

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

岡田 真実 鈴木 秀和
名城大学理工学部

1 はじめに

近距離無線通信技術の発達により，宅内に存在する Bluetooth 対応機器を外出先から遠隔操作したいという要求が高まっている．しかし，Bluetooth には通信可能範囲が制限されているため，ユーザは制限された範囲内でしか Bluetooth 対応機器を操作することができない．文献 [1] では，遠隔地から Bluetooth 対応機器を操作するシステム（以下，従来手法）を提案している．従来手法では，ユーザが宅内に設置されたゲートウェイ機器を経由し，Bluetooth 通信を行うことで，外出先から間接的に遠隔操作を行うことが可能となる．しかし，近年普及が進んでいる Bluetooth 4.0 LowEnergy (BLE) 機器に対応していないという課題が存在する．

本稿では，従来手法を拡張することにより，遠隔地に存在する BLE 機器とシームレスに接続する手法についての提案と実装を行う．

2 Bluetooth

Bluetooth はソフトウェアがハードウェア機能にアクセスするために共通のインタフェース方式として，HCI (Host Controller Interface) を用いている．図 1 に HCI 層のスタック層を示す．Bluetooth では，ソフトウェアで構成される上位層を Host，ハードウェアで構成される下位層を Controller と呼称する．Bluetooth 端末は，Host と Controller の間で HCI コマンド，HCI イベント，HCI データという制御メッセージを交換することで通信を可能にしている．

しかし，Bluetooth はバージョンによって互換性が異なるという特徴を持っており，近年普及が予想される BLE は従来普及された Bluetooth 3.0 とは互換性を持たない．

3 提案手法

図 2 に提案手法の概要を示す．提案手法の前提として宅内に Bluetooth インタフェースと IP インタフェースを搭載した BGW (Bluetooth Gateway) を設置する．ユーザが操作する Bluetooth 搭載端末 CD (Control Device) 間で交換される HCI メッセージをフックし，インターネット経由で宅内の BGW へ送信する．その際，HCI メッセージはカプセル化される．操作端末からの HCI メッセージを受け取りデカプセル化した BGW は，HCI メッセージの中身から宅内にある Bluetooth 端末 RD (Remote Device) と接続およびデータのやり取りを行う．

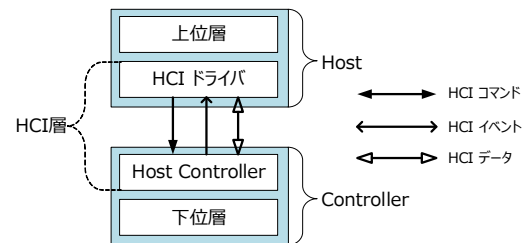


図1 Bluetoothのスタック

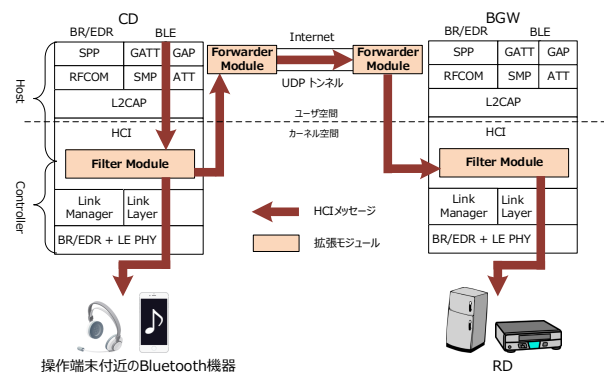


図2 提案手法の概要

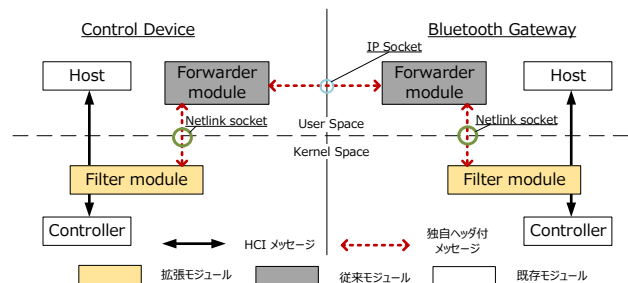


図3 モジュール構成

BLE と従来の Bluetooth 規格である BR/EDR は，Host 部におけるプロファイルやプロトコルは異なるが HCI 部分は共通であり，BLE も BR/EDR 同様，端末内の Host と Controller の間で HCI メッセージをやりとりすることで接続や通信を確立している．そこで，従来手法で実装されている Filter モジュールを拡張し，BLE 仕様の HCI メッセージを新たにフックすることで提案手法を実現する．提案手法の具体的な処理内容については，4.1 節の実装モジュールの説明にて詳述する．

4 実装・評価

4.1 実装

実装設計のシステム構成について図 3 に示す．提案手法を実装するにあたり，BlueZ[2] を用いて実装を行う．

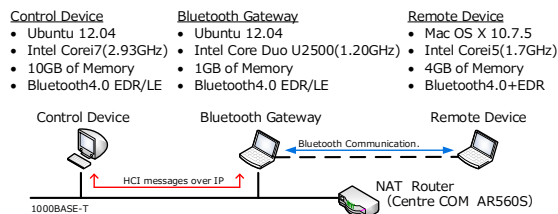


図4 動作環境

BlueZ は Android および Linux の Bluetooth プロトコルスタックとして採用されているスタックである。カーネル空間の BlueZ に Filter モジュールを、またユーザ空間に Forwarder モジュールを実装する。Filter モジュールと Forwarder モジュールの間は、Netlink ソケットを用いてデータの受け渡しを行う。

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 モジュールには以下の機能を持たせる。プロトタイプとして (1)-(3) の機能を実装した。

- (1) Host と Controller 間で交換される HCI メッセージのコピーをフックする。この際、従来の規格の HCI メッセージだけでなく、BLE の HCI メッセージもコピーし、フックする。
- (2) フックした HCI メッセージを独自ヘッダ付でカプセル化/デカプセル化する。
- (3) HCI Forwarder モジュールと独自ヘッダでカプセル化した HCI メッセージを、Netlink ソケットを用いて送信する。
- (4) 転送されてきた HCI メッセージの独自ヘッダに記載されている情報をもとに、ソケットバッファを構築し、既存の Bluetooth モジュールに渡す。

4.1.2 Forwarder モジュール

Forwarder モジュールは従来手法の構成のまま実装することが可能なモジュールである。プロトタイプには (3), (4) の機能を実装した。

- (1) トンネル構築処理およびトンネルテーブルの管理を行う。
- (2) HCI メッセージをチェックし、メッセージの転送先を確認する。
- (3) UDP ヘッダのカプセル化/デカプセル化を行う。
- (4) トンネルテーブルに記録された情報をもとに UDP パケットの送受信を行う。

4.2 動作確認

一部機能を実装したプロトタイプの動作確認を行った。図4に検証環境を示す。CD と BGW は Linux PC で構築し、Bluetooth インタフェースには Bluetooth/USB ドングルを用いた。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を記載した。CD のターミナル上で、Bluetooth 端末の探索を行うコマンドを実行し、CD の Host から BGW の Controller へ端末探索コマンドの送信を行った。その際、CD と BGW 間で

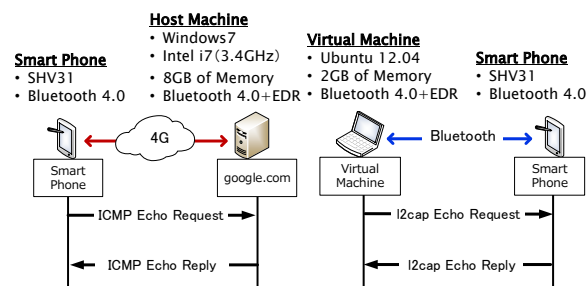


図5 試験構成

表1 RTT 測定

	4G 回線	Bluetooth
平均時間 (msec)	90.00	24.06
最大時間 (msec)	382.02	48.19

Wireshark を用いて UDP でカプセル化された HCI メッセージのキャプチャを行った。結果、BGW で取得した RD の MAC アドレスが CD の Filter モジュールまで届いたことを確認した。

4.3 性能評価

提案手法では Bluetooth の 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、CD-BGW 間および BGW-RD 間における RTT を測定した。図5に試験構成を示す。CD-BGW 間は 4G 回線を想定し、スマートフォンから Google のサーバに対して ping を実行した。BGW-RD 間では、Bluetooth の L2CAP 層間で Echo Request/Response を送受信する l2ping コマンドを使用して RTT を測定した。測定回数は 4G, Bluetooth とも 100 往復である。

表1に RTT の各測定結果を示す。4G および Bluetooth の RTT の最大時間は、それぞれ 382.02[msec], 48.19[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、提案手法では通常の Bluetooth 通信がタイムアウトする前に、CD は HCI メッセージを BGW へ転送し、BGW が代理で探索して発見した RD の情報を受信できることがわかる。したがって、提案手法の実装が完了すれば、CD は BGW を中継して RD との間で Bluetooth 通信を実現することができる。

5 まとめ

提案手法では、従来手法を拡張することで、遠隔地にある BLE 機器を操作することが可能となる。今後は、Linux カーネルモジュールの実装を完成させる予定である。

参考文献

- [1] Kazuma Tsuda, et al.: Proposal for a Seamless Connection Method for Remotely Located Bluetooth Device, ICMU2014, pp. 78-79 (2014).
- [2] BlueZ : <http://www.bluez.org/>.