

無線センサネットワークのためのプロトコル実装プラットフォーム

川村 健太[†] 桧垣 博章[†]

東京電機大学大学院 未来科学研究科 ロボット・メカトロニクス学専攻[†]

1 はじめに

無線ネットワーク技術の高度化により、あらゆるモノが無線ネットワークによって相互接続するIoT(Internet of Things)が構想され、そのパイロットアプリケーションとして無線センサネットワークが注目されている。多数の無線センサノードによって取得されたセンサデータを無線基地局を通じて収集するために、無線マルチホップ通信の適用が考えられる。ここでは、隣接無線センサノード間の無線アドホック通信を基礎として、各センサノードと無線基地局との無線マルチホップ通信を実現するための諸プロトコルの実装が必要となる。すなわち、センサデータを配送するためのルーティングプロトコルが必要となるが、配送遅延の短縮、無線信号の衝突によるセンサデータの紛失の削減、回避、外部からの継続的な電力供給のない無線センサノードの電力消費への削減や均等化、等を考慮することが求められることから、多様なプロトコルの開発と実装が求められる。この開発、実装を容易に実現するためには、隣接無線ノード間の無線アドホック通信を基盤機能として開発プロトコルが容易に使用可能となるようなハードウェア/ソフトウェアプラットフォームが必要である。

2 関連研究

無線センサネットワークで用いられる無線マルチホップ通信は、各無線ノードが送信元無線ノード、送信先無線ノードとして機能するばかりでなく、中継無線ノードとしても機能することが求められる。また、有線ネットワークでは、同一LANに接続されたノード間でのみ直接通信され、ブロードキャストドメインはノード集合の分割として定められているが、無線ネットワークでは各無線ノードの無線信号到達範囲に含まれる隣接無線ノードと直接通信されることから、ブロードキャストドメインもノードごとに相対的に定まるものとなっている。このような無線センサネットワークの特性から、ルーティング等のネットワーク層のプロトコル開発、実装においても、その特性を考慮したプラットフォームの構築が求められている。[3]では、Raspberry Piを用いた無線マルチホップ通信プロトコルのテストベッド構築法が提案されている。また、[5]では、GNU Radioを用いたプロトコル性能の実機評価環境が提案されている。ただし、これらの環境上でプロトコル開発、実装を容易に行なうための手法が検討されていない。

3 提案手法

本論文では、無線センサネットワークにおけるネットワーク層プロトコルの開発、実装環境の実現を目指す。

一般に、現在のオペレーティングシステムでは、ネットワーク層のプロトコルはオペレーティングシステムのカーネル機能として実装される。例えば、TCP/IPプロトコルスタックは、Linux等のUNIX系オペレーティングシステムでもWindowsオペレーティングシステムでもオペレーティングシステムカーネルの機能として提供されている。実装性能の高さを指標とする場合には、このようなオペレーティングシステムカーネルにおける実装が最適であり、無線センサネットワークのためのプロトコルにおいても同様に考えることができる。しかし、プロトコル開発環境を想定すると、通信性能の高さよりも簡易に開発、実装することが可能となることが望ましい。そこで、開発プロトコルはオペレーティングシステムカーネル外のアプリケーションとして実現することとする。

このとき、本論文で提案するプラットフォームは、オペレーティングシステムと開発プロトコルを相互に接続する中間層の役割を担う。この位置でのプラットフォームの実現方法には、(1)ソフトウェアライブラリとして実現する、(2)オペレーティングシステムサーバとして実現する、という2つが主に考えられる。前者のソフトウェアライブラリとして実現する方法は、プラットフォーム機能をライブラリ関数として実現し、開発プロトコルはそのAPI(Application Program Interface)を介してライブラリ関数の機能を利用する。開発プロトコルのプログラムとプラットフォームが提供するライブラリ関数をコンパイラ(正確にはリンカ)の機能によって一体化し、アプリケーションプロセスとして開発プロトコルを機能させる。一方、後者のオペレーティングシステムサーバとして実現する方法では、マイクロカーネルオペレーティングシステムにおける機能サーバのひとつとして無線センサネットワークのための無線アドホック通信機能を提供するオペレーティングシステムサーバを用意する。開発プロトコルのプログラムは単体のプロセスとして実装し、サーバ機能をプロセス間通信による機能呼び出しとして実装する。いずれの方法でも、オペレーティングシステムカーネルの改造を必要としない点で優れているが、本論文ではより簡易に実現可能なソフトウェアライブラリとして実現することとする。

有線ネットワークにおける通信プロトコルは、同一のLANに接続されたネットワークインタフェース間のユニキャスト通信とマルチキャスト通信(ブロードキャスト通信を含む)を基礎として設計される。そこで、各メッセージは送信先ノードあるいは送信先ノード集合を送信先として送信され、送信先として指定されたノードがこのメッセージを受信する。一方、無線センサネットワークを含む無線マルチホップネットワークにおける通信プロトコルでは、ユニキャスト通信とマルチキャスト通信に加えて、隣接無線ノードから送信されたメッセージの傍受(overhear)が重要な役割を果たす。無線ネットワークにおいて、ある無線ノードが送信した無線信号はその無線信号到達範囲に含まれる

Design and Implementation of Platform for Wireless Sensor Network Protocols

[†]Kenta Kawamura and [†]Hiroaki Higaki

[†]Department of Robotics and Mechatronics, Tokyo Denki University

すべての隣接無線ノードに到達することから、ある無線ノードがユニキャスト送信したメッセージはそのすべての隣接無線ノードが受信することができる。これを利用した様々なプロトコルが開発されており、今後開発されることが見込まれる。[2]では、迂回無線マルチホップ配送経路の検出に制御メッセージの傍受を活用しており、[1]ではブラックホール無線ノードの検出に制御メッセージの傍受を用いている。また、[4]では故障無線ノードの検出にデータメッセージの傍受を用いている。したがって、無線センサネットワークのプロトコル開発プラットフォームでは、メッセージの傍受をサポートすることが必要である。

メッセージの傍受を実現する手法として、本論文では raw socket を用いてオペレーティングシステムカーネルのプロトコル処理を経由せずにプラットフォーム機能を提供するライブラリが直接メッセージを受信することとする(図1)。これによって、オペレーティングシステムカーネルの処理によって傍受する無線ノードを送信先としないメッセージが破棄されることを防ぐことができる。ここで、raw socket によってプロトコルに渡されるメッセージはその無線ノードが受信したすべてのメッセージが対象となり、開発プロトコルは膨大なメッセージの中から必要なメッセージを判別する必要がある。そこで、本論文の提案するプラットフォームは raw socket によって受信したメッセージをフィルタリングする機能を搭載する。

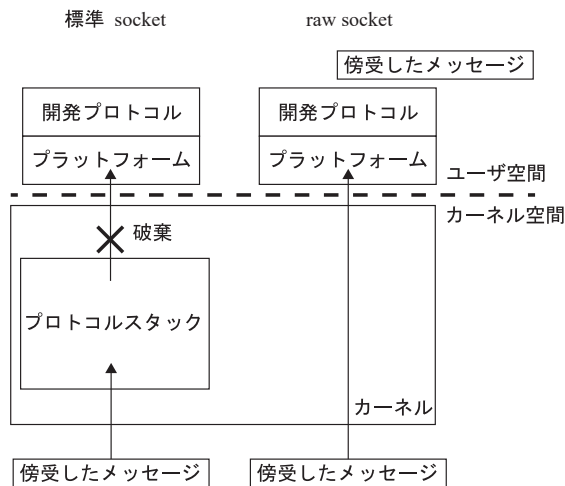


図1: raw socket によるプラットフォームの実装。

一方、開発されるプロトコルでは、メッセージの送受信が基本プリミティブとなることから、メッセージの送信関数 send() と受信関数 receive() が開発プラットフォームによって提供されることが必要である。ここで、各無線ノードの動作は状態遷移機械(state machine)で記述されることが一般的である。このとき、状態の遷移は(1)メッセージの受信、(2)タイムアウト、が主に用いられる。このうち(1)のメッセージの受信による状態遷移では、無線ノードがどの隣接無線ノードからどの種類のメッセージを受信したかによって異なる状態に遷移する。このような、受信メッセージによる分岐が容易に記述できることが求められる。さらに、複数の隣接無線ノードをもつ無線ノードでは、これらの隣接無線ノードから非同期的にメッセージを受信することから、プロトコルに記述されたメッセージの受理順序とは異なる順序で物理的にはメッセージを受信することが考えられる。TCP/IP プロトコルに対する socket イ

ンタフェースでは、FIFO(First In First Out)あるいはFCFS(First Come First Serve)によりメッセージを受信するが、これを用いた場合にはプロトコル実装部分でメッセージのバッファリングによる受信順序の調整を記述しなければならない。これは、プロトコルの実装においてコーディング量を著しく増加させることとなる。そのため、本論文の提案するプラットフォームでは、受信メッセージのバッファリングと受理順序の調整機能を備えることとする。(2)のタイムアウトによる状態遷移では、受信待機状態にある無線ノードが一定時間以上必要なメッセージを受信しなかった場合に状態を遷移する。受信のタイムアウトを実現するためには、カウントを行なうタイマ機能と、タイムアウトまでに開発プロトコルが必要とするメッセージを受信したか判断する機能が必要となる。そこで、本論文の提案するプラットフォームではタイマ機能とメッセージの判別機能を実装することで、メッセージ受信待機中のタイムアウト機能を備えることとする。

4 実装

本論文で提案するプラットフォームの基本機能を Raspberry Pi 2 model B 上に実装した。ここでは、Raspbian オペレーティングシステム用のソフトウェアライブラリとして無線センサネットワークプラットフォームを構築している。これを用いて AODV ルーティングプロトコルを実装し、3-10 台の Raspberry Pi からなる無線センサネットワークにおけるルーティングが正しく行なわれることを確認した。また、構築したプラットフォーム上にメッセージの傍受を行なう Watchdog プロトコルを実装し、動作を確認した。

5 まとめ

本論文では、無線センサネットワークのためのネットワーク層プロトコルを開発、実装するためのプラットフォーム構築法を提案した。無線センサネットワークプロトコルの開発、実装のために、プラットフォームがメッセージの傍受とメッセージ受信順序の調整機能を備えることが求められることを示した。また、プラットフォームの基本機能であるメッセージの送受信関数、メッセージのフィルタリング機能、メッセージ受信中のタイムアウト処理をソフトウェアライブラリとして実装し、Raspberry Pi 上に実現した。

参考文献

- [1] Jain, S., "Review of Prevention and Detection Methods of Black Hole Attack in AODV-based on Mobile Ad Hoc Networks," International Journal of Information and Computation Technology, vol. 4, no. 4, pp. 381-388 (2014).
- [2] Karnapke, R. and Nolte, J., "BuckshotDV - A Robust Routing Protocol for Wireless Sensor Networks with Unstable Network Topologies and Unidirectional Links," Proceedings of the 8th International Conference on Sensor Technologies and Applications, pp. 60-65 (2014).
- [3] Oda, H., Kulla, E., Ozaki, R. and Nishihara, N., "Design of an Adhoc Testbed for IoT and WSN Applications Using Raspberry Pi," Proceedings of the 11th International Conference on Broadband and Wireless Computing, Communication and Application, pp. 535-546 (2016).
- [4] Sota, N. and Higaki, H., "Cooperative Watchdog for Malicious Failure Notification in Wireless Ad-Hoc Networks," Proceedings of the 8th IFIP International Conference on New Technologies, Mobility and Security (2016).
- [5] 金原, 石原, "GNU Radio を用いた無線アドホックネットワークにおけるデータ配信手法評価のための省スペース実験環境の構築," 情報処理学会論文誌, vol. 54, no. 8, pp. 2025-2035 (2013).