Vol.2016-MBL-80 No.13
Vol.2016-CDS-17 No.13
2016/8/25

遠隔地にあるBluetooth LE機器のシームレス接続手法の検証

岡田 真実[1]　鈴木 秀和[1]

概要：IoT（Internet of Things）デバイスが普及しつつある今日，宅内にはスマートフォンやタブレットで操作可能な様々な Bluetooth 搭載機器がある．しかし，Bluetooth は通信可能範囲が定められているため，ユーザは通信可能範囲を越えた場所から Bluetooth 搭載機器を操作することができない．また，規格によって操作できる機器が限られているため，ユーザは自身が保持する Bluetooth 端末のバージョンを意識する必要がある．これらの課題を解決するために，筆者らは遠隔地にある Bluetooth LE 機器に対してシームレスに接続する手法を提案している．提案手法を操作端末に実装することにより，ユーザは規格の違いを意識することなく，またどこからでも宅内の Bluetooth 搭載機器が仮想的に近傍に存在しているように認識し，遠隔制御することが可能となる．これまでに提案手法を Linux や Android に搭載されている Bluetooth プロトコルスタック BlueZ に実装してきた．本稿では提案手法の一部について動作検証を行ったため，その結果について報告する．また，Andorid 4.2（Jelly Bean）から新たな Bluetooth プロトコルスタックとして採用されている Bluedroid に対して，提案手法が適用できるか検討する．

Verification of Seamless Connection Method for Bluetooth Low Energy Devices in Remote Locations

Okada Mami[1]　Suzuki Hidekazu[1]

1. はじめに

Bluetooth は一般コンシューマに広く普及している最も代表的な近距離無線通信技術である．ユーザは Bluetooth 搭載のスマートフォンを使用することで，宅内にある Bluetooth 対応機器に接続し，機器の操作や機器が有するサービスを利用することができる．しかし，Bluetooth は通信可能範囲が物理的に制限されているため，ユーザは宅外から宅内にある Bluetooth 対応機器を直接操作することができない．また，Bluetooth には複数のバージョンが存在しており，バージョンの違いによっては機器同士が近隣に存在していても通信できない課題がある．

遠隔地にある Bluetooth 対応機器を操作する手法として，Bluetooth 以外のプロトコルを用いて Bluetooth 通信の結果を取得する方式と，Bluetooth 通信を遠隔地まで伝送する方式に大別できる．Bluetooth 以外のプロトコルを用いて Bluetooth 通信の結果を取得する方式では，インターネット上および宅内に専用のサーバとゲートウェイをそれぞれ設置し，外出先の操作端末はサーバを経由してゲートウェイに命令を送信することにより，ゲートウェイが代理で宅内の Bluetooth 機器と通信を行う．ゲートウェイは Bluetooth 通信の結果を各種プロトコルを用いて操作端末まで返信することにより，操作端末が遠隔地の Bluetooth 機器と通信することができる．この方式は多くのサービスで採用されており，例えば東芝 HEMS（Home Energy Management System）[1] では，フェミニティ倶楽部と呼ぶ Web サービスを利用して，宅内に設置する IT ホームゲートウェイに操作要求を行う．これにより，宅内に設置されている Bluetooth 搭載の ECHONET Lite 機器を遠隔制御することができる．この他にも，異種ネットワーク上の機器の相互接続を可能とする規格である PUCC（P2P

[1] 名城大学大学院理工学研究科
Graduate School of Science and Technology, Meijo University

Universal Computing Consortium）を用いた手法 [2–4] や，SIP（Session Initiation Protocol）を利用する UbiGate [5] などが提案されている．これらの手法は操作端末に専用の操作アプリケーションをインストールする必要があったり，宅内と宅外で使用する操作アプリケーションを使い分ける必要がある．また，インターネット上に専用のサーバを設置したり，クラウドサービスを利用する必要があり，仮にメーカやサービスプロバイダが遠隔制御サービスを終了してしまうと，宅内の Bluetooth 機器を遠隔制御できなくなってしまう．

Bluetooth 通信を遠隔地まで伝送する方式として，有線により Bluetooth ネットワークを拡張する手法 [6] や，UbiGate におけるゲートウェイ間をインターネットを通じて接続することにより Bluetooth ネットワークを拡張する UbiPAN [7] などが提案されている．これらの手法は操作端末が発信した Bluetooth の電波を近隣に設置した Bluetooth ゲートウェイ（以後，BGW）が受信し，BGW が Bluetoth の制御メッセージを有線ネットワークを利用して宅内に設置した BGW まで伝送する．宅内の BGW は制御メッセージに基づいて代理で Bluetooth 通信を行うことにより，遠隔制御を実現している．この方式はユーザが宅内と宅外の違い，すなわち自身の位置を意識することなく，常に同じ操作アプリケーションを利用することができる．しかし，操作端末の近傍に専用の BGW を設置したり，常に携帯する必要がある．

そこで筆者らはこれらの手法における BGW の機能を操作端末に組み込むことにより，遠隔地にある Bluetooth 機器をシームレスに接続できる手法を提案している [8]．この手法（以後，従来手法）は操作端末が近隣の機器を探索すると，遠隔地に存在する Bluetooth 機器を発見することができ，あたかもユーザの周囲に宅内の機器が存在しているかのように認識することができる．しかし，従来手法は Bluetooth 3.0 までしか対応しておらず，現在普及している BLE（Bluetooth 4.0 Low Energy）には対応していなかった．そのため，従来手法を拡張することにより，BLE に対応させたシームレス接続手法を提案してきた [9]．

本稿では，提案手法を実現するための具体的な仕様を示し，Linux PC を用いたプロトタイプ実装について述べる．また，実環境における動作検証および通信遅延の評価結果について報告する．また，Android における Bluetooth の実装について調査し，提案手法が適用できるか検討する．

以下，2 章で Bluetooth の概要，3 章で提案手法について述べる．4 章で実装について述べ，5 章で評価および検討結果を示し，6 章でまとめを行う．

2. Bluetooth

Bluetooth は**図 1** のように Bluetooth Host と Bluetooth Controller から構成されており，両者の間に定義されている

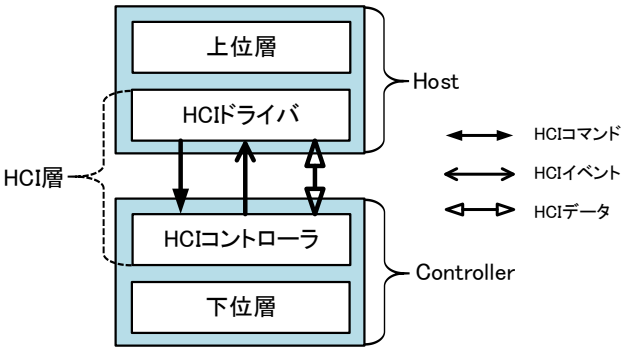

図 1 Bluetooth の構成
Fig. 1 Constitution of Bluetooth.

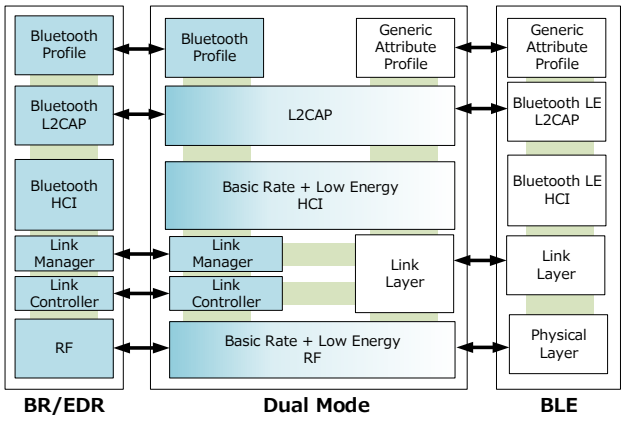

図 2 Bluetooth の規格の関係
Fig. 2 Relationship of Bluetooth architectures.

HCI（Host Controller Interface）層に含まれる HCI ドライバ（ソフトウェア）と HCI コントローラ（ハードウェアのファームウェア）間で HCI コマンドおよび HCI イベントと呼ばれる制御メッセージおよび HCI データを交換することで他の Bluetooth 機器との通信を実現している [10, 11]．本稿では HCI 層で交換されるこれらの制御メッセージおよびデータをまとめて，HCI メッセージと呼ぶ．Host が Bluetooth 通信を行う場合，Controller に対して HCI コマンドを発行する．Controller は順次コマンドを実行するが，その応答はイベントとして非同期で通知される．

また，Bluetooth で通信する際，機器の種類毎に策定されたプロトコルを使用するために Bluetooth プロファイルが定義されている．**図 2** に規格の関係について示す．Bluetooth はバージョン 2.0/2.1 である Bluetooth BR/EDR（Basic Rate/Enhanced Data Rate）と，バージョン 4.0/4.1/4.2 である BLE では，プロファイルを含むソフトウェアスタックが異なっており，BLE と BR/EDR は共通するプロファイルを持っていないため，BR/EDR 機器と BLE 機器の間で通信を行うことができない．なお，Bluetooth 機器がバージョン 4.0 以降で規定されている Dual モードをサポートすることにより，BR/EDR と BLE 双方のプロファイルを持つことができ，双方のバージョンに対応した Bluetooth

機器と通信を行うことが可能である．

2014 年 12 月に発表された Bluetooth 4.2 では新たに IPSP（Internet Protocol Support Profile）と呼ぶインターネット接続用プロファイルが定義された [12]．このプロファイルを利用することにより，BLE 機器はスマートフォンなどとペアリングすることなく，6LoWPAN（IPv6 over Low power Wireless Personal Area Networks） over BLE をサポートしたルータを介して直接インターネットに接続できる．しかし，ユーザが遠隔制御したい機器が Bluetooth 4.2 以上を搭載しており，かつ IPSP プロファイルを利用したアプリケーションが稼働している必要がある．

3. 提案手法

3.1 概要

図 3 に提案手法の概要を示す．提案手法では，他の既存技術と同様に宅内に BGW を設置するが，Bluetooth スタックにモジュールを追加することにより，操作端末の近隣に専用のハードウェアを不要とし，かつユーザは宅内と同じ通常の Bluetooth アプリケーションにより遠隔地の Bluetooth 機器と通信することを実現する．以後，ユーザが操作する操作端末を CD（Control Device），操作となる遠隔地にある Bluetooth 機器を RD（Remote Device）と呼称する．

2 章で述べたとおり，Bluetooth は，同じプロファイルを持つ機器同士でしか通信できないという制約があるが，Bluetooth スタックにおける Host と Controller の間で HCI メッセージを交換する HCI 部分は共通である．そのため，従来のシームレス接続手法で実装されていた Filter モジュールを拡張し，BR/EDR 仕様の HCI メッセージだけでなく，BLE 仕様の HCI メッセージを新たにフックする．CD はフックした HCI メッセージをインターネットを通じて宅内の BGW へ転送し，BGW が代理で宅内の RD と通信を行う．RD との通信により発生する HCI メッセージおよび HCI データを逆の手順で CD まで返信し，Filter モジュールを介して CD の Host へ戻すことにより，RD との通信を実現する．すなわち，CD の Host と BGW の Controller をペアにすることにより，CD の近隣に RD が存在しているかのように認識することができる．提案方式により，下記 3 点を実現することができる．

- ユーザは Bluetooth 通信の通信範囲を意識することなく，宅内の RD を操作できる．
- ユーザが Bluetooth の規格の違いを意識することなく，様々な Bluetooth 機器を操作できる．
- ユーザが RD の操作を行う際，自身の位置に応じてアプリケーションを切り替える必要がない．

3.2 BLE の HCI メッセージ

BR/EDR と BLE は HCI 層でメッセージを処理する点

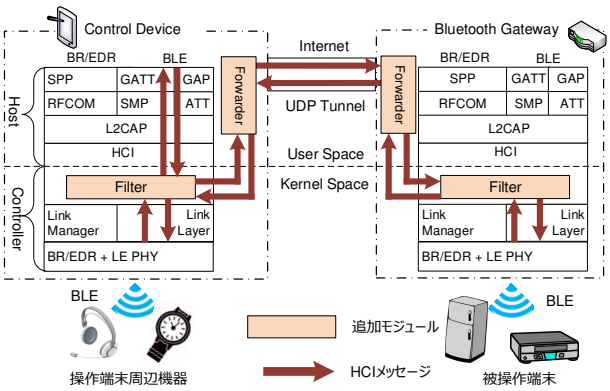

図 3 提案手法の概要
Fig. 3 Overview of the proposed system.

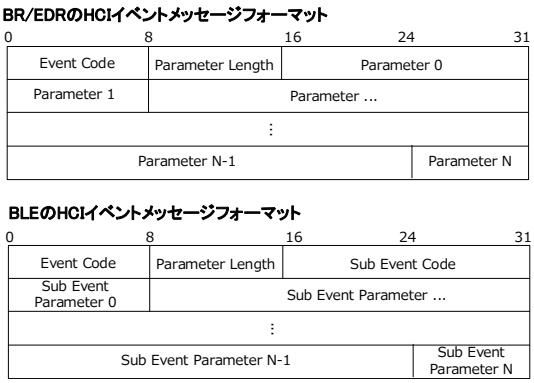

図 4 BR/EDR と BLE の HCI イベントメッセージの違い
Fig. 4 Difference of HCI events messages between BR/EDR and BLE.

は同じであるが，HCI メッセージのフォーマットが異なる．そのため，Filter モジュールが BLE の HCI メッセージをフックする際，メッセージフォーマットの違いを意識する必要がある．BLE と BR/EDR では HCI イベントのフォーマットのみが図 4 のように異なる．Bluetooth は HCI メッセージを処理する際，メッセージの 1 バイト目に記載されている Event Code で HCI メッセージの種類を判別している．BR/EDR の Event Code は `0x04` となっており，BLE では `0x03E` となっている．このことから，CD が HCI イベントを受信する際，Event Code を識別するだけで，BR/EDR と BLE の HCI イベントメッセージを判定することができる．なお，HCI コマンドのメッセージフォーマットに違いはないため，BGW は規格の違いを意識することなく HCI コマンドを受信し，処理することができる．

3.3 通信シーケンス

本節では宅外に存在する CD が宅内に存在する RD を発見する流れを示す．

(1) CD は Bluetooth 機器を利用するアプリケーションを起動すると，周辺に存在する Bluetooth 機器の探索処理を行う．ここで CD の Host と Controller 間で交

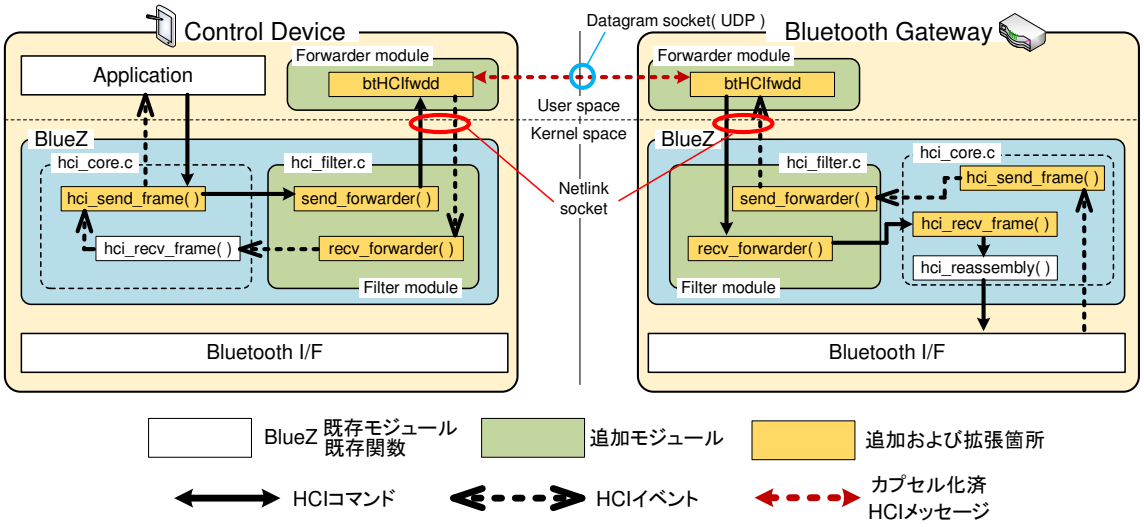

図 5　モジュール構成
Fig. 5　Module configuration.

換される BR/EDR または BLE の HCI メッセージを CD の Filter モジュールでフックする．

(2) フックした HCI メッセージは，CD の Forwarder モジュールに渡され，インターネット上に事前に構築した UDP トンネルを用いて，BGW の Forwarder モジュールへ送信される．

(3) BGW は受信した HCI メッセージを自身の Filter モジュールへ渡し，HCI メッセージの内容に基づいて BR/EDR または BLE の Controller 部へ処理を依頼する．これにより，BGW は宅内に存在する RD を発見することができる．

同様の仕組みを用いて，CD と RD は接続および通信を行う．

4. 実装

4.1 システム構成

図 5 に提案方式のモジュール構成を示す．本稿では提案手法を実装するにあたり，Linux に搭載されている Bluetooth 標準プロトコルスタックである BlueZ を用いて実装を行った．提案方式における Filter モジュールおよび Forwarder モジュールは，Linux カーネル空間に実装されている BlueZ から呼び出されるカーネルモジュールとして，また，ユーザ空間で動作するデーモン `bthcifwdd` としてそれぞれ実装した．なお，操作端末 CD には BLE を，また BGW には BLE，BR/EDR 双方と接続可能な Dual モードを搭載する．

4.1.1 Filter モジュール

Filter モジュールは BLE 用に拡張が必要なモジュールである．Filter モジュールでは，CD の Application から BlueZ の `hci_send_frame()` 関数に送信される HCI コマンドのコピーをフックし，モジュール内の `send_forwarder()` 関数へ渡す．BR/EDR と BLE の HCI コマンドメッセージフォーマットは同じであることから，BLE の HCI コマンドは BR/EDR の HCI コマンドと同様にフックすることが可能である．その際，図 6 に示す独自ヘッダを HCI コマンドのコピーに付与する．ヘッダの各フィールドの定義は以下の通りである．

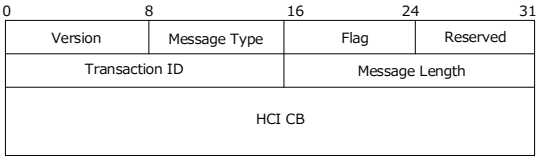

図 6　独自ヘッダのフォーマット
Fig. 6　Original header format.

- Version：独自ヘッダ付 HCI メッセージのバージョン情報．
- Message Type：HCI メッセージの種類．
- Flag：HCI メッセージの正常／異常を示すフラグ．
- Reserved：予約用フィールド．
- Transaction ID：トランザクションを示す識別子．
- Message Length：独自ヘッダ以降のメッセージ長．
- HCI Control Block：HCI メッセージの処理に必要な情報．

独自ヘッダ付きの HCI コマンドは Netlink ソケットを用いて，ユーザ空間にある Forwarder モジュールへと転送される．

Filter モジュールは Forwarder モジュールから `recv_forwarder()` 関数宛に転送される独自ヘッダ付きの HCI イベントを受け取ると，独自ヘッダに記載されている情報をもとに HCI イベントを生成し，BlueZ の `hci_recv_frame()` 関数へ送信する．その後，既存の Bluetooth Application へ HCI イベントが渡される．これらの

処理により，遠隔地から受信した HCI メッセージを，自端末の Bluetooth Application と Bluetooth I/F 間で交換される HCI メッセージと同じものとして処理できる．

なお，Filter モジュールから Forwarder モジュール宛の処理は実装が完了しているが，Filter モジュールの `hci_recv_frame()` 関数における HCI イベントメッセージの生成処理は本稿執筆時点で実装中である．

4.1.2 Forwarder モジュール

Forwarder モジュールは従来手法から変更はないが，実装の詳細を下記に示す．Forwarder モジュールではトンネルテーブルの管理および CD-BGW における Forwarder モジュール間の通信を行う．Filter モジュールから受け取った HCI メッセージをそのままインターネット上で送信することができないため，CD-BGW 間に UDP トンネルを構築し，HCI メッセージをカプセル化して送信する．

事前に CD は BGW と接続しておき，その際に使用している接続先 IP アドレスおよびポート番号を**表 1** に示すトンネルテーブルに記録しておく．トンネルテーブルは UDP トンネルの端点情報だけでなく，転送する HCI メッセージに含まれる OpCode や Handle，RD のデバイスアドレスをトンネルテーブルに記録することで，BGW は CD が複数台あっても宛先の CD を特定することができる．なお，宅内の BGW に通信を開始する必要があるため，ブロードバンドルータにポートフォワーディングの設定などを行う必要がある．

5. 評価

5.1 動作確認

プロトタイプ実装した提案方式の動作検証を行うために，**図 7** に示す環境において実験を行った．CD と BGW は Linux PC（Ubuntu 12.04）で構築し，各 Bluetooth I/F には BUFFALO 社製の Bluetooth/USB ドングル BSBT4D09BK を使用した．CD と BGW は事前にトンネル構築処理を行い，それぞれのトンネルテーブルにトンネル構築先の IP アドレスおよびポート番号が記載されている状態とした．CD のターミナル上で Bluetooth 端末の探索を行うコマンド `hcitool inq` を実行した．その際，Wireshark を用いて CD と BGW 間でやり取りされるパケットをキャプチャした．

動作検証の結果，CD の Host で生成された HCI イベントメッセージ（Inquiry）が UDP データグラム内に格納されていることをパケットキャプチャデータから観測した．また，BGW が発見して取得した RD の MAC アドレスが CD の Filter モジュールまで届いていることを確認した．なお，今回のプロトタイプ実装では CD の Filter モジュールから Host 部へ渡す処理が未完成のため，CD で実行している Bluetooth アプリケーションに RD の MAC アドレスは渡されていないが，提案方式の特徴である CD の Host と BGW の Controller をペアにして遠隔地に存在する RD を発見できることは確認できた．

表 1 トンネルテーブル
Table 1 Tunnel Table.

カラム名	サイズ	説明
OpCode	16bit	HCI コマンドの OpCode の値
RD ADDR	48bit	RD の BD ADDR の値
Handle	12bit	RD のリンクを識別する値
Dest IP address	32bit	トンネル構築先 IP アドレス
Dest Port	16bit	トンネル構築先ポート番号

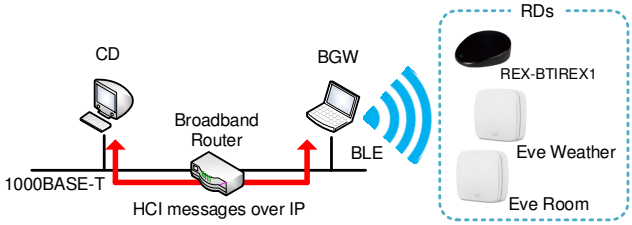

図 7 動作環境
Fig. 7 Verification environment.

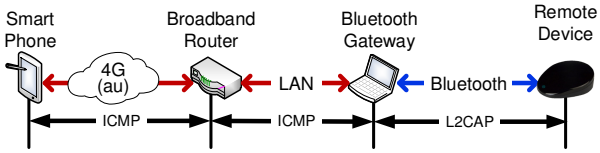

図 8 測定環境
Fig. 8 Measurement environment.

5.2 実環境による検証

5.2.1 RTT 測定

提案手法では Bluetooth 通信における Host と Controller の間にインターネット上で発生する伝送遅延が加わるため，Bluetooth 通信がタイムアウト時間内に完了するかを明らかにする必要がある．そこで，**図 8** に示す測定環境をを用いて RTT を測定することで，実環境における提案手法の実現性を確認する．CD と BGW 間の RTT を測定するために `ping` を用いるが，スマートフォンおよび BGW はともにプライベートネットワーク内に存在するため，直接 ICMP パケットをやり取りすることができない．そこで，CD-BGW 間では 4G 回線を想定し，4G LTE で接続可能な au 社製の Android スマートフォン（AQUOS SERIE mini SHV31）からブロードバンドルータに対して，また，BGW からブロードバンドルータに対してそれぞれ `ping` を実行した．BGW-RD 間は，Bluetooth 通信で Echo Request/Response を送受信して RTT を測定することが可能な `l2ping` コマンドを実行した．測定回数は各環境とも 100 往復である．

表 2 に RTT の測定結果の平均値と最大値を示す．4G

表 2　RTT 測定結果
Table 2　Measurement results of RTT.

	4G	LAN	Bluetooth
平均時間 [msec]	69.60	0.88	35.80
最大時間 [msec]	84.88	1.13	59.54

通信から Bluetooth 通信までの RTT の最大時間は，それぞれ 84.88[msec]，1.13[msec]，59.54[msec] となり，合計すると 145.55[msec] となった．通常の Bluetooth 通信におけるタイムアウト値は 1000[msec] であることから，提案手法は通常の Bluetooth 通信がタイムアウトする前に，CD が RD の情報を受信できることがわかった．

5.3　考察

本稿では BlueZ を用いて提案手法のプロトタイプ実装および検証を行った．筆者らは Android スマートフォンに提案手法を実装し，それを CD として利用することを想定している．Android は Linux カーネルを採用しており，Bluetooth プロトコルスタックとして BlueZ をこれまで採用してきた [13]．そのため，提案手法における Filter モジュールおよび Forwarder モジュールをクロスコンパイルすれば移植することができる．しかし，現在市販されている Android 端末（4.2 JellyBean 以降）には Bluedroid と呼ぶ新しい Bluetooth プロトコルスタックに変更されている．そこで，提案手法が Bluedroid に適用可能か検討した．図 9[*1] に Bluedroid のスタック構成を示す [14]．図中の Bluetooth Stack が Bluedroid になっており，Bluedroid には HCI が含まれている．提案手法は端末内で交換される HCI メッセージに着目した手法であることから，Bluedroid 搭載端末にも提案手法を適用することが可能であると考えられる．

6.　まとめ

本稿では，遠隔地にある Bluetooth 搭載機器をシームレスに接続および通信する手法を説明し，その実装方法について述べた．宅内に設置する BGW と操作端末 CD に HCI メッセージをフックしカプセル化するモジュールと，カプセル化した HCI メッセージをインターネット経由で通信するモジュールを追加実装することで提案手法を実現することができる．提案手法のプロトタイプ実装を行い，動作検証および実ネットワーク環境での通信遅延を評価した結果，Bluetooth で規定されているタイムアウト時間内に CD が遠隔地の RD を発見できることを確認した．また，現在市販されている Android スマートフォンに搭載されている Bluetooth プロトコルスタック Bluedroid について調査した結果，提案手法を適用できる見込みが明らかとなった．

[*1] 文献 [14] より引用

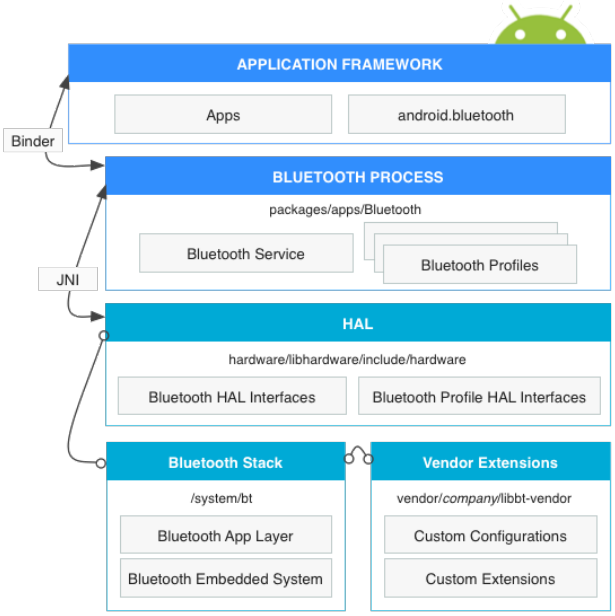

図 9　Bluedroid のスタック
Fig. 9　The stack of Bluedroid.

今後は実装を完成させ，Bluedroid への移植を行うことにより，実環境で RD の遠隔制御ができることを確認する予定である．

参考文献

[1] 一色正男，河口俊朗，平原茂利夫：広がる東芝ネットワーク家電 “フェミニティ” シリーズ，東芝レビュー，Vol. 60, No. 4, pp. 23–27 (2005). https://www.toshiba.co.jp/tech/review/2005/04/60_04pdf/a07.pdf.
[2] Sumino, H., Ishikawa, N., Murakami, S., Kato, T. and Hjelm, J.: PUCC architecture, Protocol and Applications, *Proc. of 4th IEEE Consumer Communications and Networking Conference (CCNC)*, pp. 788–792 (2007).
[3] 伊藤崇洋，加藤悠一郎，峰野博史，石川憲洋，水野忠則：異種デバイス連携基盤を用いたセンサ・家電制御アプリケーション，情報処理学会研究報告コンピュータセキュリティ，Vol. 2011-CSEC-52, No. 35, pp. 1–6 (2011).
[4] 田中　剛，伊藤崇洋，加藤悠一郎，峰野博史，水野忠則：Android 端末を用いた異種ネットワークデバイス連携システムの開発，マルチメディア，分散，協調とモバイル（DICOMO2011）シンポジウム論文集，Vol. 2011, pp. 1257–1264 (2011).
[5] d’Assise Bissyandéand, T. F., Réveillère, L. and Bromberg, Y.-D.: UbiGate: a gateway to transform discovery information into presence information, *Proc. of the 4th International Workshop on Services Integration in Pervasive Environments (SIPE 09)*, No. 6, pp. 19–24 (2009).
[6] 井波政朗，丹　康雄：Bluetooth ネットワークの有線拡張方式に関する検討，電子情報通信学会技術研究報告.CS, Vol. 103, No. 415, pp. 47–52 (2003).
[7] Albert, J., Bissyandé, T. F., Bromberg, Y.-D., Chaumette, S. and Réveillère, L.: UbiPAN: A Bluetooth Extended Personal Area Network, *Proc. of International Conference on Complex, Intelligent and Software Intensive Systems (CISIS) 2010*, pp. 774–778 (2010).

[8] Tsuda, K., Suzuki, H., Asahi, K. and Watanabe, A.: Proposal for a Seamless Connection Method for Remotely Located Bluetooth Device, *Proc. of 7th International Conference on Mobile Computing and Ubiquitous Networking (ICMU)*, pp. 78–79 (2014).
[9] 岡田真実，鈴木秀和：遠隔地にある Bluetooth LE 機器のシームレス接続システムの実装，第 78 回情報所学会全国大会講演論文集，Vol. 2016, No. 3, pp. 489–490 (2016).
[10] Bluetooth SIG: Bluetooth core specification, available from ⟨https://www.bluetooth.com/specifications/bluetooth-core-specification⟩ (accessed 2016-07-26).
[11] Bluetooth SIG: Host Controller Interface (HCI) Architecture, available from ⟨https://developer.bluetooth.org/TechnologyOverview/Pages/HCI.aspx⟩ (accessed 2016-07-26).
[12] Internet WG: Internet Protocol Support Profile Bluetooth Specification, Bluetooth SIG (2014).
[13] BlueZ Project: BlueZ, available from ⟨http://www.bluez.org/⟩ (accessed 2016-07-26).
[14] Android Open Source Project: Bluetooth, available from ⟨https://source.android.com/devices/bluetooth.html⟩ (accessed 2016-07-29).