

無線ネットワークにおけるIPv6スタックの実現と移動ノードのエミュレーション

2005MT045 泉井 雄仁 2005MT070 水谷 由依
指導教員 後藤 邦夫

1 はじめに

無線センサネットワークの物理層，MAC 層にあたる部分は IEEE802.15.4[1] で規格が策定されている．2009 年 1 月現在で IPv4/v6 通信機能が実装されていない IEEE802.15.4 の IPv6 通信機能を実装したい．そのために，IEEE802.15.4 に準拠した無線センサデバイス Sun Small Programmable Object Technology (以下，Sun SPOT 略) [4] を用いる．Sun SPOT では実ノードと仮想ノードを用いた無線センサネットワークの構築を行うことができ，ネットワーク実験を行うことに適している．Sun SPOT をインターネットデバイスとし，無線センサネットワークとインターネットとの融合をはかる．本研究ではユーザ空間で IPv6 スタックを実現する．IPv6 スタックは UDP 通信と ICMPv6 通信を可能にすることを目標とする．

また，エミュレーションを正確なものにするため，IPv6 スタックとは別に座標付加アプリケーション開発も行う．センサノードに座標を定義し，ノード間の距離からフレームロス率を算出する．

本研究が完成すればユビキタスネットワーク社会が実現に近づき，全ての機器がインターネットに繋がることになる．

2 システム概要

本研究の対象である無線センサネットワークの特徴を述べ，用いるエミュレータの紹介と，システムの構成を述べる．

2.1 無線センサネットワーク

無線センサネットワークとは，センサノード（センサとデータ処理機能や無線機能を実装した装置）を広範囲に分布させ，ノード間で測定したデータを無線通信するものである．センサノードを自由に配置することができるようになり，多数のノードより同時にセンサ応答を取得できる．

無線ネットワークの規格について以下に述べる．

IEEE802.15.4

IEEE802.15.4 とは，2003 年に IEEE により承認された無線規格で，規格範囲は物理層と MAC 層のみとなっており，上位プロトコルは別途開発されている．ハードウェアでは電力供給，電池交換は考えていない低消費電力となっている．その結果，低出力，通信距離の短距離，低通信速度が特徴となっている．

lowPAN

low power PAN (以下，lowPAN 略) [2] とは低消費

電力で通信が可能なワイヤレス低価格通信ネットワークである．IEEE802.15.4 上に IPv6 をマッピングするための規格である．

2.2 Sun SPOT

Sun SPOT とは本研究で使用するデバイスで，OS を必要とせず，CPU 上で直接実行できる Java ベースのセンサ・アプリケーションを構築するための技術として開発された無線センサネットワークデバイスである．センサ・アプリケーションは，Sun SPOT デバイス上で直接動作することが可能なものである．また，通信規格は IEEE802.15.4 に準拠している．Sun SPOT はデバイスの機能を実装した仮想ノードを簡単に起動させることができる．仮想ノードを大量に起動させ相互通信を行うことによってネットワークエミュレータとして使用することができる．Squawk VM を使って Sun SPOT デバイス上で直接動作することが可能である．Squawk VM とはリソースの制限された小型デバイス向けに設計された Java 仮想マシンである．この VM は OS を必要とせず，Sun SPOT デバイス上で直接動作することが可能なものである．

2.3 システムの構成

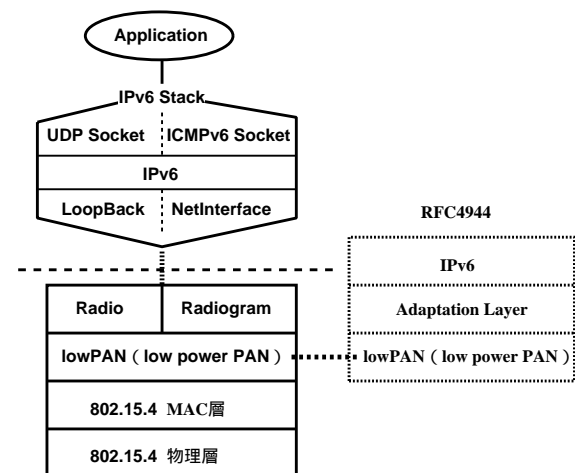


図 1 Sun SPOT の通信プロトコルスタックと RFC4944 スタック

図 1 Sun SPOT の通信プロトコルスタックでは IPv6 へのネットワークスタックが用意されていない．本来，IEEE802.15.4 における IPv6 通信は図 1 のように lowPAN の上から IPv6 へ繋がるが，Sun SPOT は IPv6 をサポートしていない．本研究ではユーザ空間で IPv6 スタックを実現する．ユーザ空間により IPv6 通信を可能

にすれば機種依存なく通信を行うことができる。IPv6を採用した理由は、RFC4944 Transmission of IPv6 Packets over IEEE 802.15.4 Networks [3] により上位プロトコルとして 6LoWPAN と呼ばれる IPv6 プロトコルの仕様が公開され、IPv6 利用が妥当なためである。すでに IPv6 通信を実装している機器が商品化されているが、機種依存があるため、機種依存のない IPv6 スタックが必要になると考える。本研究では通信速度は度外視し、機種依存のない IPv6 スタックと socket API を用意する。

また、Sun SPOT を用いた移動ノードのエミュレーションをより有効性のあるものにするために、位置情報を付加し、ノード間のフレームロス率から通信性能を評価する。

図 1 Sun SPOT の通信プロトコルスタックでは、2 つのプロトコル「Radio」「Radiogram」が用意されている。Radio はデータの紛失・重複や順序の入れ替えが起こらない信頼性の高いストリーム型のプロトコルであり、Radiogram は信頼性の低いデータグラム型のプロトコルである。本研究では処理の簡易さのため Radiogram を使用する。

2.4 開発環境

Sun SPOT には GUI ベースのツールとなっている IDE (統合開発環境) が用意されている。この IDE を用いることによって Sun SPOT の I/O などの機能をソフトウェアで実装したエミュレータである仮想ノードを簡単に起動させることができ、実ノードと仮想ノードを通信させることもできる。IDE を使った Sun SPOT の実ノードと、仮想ノードとの通信を図 2 に示す。

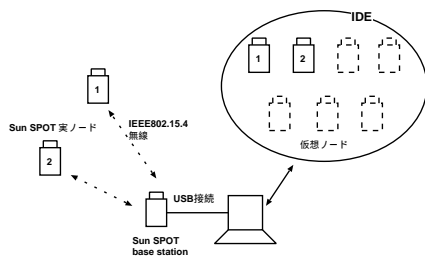


図 2 IDE を使った Sun SPOT デバイス、Virtual SPOT との通信

この通信に関しては、UDP ポートを使っていることが netstat コマンドによって確認された。仮想ノードを起動させることで、UDP ポートが開かれている。

Sun SPOT を使用しての開発は通常の Java 開発環境とは異なり特殊な制約がある。Sun SPOT API に従い使用できる関数が制限され、MIDlet を継承する。javac のコンパイラを ant で間接的に実行させアプリケーションをコンパイルする。コンパイルしたアプリケーションをノードに配置して実行する。

3 座標付加アプリケーション

Sun SPOT の端末エミュレータには位置情報が付加されていないという問題がある。実ネットワークのワイヤレスセンサネットワークでは位置情報から求められるノード間の距離は通信状態に大きく影響を与えるため重要となる。よって本研究ではアプリケーションレベルで位置情報を付加するために 2 次元直交座標付加アプリケーションを開発する。また、ノード間の距離からフレームロス率を定義することで通信性能を評価するおまかな指標となり、ノード間の通信状態を確認することができる。

3.1 送信側処理内容

送信側ノードに与える処理の手順を示す。

- 0 以上 30 以下の乱数を 2 次元座標とする
- アクセスがあった場合、ペイロードに座標情報を含め送信

ここで、座標の単位は m と仮定する。x, y 座標を 0 以上 30 以下と定めたのは IEEE802.15.4 の通信距離が 30m と定められているからである。この条件で 2 ノード間の最大通信距離は $\sqrt{30^2 + 30^2} = \text{約 } 30.74\text{m}$ が実現できる。これ以上の距離を使用してもロス率が非常に大きくなることからユーザ側で使用しないことを想定して定めた。

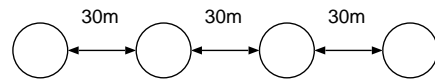


図 3 長距離通信

図 3 のように最大通信距離を満たしたノードを大量に設置することによって長距離通信も可能である。次にペイロードに含めるデータフォーマットを図 4 に示す。

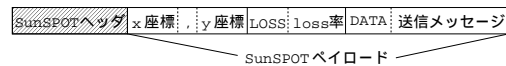


図 4 座標付加アプリケーションにおけるデータフォーマット

ペイロードには文字列の形式で座標情報を含め、データ分解の際のフラグとするために「,」「LOSS」「DATA」の文字列をそれぞれ含める。

3.2 受信側処理内容

受信側ノードに与える処理の手順を示す。

- 受け取ったペイロードから、座標情報を抽出
- 抽出した座標と自分の座標から、距離を求める
- 距離からフレームロス率を決定
- ロス率により受信データを削除

次にフレームロス率の決定方法を示す。誤り訂正符号を使用しないことを前提とする。単位面積あたりの電力

密度は距離の 2 乗に反比例し、ビットロス率は電力密度から

$$\text{ビットロス率} = (k \times \text{距離}^2) \quad (1)$$

(1) 式から

$$\text{フレームロス率} = 1 - \{1 - (\text{ビットロス率})^{\text{length}}\} \quad (2)$$

(2) 式をテーラー展開すると

$$\text{フレームロス率の近似値} = (k \times \text{距離}^2) \times \text{length} + 0.0001 \quad (3)$$

よってフレーム長に比例する。また、距離が 0 の場合であってもフレームロス率 0 であることを考慮し 0.0001 を加える。距離が 30 以上だった場合 ビットロス率 = 0.2 で固定する。無線通信は長距離になるにつれてビットロス率が急激に増加し信頼性が失われる。ビットロス率 = 0.2 つまりビットロス率 20% は、受信データのビットロス率が指数関数により急激に増加する直前の値と仮定して定めた。

計算されたフレームロス率によって、図 4 座標付加アプリケーションにおけるデータフォーマットの中のメッセージ部分からロス率分だけ削除し、表示させる。

4 IPv6 スタックの実現

IPv6 通信の最低限の機能として、ICMPv6 隣接ノード発見, ECHO と UDP 通信だけを実現する。TCP 通信は処理が複雑であるため、実現を見送った。

4.1 プログラム処理内容

IPv6 スタックの実現において、IPv6 スタックインスタンスの作成を行うクラス, Socket の入出力データ構造や操作を行うクラスを用意した。

IPv6 スタックインスタンスの作成クラスでは API として UDP Socket, ICMPv6 Socket を用意し、IPv6 スタックを構成することで IPv6 パケットの送受信などを行う。用意したプログラム階層とクラスの繋がりを示したものを図 5 に示す。図 5 で

- ・ UDP Socket : UDP 通信を行うための API
- ・ ICMPv6 Socket : ICMPv6 通信を行うための API
- ・ IPv6 : IPv6 パケットのデータ形式をとり、IPv6 通信を行う
- ・ NetInterface : リンクローカルアドレスの設定、ICMPv6 の処理、データ挿入の処理
- ・ LoopBack : 宛先に loopback address が指定されたときの処理

を示している。

各々の受信メソッドでは、thread でポート別 queue を定期的に監視し、queue にデータが入ったら thread を終了して受信する。

また、Socket 入出力のためのデータ構造や操作のクラスではフレームアドレス、IPv6 ヘッダ情報、UDP/ICMPv6 ヘッダ情報、ペイロードを処理の対象

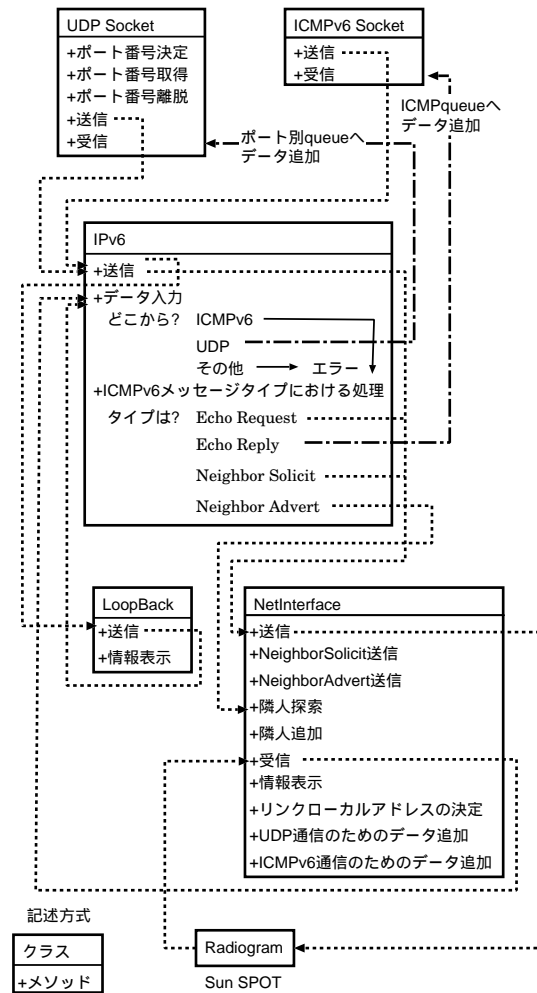


図 5 IPv6 スタックの実現におけるプログラム階層とクラスの繋がり

としている。ヘッダ情報として、目的地 MAC アドレス、発信源 MAC アドレス、Ether タイプ = 0x86DD、ペイロード長、next header、ホップリミット、目的地アドレス、発信源アドレス、目的地ポート、発信源ポート、ICMPv6 タイプ、ICMPv6 コード、checksum を用意した。ここで、IPv6 用のポート番号を独自で定めた。Radiogram でシステム予約されているポート 0-31 の中に IPv6 用のポート番号がなく、Sun SPOT 独自のものであったためである。IEEE802.15.4 に準拠している Sun SPOT で扱える最大ペイロードサイズは 1260bytes で、Radiogram 通信では 200bytes 以上のデータはロスする可能性がある。データを送信する際、分割する処理を用意した。

4.2 ルータ機能の実現

第 2.4 節図 2 で示された Sun SPOT 実ノードと、仮想ノードとの通信には UDP ポートが開かれていることが分かっている。ルーティング機能については、PC につ

ないだ Sun SPOT base station で必要な機能だが、PC と UDP トンネルで繋ぐことで、実際のルータの処理は PC で行う。tuntap と接続し、受信した IPv6 パケットをそのまま PC ヘトンネリングする。

4.3 Sun SPOT で動作させるための入出力クラスを用意

現在、オブジェクトの送信によって IPv6 スタックの動作を行っている。

Sun SPOT での通信のために、変更点を以下に示す。

- Radiogram 通信インターフェースの作成
- リンクローカルアドレスの決定方法の変更
- MIDlet の継承

Radiogram 通信インターフェースの作成

IPv6 スタックで定義された IPv6 パケットを送受信する。

送信

Radiogram の形式で宛先を指定する。宛先がマルチキャストなら Radiogram の broadcast を使用するように変更する。

目的地アドレスがマルチキャストであるかどうかの判断は、IPv6 スタック中の Neighbor Solicit の送信と Socket データ構造中で boolean multicast を追加する。

また、データパケットは byte 配列に変換し、ヘッダフレームは自動で構成される。

受信

受け取った IPv6 パケットを分解する。

発信源 MAC アドレスの取得には Sun SPOT API のメソッドを使用し、目的地 MAC アドレスの取得は受け取った IPv6 パケットから区別可能である。

Neighbor 処理については、MAC アドレスを Ethernet の 6bytes しか代入していないため、IEEE802.15.4 の MAC アドレスの長さ分の代入が必要である。

リンクローカルアドレスの決定方法の変更

RFC4944 による IEEE802.15.4 の IPv6 リンクローカルアドレスを図 6 に示す。

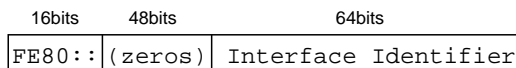


図 6 IPv6 リンクローカルアドレスの形式

IPv6 スタックプログラムの NetInterface クラス中で、リンクローカルアドレスの決定メソッドを用意している。この箇所は現在 Ethernet アドレスが与えられたときの設定になっているため、IEEE802.15.4 の拡張 MAC アドレス 64bits を図 6 の Interface Identifier に入れる。

MIDlet の継承

main プログラムで javax.microedition.midlet.MIDlet を継承する。MIDlet アプリケーションとして作成することで、Sun SPOT 上で動作が可能になる。

5 実験

本研究にて実現させた Sun SPOT エミュレーションへの位置座標付加アプリケーション、IPv6 スタックを動作させた。実験にあたって、JDK 1.3、Sun SPOT SDK 3.0 (purple) を使用した。

5.1 実験結果

Sun SPOT への位置座標付加アプリケーション動作確認のために 3 ノード間の通信を行った。結果、無線センサネットワークにおける通信性能を示すことが確認できた。ノード間の距離が 30 以内の範囲ではほとんどロスはなく、距離が短くなる程ロス率は小さくなるという理論が実現できた。また、IPv6 スタック動作確認のためにループバックアドレス、tunnel/tap デバイス、2 つの IPv6 スタック間でそれぞれ ICMPv6 ECHO を用いて ping6 を行った。オブジェクト送信の確認ができた。

6 おわりに

本研究の主な成果物を以下に述べる。

1. Sun SPOT エミュレータへの位置座標付加アプリケーション
2. IPv6 スタック実現のためのアプリケーション

Sun SPOT エミュレータへの位置座標付加アプリケーションに関しては今後、既存のネットワークエミュレータに対応させ、delay、lossなどを加えることで本格的なネットワークエミュレーションに発展させることができる。

IPv6 スタックに関しては TCP 通信機能が含まれていない。また、Sun SPOT にルータ機能を実装させることができず、ルータ機能は PC の OS (データリンク層)で行った。今後は IPv6 スタックの TCP 通信機能の実装と、ルータ機能を Sun SPOT に実装することが目標となる。

参考文献

- [1] IEEE Computer Society: *IEEE Standards 802.15.4* (2003). <http://standards.ieee.org/getieee802/download/802.15.4-2003.pdf>.
- [2] Kushalnagar, N., Montenegro, G. and Schumacher, C.: *IPv6 over Low-Power Wireless Personal Area Networks (6LoWPANs): Overview, Assumptions, Problem Statement, and Goals*, RFC4919 (2007). <http://www.ietf.org/rfc/rfc4919.txt>.
- [3] Montenegro, G., Kushalnagar, N., Hui, J. and Culler, D.: *Transmission of IPv6 Packets over IEEE 802.15.4 Networks*, RFC4944 (2007). <http://www.ietf.org/rfc/rfc4944.txt>.
- [4] Sun microsystems, Inc.: Sun SPOT 無線センサネットワークデバイス (accessed May 2008). <http://jp.sun.com/products/software/sunspot/>.