No. 1, pp. 1–13 (online), DOI: 10.1109/JSAC.2014.2369612 (2015).
- [4] Andrews, J. G., Buzzi, S., Choi, W., Hanly, S. V., Lozano, A., Soong, A. C. K. and Zhang, J. C.: What Will 5G Be?, *IEEE Journal on Selected Areas in Communications*, Vol. 32, No. 6, pp. 1065–1082 (online), DOI: 10.1109/JSAC.2014.2328098 (2014).
- [5] Xu, S. and Kwak, K. S.: Network Assisted Device Discovery for D2D Underlying LTE-Advanced Networks, *2014 IEEE 79th Vehicular Technology Conference (VTC Spring)*, pp. 1–5 (online), DOI: 10.1109/VTC-Spring.2014.7023002 (2014).
- [6] Nguyen, P., Wijesinghe, P., Palipana, R., Lin, K. and Vasic, D.: Network-assisted device discovery for LTE-based D2D communication systems, *2014 IEEE International Conference on Communications (ICC)*, pp. 3160–3165 (online), DOI: 10.1109/ICC.2014.6883807 (2014).
- [7] Khan, G. Z., Park, E. C. and Gonzalez, R.: M3-Cast: A Novel Multicast Scheme in Multi-Channel and Multi-Rate WiFi Direct Networks for Public Safety, *IEEE Access*, Vol. 5, pp. 17852–17868 (online), DOI: 10.1109/ACCESS.2017.2749482 (2017).
- [8] Bluetooth SIG: Bluetooth, Bluetooth SIG (online), available from <https://www.bluetooth.com/what-is-bluetooth-technology/how-it-works> (accessed 2017-10-24).
- [9] Putra, G. D., Pratama, A. R., Lazovik, A. and Aiello, M.: Comparison of energy consumption in Wi-Fi and bluetooth communication in a Smart Building, *2017 IEEE 7th Annual Computing and Communication Workshop and Conference (CCWC)*, pp. 1–6 (online), DOI: 10.1109/CCWC.2017.7868425 (2017).
- [10] Jung, C., Kim, K., Seo, J., Silva, B. N. and Han, K.: Topology Configuration and Multihop Routing Protocol for Bluetooth Low Energy Networks, *IEEE Access*, Vol. 5, pp. 9587–9598 (online), DOI: 10.1109/ACCESS.2017.2707556 (2017).
- [11] Mikhaylov, K. and Tervonen, J.: Multihop data transfer service for Bluetooth Low Energy, *2013 13th International Conference on ITS Telecommunications (ITST)*, pp. 319–324 (online), DOI: 10.1109/ITST.2013.6685566 (2013).
- [12] Pei, Y. and Liang, Y. C.: Resource Allocation for Device-to-Device Communications Overlaying Two-Way Cellular Networks, *IEEE Transactions on Wireless Communications*, Vol. 12, No. 7, pp. 3611–3621 (online), DOI: 10.1109/TWC.2013.061713.121956 (2013).
- [13] Zeng, Y., Hu, H., Xu, T. and Jia, B.: User Pairing Stability in D2D-Relay Networks, *IEEE Communications Letters*, Vol. 21, No. 10, pp. 2278–2281 (online), DOI: 10.1109/LCOMM.2017.2721364 (2017).
- [14] 堤修一, 松村礼央: iOS × BLE Core Bluetooth プログラミング (2015).