

タブレット型端末による無線センサネットワークの管理に関する研究 (第2報)

油井誠志・宮本博永・布施嘉裕・中込広幸・古屋雅章

Study on Wireless Sensor Network Management by Tablet Terminal (2nd Report)

Seishi YUI, Hironaga MIYAMOTO, Yoshihiro FUSE, Hiroyuki NAKAGOMI and Masaaki FURUYA

要 約

IEEE802.15.4 に準拠した通信モジュールと ZigBee 通信プロトコルを使用して、メッシュ型の無線ネットワークを構築し、タブレット型端末上でネットワークの状態を管理するソフトウェアを開発する。昨年度開発したメッシュ型ネットワークの形状を把握するアプリケーションを拡張し、任意の通信モジュールにコマンドを送信してレスポンスを受信できる機能を追加した。また、ユーザーインターフェイスとして Web アプリケーションを開発した。

1. 緒 言

近年、東日本大震災の影響もあって省エネルギー技術が脚光を浴びている。その1つであるエネルギー管理システムは、センサネットワークから入手した情報から無駄なエネルギー消費を抑えようという試みであり、そのハードウェア基盤となるセンサネットワークも重要性を増している。

例えば、工場等でエネルギー管理システムを導入する目的でネットワークを新たに構築する場合、エネルギー管理のためのセンサと有線ケーブルの設置工事が必要となる場合が多い。センサを壁面に埋め込み、ケーブルを壁面内で配線しようとする、一時的に操業を停止しなければならない場合もあり、多大な費用と時間が発生してしまう。安価に抑えようとするセンサやケーブルが露出して見た目も悪く、ケーブルが運搬等の障害になることも少なくなかった。これらの理由から、今後は無線でセンサネットワークを構築する場合が多くなると考えられる。

無線センサネットワークの構築に適しているとされる無線通信規格が IEEE802.15.4 である。低消費電力と低コストを実現しており、メッシュ型ネットワークを構築できることが特徴となっている。しかし、メッシュ型ネットワークは管理が難しく、管理用のツールも多くはない。

そこで、工場等のエネルギーマネジメントを実施する場合に活用できるタブレット型端末から、ネットワークが管理できる無線ネットワークシステムの構築を目的に開発を行った。今年度は昨年度開発したメッシュ型ネッ

トワークの形状を把握するアプリケーションを拡張し、任意の通信モジュールにコマンドを送信してレスポンスを受信できる機能を追加した。また、ユーザーインターフェイスとして Web アプリケーションを開発した。

2. 仕 様

2-1 ハードウェア構成

図1のようなハードウェア構成で動作するものとする。IEEE802.15.4 をベースとしてメッシュ型ネットワークを構築し、通信プロトコルに ZigBee を使用する。任意の通信モジュールにセンサを取り付けることで、センサネットワークとして機能するようになる。Coordinator モジュールをシングルボードコンピュータに USB 接続し、シングルボードコンピュータを LAN に接続することで、LAN 上のタブレット型端末からセンサネットワークの情報を参照することが可能になる。

2-2 ソフトウェア構成

ソフトウェア構成図は図2のようになる。タブレット型端末にネットワーク情報を出力するための Web アプリケーション、および、通信モジュールから収集した情報を分析するバックグラウンドアプリケーションを開発対象とする。

2-3 使用機器

LAN とセンサネットワークを中継するシングルボードコンピュータに、ラズベリーパイ財団が開発している Raspberry Pi Model B を使用している。OS として同財団が配布している Linux ディストリビューション Raspbian をインストールした。Web アプリケーションサーバと

して Tomcat 7.0 を、データベースサーバとして MySQL 5.5 をパッケージインストールしている。Java の最新バージョンは 8 であるが、Tomcat パッケージが要求しているバージョンが 7 のため、開発言語として OpenJDK 7 をパッケージインストールした。開発ライブラリとして librx-java パッケージを追加インストールし、GitHub から xbee-api をダウンロードし展開している。

通信モジュールとしてディジインターナショナル社の XBee-PRO ZB S2B モジュールを使用している。ZigBee の仕様に合わせて 1 台を Coordinator としてファームウェアを設定し、残りは Router または End Device としてファームウェアを設定する。通信モードは API モードで更新する。

タブレット型端末には Google 社の Nexus7 (2013) を使用しており、OS のバージョンは Android 6.0.1 となっている。

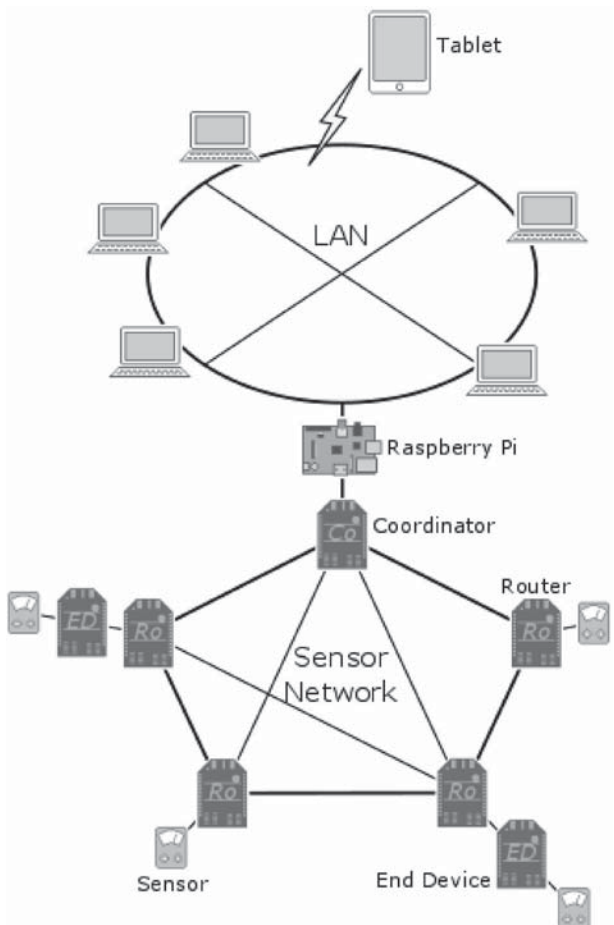


図 1 ハードウェア構成図

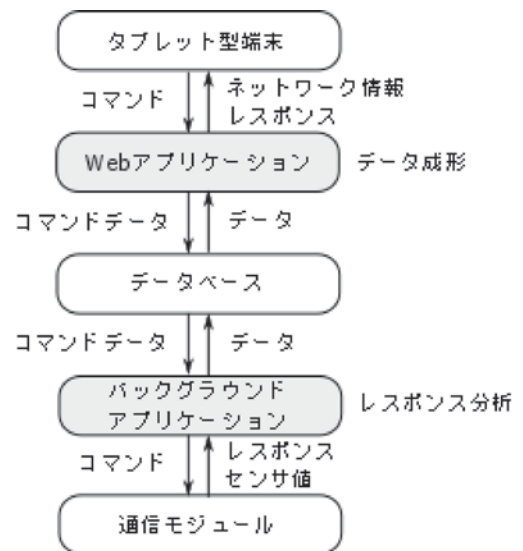


図 2 ソフトウェア構成図

3. 結果

3-1 ネットワーク形状の把握

バックグラウンドアプリケーションを起動すると、ネットワークの形状を把握するためのスキャンを始める。十分な時間（通信モジュール数×10 秒程度）が経過してからタブレット端末で Web ブラウザを起動し、Web アプリケーションを起動すると、図 3 のような画面が出力される。

図 3 はネットワーク形状をスキャンした結果を、表形式で出力したものである。この状態ではわかりづらいが、表を解析し図示すると図 4 のように表すことができる。これらより、通信モジュールに対するコマンド送受信を行うための前段階であるネットワーク形状の把握は良好に機能していることを確認した。

3-2 通信モジュールの把握

図 3 で、任意の通信モジュールの 64 ビットアドレスをクリックすることで、各モジュールの状態が出力される（図 5）。

モジュールの役割や 16 ビットアドレス、接続モジュールの下には、コマンドとパラメータの入力用テキストボックスを設置した。テキストボックスにコマンドやパラメータ（16 進数）を入力して send ボタンをタップすることで、コマンドをモジュールに送信することができる。

送信したコマンドとレスポンスの一覧を、時系列で最下部に出力している。センサ値などを定期的に受信していた場合、出力が非常に長くなってしまったため最新の 20 件を出力するにとどめている。コマンドとレスポンスの違いは、先頭列で区別が可能なようにした。送信したコマンドは「Send」、受信したレスポンスは「Recv」と出力している。最後列をパラメータ値とレ

スポンズ値で共有しているため、先頭列でそのどちらかを判断しなくてはならない。また、確認コマンドの場合、パラメータを伴わないことが多く、パラメータがない場合は null を出力している。以上により、目的とする任意の通信モジュールに対するコマンド送受信機能が正常に動作することを確認した。

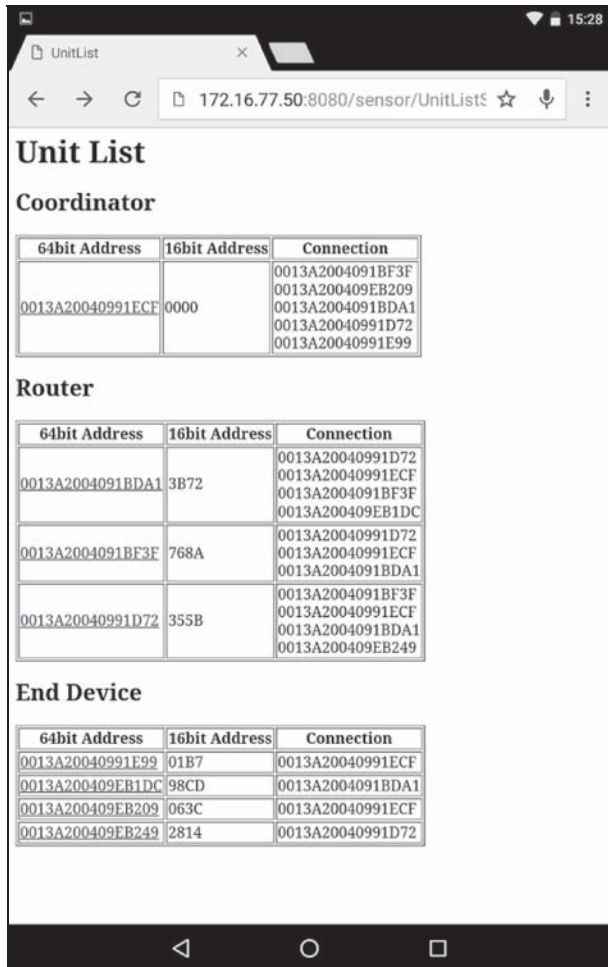


図3 ネットワーク形状の出力

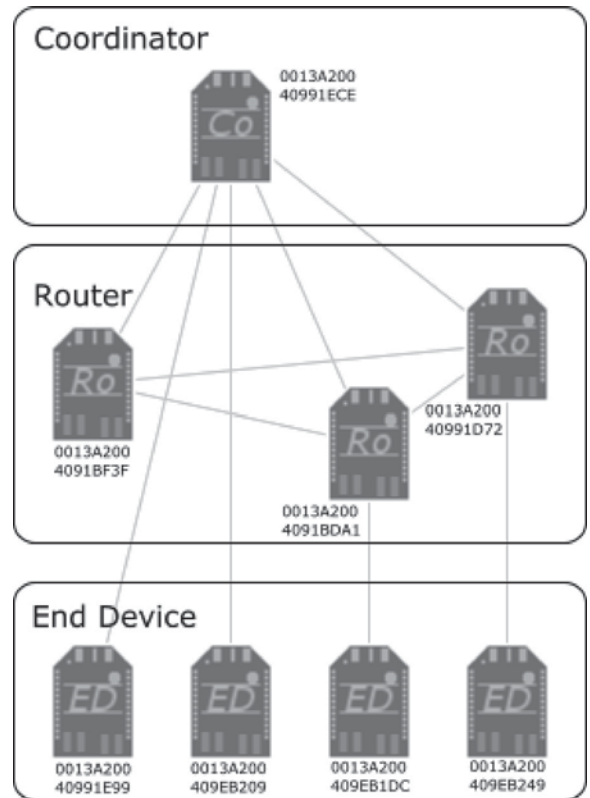


図4 ネットワーク解析図

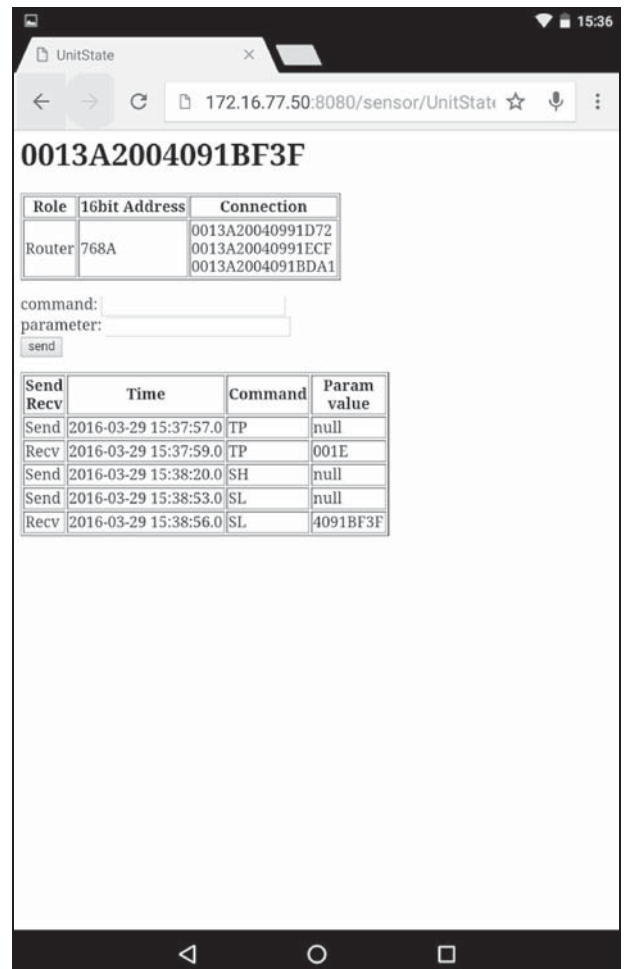


図5 モジュール状態

3-3 パケットの消失

ZigBee では通信時に受信確認を行わないため、パケットの消失がしばしば発生する。たとえば図5において、15時38分に送信したコマンド SH のレスポンスが返ってきていない。これは、送信したコマンドがこのモジュールに届かなかったか、レスポンスが Coordinator に届かなかった可能性が高い。

これらの現象は UDP による通信と非常に似ていることから、本送受信機能の将来的な用途も UDP に準ずるものと思われる。すなわち部分的に欠損が生じても問題がないデータの転送への利用に向いている。確実にデータを送受信する必要がある場合、ソフトウェア的に受信確認、再送信の機能を別途構築しなくてはならない。

3-4 非同期通信について

ネットワーク形状を把握するために、バックグラウンドアプリケーションは、定期的に全てのモジュールに結合状態を確認するコマンドを送信している。これまでは、ネットワーク形状の把握が目的であったため、コマンドの送信とレスポンスの受信をペアで行う同期通信を用いてアプリケーションを構築していた。

今回、機能を強化するにあたり、送信と受信をペアで行わない非同期通信方式にプログラムを書き換えている。バックグラウンドアプリケーションは、受信状態を常時確認し、何らかのデータを受信したときは、データを解析してデータベースに格納している。送信は一定間隔で行っており、送信タイミングでユーザーコマンドの有無を確認し、ユーザーコマンドが指定されていればコマンドを送信、無ければ結合状態確認コマンドを送信している。前述したように通信パケットが消失する、転送が遅れるなどの理由により、コマンドとレスポンスが必ずしも正しく対応できるとは限らない。その例を図6に示す。これは図3と同じネットワークに対して、バックグラウンドアプリケーションを起動した直後の状態をタブレット型端末で参照したものである。バックグラウンドアプリケーション起動直後は、多くの通信モジュールに対して結合状態確認コマンドを送信するため、コマンドとレスポンスのミスマッチが発生しやすい。

4. 結 言

IEEE802.15.4 に準拠した通信モジュールによって構成されるセンサネットワークの情報を収集するバックグラウンドアプリケーションと、収集された情報を参照するための Web アプリケーションを開発した。また、任意のモジュールに対して、コマンドを送信する機能も実装した。これにより、ネットワーク形状の把握を行うことができ、それからある程度の位置情報がわかるため、例

えば、動物の生息⁵⁾や土砂災害状況⁶⁾等の観測への活用が期待される。

64bit Address	16bit Address	Connection
0013A20040991ECF	0000	0013A2004091BF3F 0013A200409EB209 0013A2004091BDA1 0013A2004091D72 0013A20040991E99

64bit Address	16bit Address	Connection
0013A2004091BDA1	3B72	0013A20040991D72 0013A20040991ECF 0013A2004091BF3F 0013A200409EB1DC
0013A2004091BF3F	768A	0013A20040991D72 0013A20040991ECF 0013A2004091BDA1
0013A20040991D72	355B	0013A2004091BF3F 0013A20040991ECF 0013A2004091BDA1 0013A200409EB249 0013A200409EB209 0013A20040991E99

64bit Address	16bit Address	Connection
0013A20040991E99	01B7	0013A20040991ECF
0013A200409EB1DC	98CD	0013A2004091BDA1 0013A20040991D72
0013A200409EB209	063C	0013A20040991ECF
0013A200409EB249	2814	0013A20040991D72

図6 ミスマッチの例

参考文献

- 1) xbee-api: <<https://code.google.com/p/xbee-api/>> (2014/10/16 - 2015/02/13)
- 2) Linux 工作室: <<http://penguin.tantin.jp/hard/XBee.html>> (2014/10/16 - 2015/02/13)
- 3) SONY AROUJE: Connecting XBee to Raspberry Pi <<http://java.dzone.com/articles/connecting-xbee-raspberry-pi>> (2014/10/27 - 2014/11/13)
- 4) XBee on the Raspberry Pi: <<http://rapplogic.blogspot.jp/2013/06/xbee-on-raspberry-pi.html>> (2014/10/27 - 2015/01/30)
- 5) 伊藤昌毅, et al.: 動物を利用した環境調査のための遭遇検知による効率的なデータ転送手法の提案, DICOMO 2014 論文集, 605-612 (2014) .
- 6) 小泉圭吾, et al.: 土砂災害監視のための無線センサネットワークの実用化に向けた実験的研究, 土木学会論文集 C (地圏工学) 69.1, 46-57 (2013)