

遠隔地にある Bluetooth 機器間のシームレス接続手法の実装と検証

津田 一磨[†] 鈴木 秀和[†] 旭 健作[†] 渡邊 晃[†][†]名城大学大学院理工学研究科

1 はじめに

Bluetooth に代表される近距離無線技術の発達により、ホームオートメーションの普及が期待されている。今後、宅内にある近距離無線通信機器を外出先から操作したいという要求が高まると考えられる。しかし、このような機器には通信可能範囲に制限があり、外出先から直接操作することができない。

我々は、Bluetooth 機器のハードウェアとソフトウェアの間で交換されるコマンド等をインターネット経由で転送することにより、遠隔地の Bluetooth 機器へ接続する手法を提案している [1]。本稿では、提案手法を Linux PC へ実装し、動作検証を行った結果を報告する。外出先の操作端末が遠隔地にある Bluetooth 機器を検出可能であることを確認した。

2 Bluetooth

図 1 に Bluetooth スタックの構成を示す。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 と呼ぶ [2]。Host と Controller は HCI (Host Controller Interface) コマンド、HCI イベント、HCI データを交換することにより Bluetooth 通信を行う。Bluetooth では、Host が Controller へ送信する HCI コマンドに応じて、Bluetooth 通信を行う。また、他の端末からの Bluetooth 通信に応じて、Controller が Host へ HCI イベントを送信し、通信の内容を通知する。このように、Host と Controller の間で制御メッセージを交換することにより、端末間で接続を確立し、HCI データに含まれるデータパケットを転送する。

3 提案手法

3.1 概要

提案手法は、ユーザが Bluetooth 機器の位置を意識することなく、一般の Bluetooth アプリケーションを用いてシームレスに接続できることを目的としている。

図 2 にシステム構成を示す。本稿では、宅内に存在

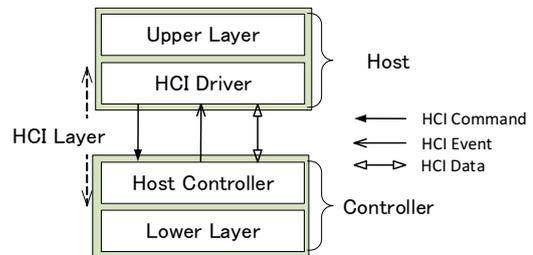


図1 Bluetooth スタックの構成

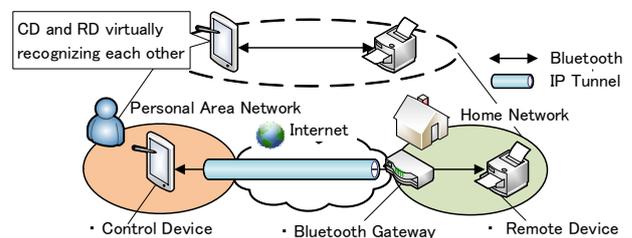


図2 システム構成

する接続対象の Bluetooth 機器を RD (Remote Device)、外出先の操作端末を CD (Control Device) と呼ぶ。CD は直接 RD と Bluetooth 通信できないため、通信を中継するためのゲートウェイ BGW (Bluetooth Gateway) を宅内のネットワークに設置する。BGW は Bluetooth インタフェースと Ethernet または Wi-Fi などの IP インタフェースを 1 つずつ実装する。提案手法では、CD の Host と Controller の間で交換される HCI メッセージ¹をフックし、インターネット経由で BGW に転送する。BGW は受信した HCI メッセージに応じて宅内の RD と接続する。このように、提案手法では CD と BGW を合わせて 1 つの Bluetooth 機器であるかのように HCI メッセージを交換できる。そのため、CD は遠隔地の RD が近傍に存在しているかのように認識することができる。さらに、ユーザは一般の Bluetooth アプリケーションを利用できる。

4 実装と検証

4.1 実装

本章では、提案方式を Linux へ実装する方法について述べる。図 3 に CD のモジュール構成および BGW のモジュール構成を示す。Linux に実装されている Blue-

Implementation and Verification of Seamless Connection Method for Bluetooth Devices in Remote Locations

Kazuma Tsuda[†], Hidekazu Suzuki[†], Kensaku Asahi[†] and Akira Watanabe[†][†] Graduate School of Science and Technology, Meijo University¹ 本稿では HCI コマンド、HCI イベント、HCI データをこのように総称する。

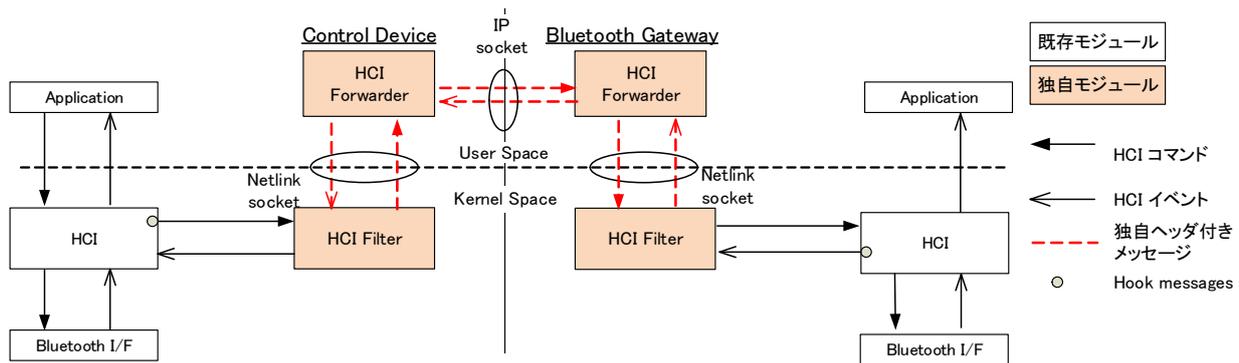


図3 モジュール構成 (左: CD側, 右: BGW側)

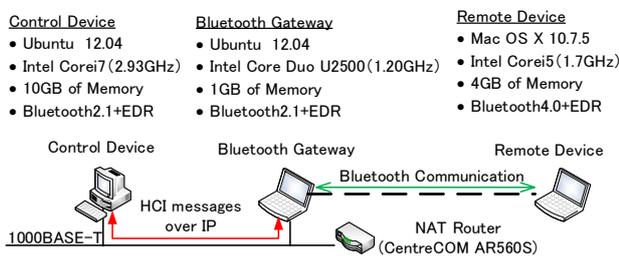


図4 動作検証環境

Bluetooth Driver Stack “BlueZ”[3] に HCI Filter モジュールを、またユーザ空間に HCI Forwarder モジュールをそれぞれ実装する。

HCI Filter モジュールは HCI メッセージをフックする機能を持つ。また、HCI Forwarder モジュールから渡された HCI メッセージを既存の HCI モジュールへ戻す役割を持つ。HCI メッセージはそのままでは、インターネット上を通過できない。そのため、HCI Forwarder モジュールが HCI Filter にフックされた HCI メッセージを IP でカプセル化することにより、転送を可能にする。また、転送されてきた IP パケットをデカプセル化し、中の HCI メッセージを取り出し、HCI Filter モジュールへ渡す機能を持つ。

上記の改造を加えた Bluetooth Driver Stack を通常の Linux カーネルへ組み込むことにより、両モジュールが連携して動作する。なお、Android に搭載されている Bluetooth Stack は Linux と同じスタックが採用されているため、Android スマートフォンへの実装も可能である [4]。

4.2 動作検証

HCI メッセージを CD と BGW 間で交換する機能まで実装が完了しているため、本稿ではその動作検証を行った。図4に動作検証環境を示す。CD と BGW は Linux PC で構築しており、Bluetooth インタフェースには、Bluetooth/USB ドングルを用いた。CD と BGW は事前にトンネルを構築し、それぞれ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 のターミナルから BlueZ が提供する hcitool を用いて、端末探索を行う HCI Inquiry コマンドを実行し、CD の Host から BGW の Controller へコマンドを送信した。CD と BGW で Wireshark を用いて、UDP でカプセル化された HCI メッセージをキャプチャしたところ、正しく HCI Inquiry コマンドを転送できていることを確認した。また、CD のカーネルで、BGW から RD のアドレスを通知する HCI Inquiry Result イベントを受信し、CD が RD のアドレスを取得できたことを確認した。

5 まとめ

提案手法では、ユーザ自身の位置や接続したい Bluetooth 機器の場所を意識することなく、常に一般のアプリケーションで Bluetooth 機器へ接続できる。提案手法の主要な機能を実装して動作検証を行った結果、外出先の操作端末が宅内の一般の Bluetooth 端末のアドレスを取得できることを確認した。今後は、Linux カーネルモジュールの実装を完成させ、提案手法による Bluetooth 通信のスループットおよび遅延に関する評価を行う予定である。

参考文献

- [1] 津田一磨, 鈴木秀和, 旭 健作, 渡邊 晃: 遠隔地にある Bluetooth 機器間のシームレス接続手法の実装, DICO2013 シンポジウム論文集, Vol. 2013, No. 1, pp. 805–811 (2013).
- [2] Bluetooth SIG: *Adopted Bluetooth Core Specifications Ver4.0*, <http://www.bluetooth.org/Technical/Specifications/adopted.htm>.
- [3] BlueZ Project: *BlueZ*, <http://www.bluez.org/>.
- [4] Android Open Source Project: *Bluetooth*, <http://www.kandroid.org/online-pdk/guide/bluetooth.html>.