

Title	センサネットワークにおける受信信号強度を用いた低遅延・高信頼型ルーティングアルゴリズムの構築
Author(s)	竹内, 常哲
Citation	
Issue Date	2007-03
Type	Thesis or Dissertation
Text version	author
URL	<a href="http://hdl.handle.net/10119/3618">http://hdl.handle.net/10119/3618</a>
Rights	
Description	Supervisor: 丁 洛榮, 情報科学研究科, 修士

修 士 論 文

センサネットワークにおける受信信号強度を用いた  
低遅延・高信頼型ルーティングアルゴリズムの構築

北陸先端科学技術大学院大学  
情報科学研究科情報システム学専攻

竹内 常哲

2007年3月

## 修士論文

# センサネットワークにおける受信信号強度を用いた 低遅延・高信頼型ルーティングアルゴリズムの構築

指導教官 丁洛榮 助教授

審査委員主査 丁洛榮 助教授  
審査委員 松澤照男 教授  
審査委員 Defago Xavier 助教授

北陸先端科学技術大学院大学  
情報科学研究科情報システム学専攻

310063 竹内 常哲

提出年月: 2007年2月

## 概要

本稿では、ユビキタスコンピューティングを実現するための重要な技術の1つであるセンサネットワークについて、従来の無線アドホックセンサネットワークとの違いについて述べるとともに、低遅延・高信頼型のセンサネットワーク用ルーティングアルゴリズムの構築を行う。また、構築したルーティングアルゴリズムについて、シミュレーションにより、その動作性能の検証を行う。

# 目次

第1章	序論	1
1.1	研究の背景	1
1.2	研究の目的	2
第2章	センサネットワークの概要	3
2.1	センサネットワークとは	3
2.2	センサネットワークのプラットフォーム	4
2.3	アドホックネットワークとの違い	4
第3章	無線ネットワーク用経路制御プロトコル	6
3.1	経路制御プロトコルの分類	6
3.1.1	リアクティブ型プロトコル	7
3.1.2	プロアクティブ型プロトコル	8
3.1.3	ハイブリッド型プロトコル	9
3.2	その他の経路制御プロトコル	9
3.3	ノード同士の位置関係を把握するためのハードウェア	10
3.4	センサネットワークに適したルーティングアルゴリズム	11
第4章	センサネットワーク用経路制御プロトコル	12
4.1	想定する環境とセンサネットワークに必要な特性	12
4.2	受信信号強度を用いた経路制御プロトコル	12
4.2.1	NOTIFICATION フェイズ	14
4.2.2	COMMUNICATION フェイズ	21
第5章	シミュレーションによるプロトコル性能評価	23
5.1	シミュレーション条件	23
5.1.1	CBR 法によるパケット生成	24
5.1.2	電波強度モデル	25
5.1.3	通信距離半減エリアの設定	25
5.2	シミュレーション結果と考察	25
5.2.1	データパケット到達率	26
5.2.2	パケット平均遅延時間	26

5.2.3	パケット最悪遅延時間 . . . . .	27
5.2.4	シミュレーションのまとめ . . . . .	27
<b>第6章</b>	<b>結論</b>	<b>35</b>
6.1	本研究のまとめ . . . . .	35
6.2	今後の課題 . . . . .	35

# 第1章 序論

## 1.1 研究の背景

近年、無線通信技術や半導体技術の飛躍的な向上とともに Mark Weiser [1] が提唱したユビキタスコンピューティングについての研究が盛んに行われている。ユビキタスとは、“遍在する” という意味のラテン語であり、ユビキタスコンピューティングとは、環境内のあらゆる場所や物体に対して、コンピュータが内蔵されている状況をさす。これらのコンピュータが互いに協調的な動作をすることで、ユーザにはコンピュータを操作している印象を与えることなく、それらの機能を享受させることが可能となる。ユビキタスコンピューティングは実世界とのインタラクションを特徴とするので、実世界から情報を取得するセンシングが不可欠となる。そのため、これらのセンサによって構成されるセンサネットワークは、ユビキタスコンピューティングを実現するための重要な技術の一つとなっている。

具体的なユビキタスコンピューティングの実証実験としては、リビングルームのインテリジェント化を試みるマイクロソフト研究所の Easy Living[2] などが挙げられる。



図 1.1: Easy Living

このようなインテリジェント化された空間 (インテリジェントスペース) では、多数のセンサが環境内に埋め込まれており、これらのセンサが協調してセンシングを行うことで、人間が部屋に入って来た場合には、灯りを点けて空調を整える、ソファに座るとその人の好みの音楽を流す、などの動作を行い、人間に対して快適な環境を提供することが試みられている。また、センサネットワークの利用具体例としては、温度や湿度、日照等のデータを収集したり、森林や海洋等にセンサを散布し、環境汚染状況などの調査に用いられている。

## 1.2 研究の目的

そこで本研究では、ユビキタスコンピューティングを実現するために必要なセンサネットワークにおいて、環境内に設置された複数のセンサから情報を収集することを目的として、センサネットワークにおける低遅延・高信頼型ルーティングアルゴリズムの構築を行う。



## 第2章 センサネットワークの概要

### 2.1 センサネットワークとは

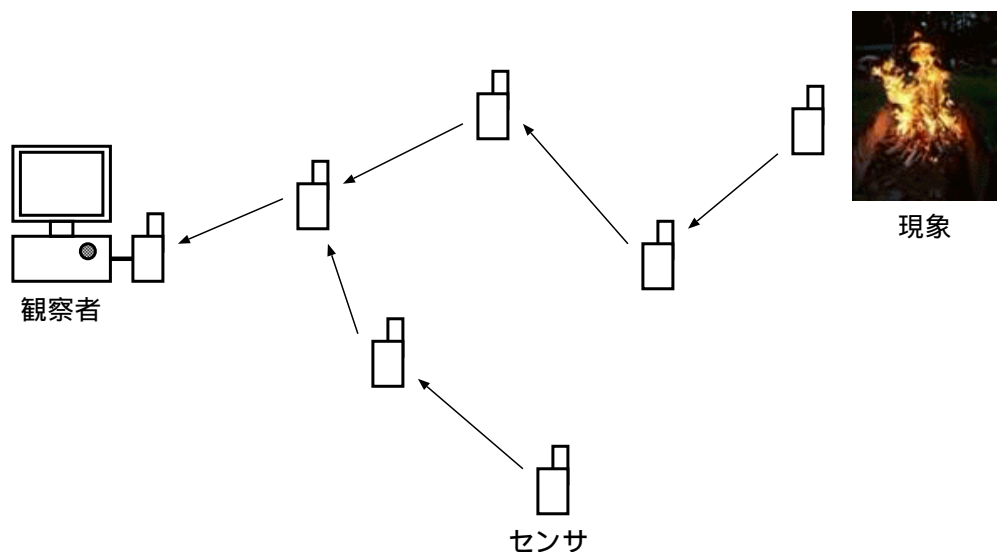


図 2.1: センサネットワークの構成

センサネットワークは、複数のセンサノードと、観察者と呼ばれる、センサノードから送信されるセンサ情報を収集するノードにより構成される。図 2.1 に示すように、一般的なセンサネットワークでは、各センサノードは自身に搭載されたセンサを用いて環境情報をセンシングするとともに、他のノードから送信されたパケットを中継することで、マルチホップによる情報の転送が行われる。図 2.1 のなかで現象と呼ばれているものは、センサにセンシングされる対象を意味しており、この例に示したように、対象が火事(火災)である場合は、センサにおいてそれが感知された場合にのみ、その情報が伝えられる。また、センシング対象が気温や湿度といったものの場合には、定期的にセンシングされた情報が観察者のもとに送信される。

センサネットワークの特徴は、複数のセンサノードが検出した情報を観察者が集約することにより、センサネットワークからの情報を利用するユーザにとっては、1つのセンサとしてみなすことができる点にある。

## 2.2 センサネットワークのプラットフォーム

現在、各種のセンサネットワークプラットフォームが開発されており、その中でも、最も早くに市販されたものが crossbow 社が開発した MICA MOTE(図 2.2(a),(b)) である。MICA MOTE の最大の特徴は tinyOS と呼ばれる OS が搭載されている点であり、価格も 1 つ数万円と高価なものとなっている。図 2.2(c) は ymatic 社が開発したセンサノードである S-NODE である。このノードは CPU として PIC16 が搭載されており、アセンブリコードによるプログラムが可能である。

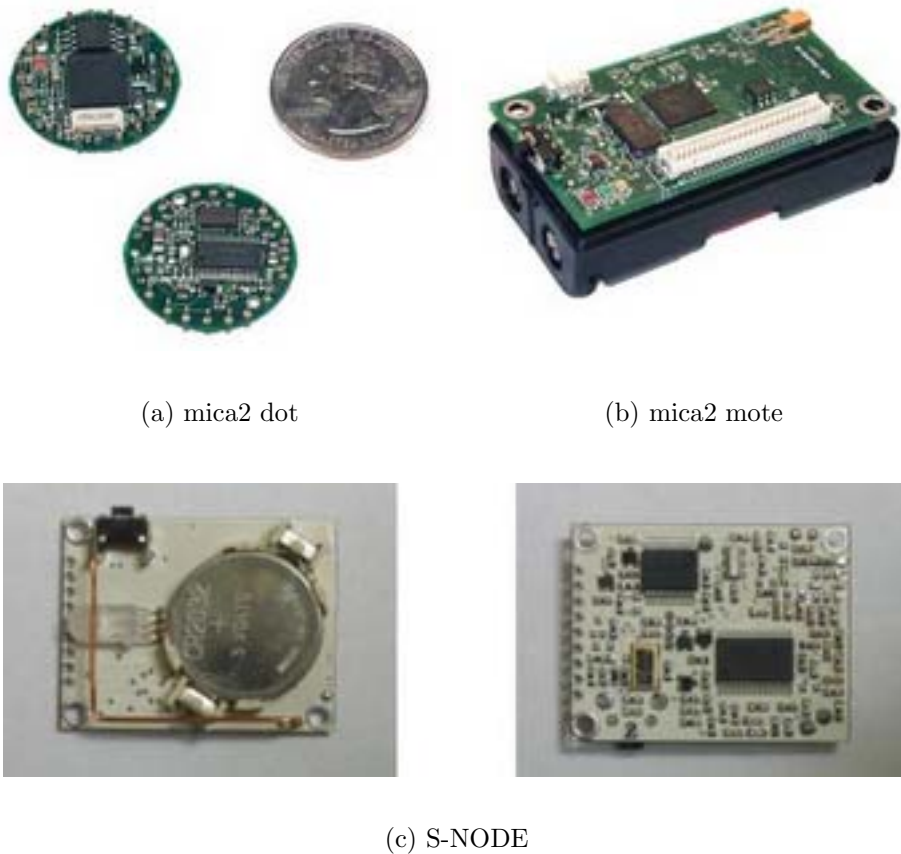


図 2.2: センサネットワークのプラットフォーム

## 2.3 アドホックネットワークとの違い

アドホック (ad hoc) とは、“その場限りでの” という意味のラテン語であり、アドホックネットワークでは、無線 LAN や携帯電話網のように基地局を必要とすることなく、端

末同士がピアツーピアでネットワークを構築する。アドホックネットワークにおいては、端末が移動することが前提とされているため、無線による通信が主流となっている。また、マルチホップにより、端末がデータパケットの中継を行うことで、電波信号が直接到達しない端末との通信が可能となる。このような性質を持つアドホックネットワークとセンサネットワークとの違いについて表 2.1 に示す。センサネットワークが図 2.2 に示したセンサノードによって構成されるのに対して、アドホックネットワークではノート PC や携帯電話、PDA などのモバイル製品によって構成されるため、センサネットワークでは、個々のノードの処理速度や通信速度が、アドホックネットワークに比べて劣る。また、アドホックネットワークではすべてのノードがデータパケットの宛先 (シンク) となるのに対して、センサネットワークでは、観察者のみがデータパケットの収集を行う。さらに、センサネットワークでは、ノードの移動が無い、もしくは少ない点や、ネットワーク中に存在するノードの密度が、アドホックネットワークに比べて高い、などの点が違いとして挙げられる。

表 2.1: 2つのネットワークの違い

	センサネットワーク	アドホックネットワーク
構成デバイス	センサノード	ノート PC、携帯電話、PDA など
ノードの処理能力	低い	高い
デバイス通信速度	数 ~ 数百 kbps	数 ~ 数十 Mbps
シンク	観察者のみ	全てのノード
ノードの移動	無し or 少ない	多い
ノードの密度	高い	低い

# 第3章 無線ネットワーク用経路制御プロトコル

前章では、センサネットワークの概要を説明し、アドホックネットワークとの違いについて明らかにした。しかし、これらの技術はセンサネットワークにとっても非常に有用である。そこでこの章では、無線ネットワークでの利用が想定されている経路制御プロトコルについての概要を示す。

## 3.1 経路制御プロトコルの分類

経路制御を行うプロトコルは、大きく分けて2つに分類することができる。1つは通信を開始する際に経路を探索するリアクティブ型プロトコルであり、もう1つは定期的な情報交換を行い、その情報をもとに事前に通信経路を決定するプロアクティブ型である。また、その2つを組み合わせたハイブリッド型のプロトコルも存在する。図3.1では、それぞれの型における主要なプロトコルを列挙しているが、この他にも多数の経路制御プロトコルが存在する。

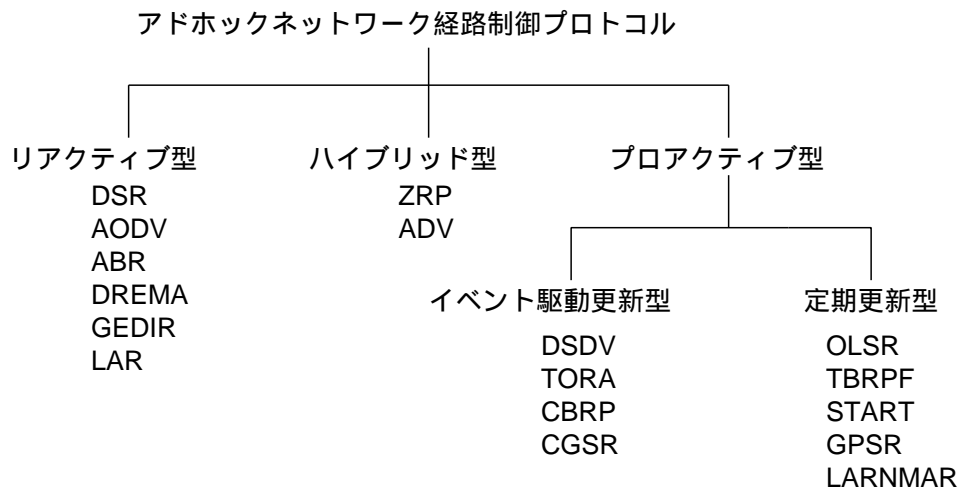


図 3.1: アドホックネットワーク経路制御プロトコルの分類

### 3.1.1 リアクティブ型プロトコル

リアクティブ型のプロトコルでは、通信開始時に経路を探索することになるため、通信要求が無い場合には制御パケットは全く送信されない。しかし、複数のノードから短期間のうちに通信要求があると、データパケットと制御パケットがネットワーク内で入り乱れることになる。なお、一度確立された通信経路については、再利用を行うことで、ネットワークを流れる制御パケットの増大を防ぐ。このようなリアクティブ型のプロトコルは、一般的にノード数が少なく、ノードの移動が頻繁に発生するネットワークに対して有効なプロトコルであると言える。

- DSR (Dynamic Source Routing)

DSR[14] は歴史の古いプロトコルであり、主な特徴として、送信ノードが宛先ノードまでの経路をすべてパケットヘッダに含んで送るソースルーティング方式を採用している点が挙げられる。宛先のわからないノードと通信する場合、送信ノードはネットワーク内に経路要求 (RREQ:Route REQuest) をフラッディングする。フラッディングとは、ブロードキャストを利用して到達可能なネットワーク内の全てのノードにパケットを伝搬する仕組みである。パケットには識別子が付加されており、各ノードは受信したパケットを一度だけブロードキャストするという規則でパケットの中継が行われる。各ノードはRREQを中継する際に、RREQ内に自分のアドレスを付加して中継する。宛先ノードに届いたRREQには、これまで経由してきたすべてのノードのアドレスが順番通りに付加していることになる。宛先ノードは、この経路の逆をたどるソースルーティングを利用して、経路応答パケット (RREP:Route REPLY) を送信ノードに返信して取得した経路を伝える。各ノードはデータ中継の際に、次のホップへの中継が成功したか否かを必ず確認しなければならない。確認方法は実装依存であるが、確認応答 (ACK) を利用する方法が一般的である。データ中継の成功が確認できない場合、経路エラー (RERR:Route ERror) を送信ノードに向けて送信する。RERRを中継するノードは経路エラーが起こったノード間のリンクを含む全ての経路情報を経路表から消去する。

- AODV (Ad-hoc On Demand Vector)

AODV??の主な特徴は、各ノードが次にどのノードにパケットを送れば良いかという経路表を保持している点である。すなわち、各ノードは次の中継先だけを知っており、その後はどのように宛先にパケットが届けられるのかは関知しない。宛先ノード (D) までの経路を知らない送信ノード (S) は、RREQをネットワーク内にフラッディングする。RREQパケットを中継するノードは、パケットを中継してきたノードをS宛の経路の次ホップノードとして記録する。RREQの処理によりSへの経路情報が全てのノードに構築される。この構築された経路を利用して、DはSに対してユニキャストでRREPを送信する。RREQの処理と同様に、RREPを中継するノードはD宛ての経路の次ホップノードを知ることができる。RREPがSに届いた時点で、SとDの双方向経路が確立する。各ノードは、RREPの受信時に、前後のノードをprecursorリストに追加する。こ

のリストは、その経路を利用する近隣ノードのリストであり、経路エラー処理に利用される。AODVにおいてもDSRと同様に、各ノードは次のノードに対する経路の有効性を確認しなければならない。もし確認できない場合は、RERRをデータ送信元に対して送信することになる。無効となったノードへのリンクを次ホップとしている宛先ノードの経路を全て無効とする必要がある。さらにprecursorリストには、その経路を利用する近隣ノードが記録されている。それらのノードは、無効となった宛先に対する経路を保持している可能性が高いため、precursorリストのすべてのノードにRERRを送る。

### 3.1.2 プロアクティブ型プロトコル

プロアクティブ型のプロトコルでは、事前に通信経路を確立するために、各ノードにおいて定期的な情報交換が行われる。その際に、いかにして交換する情報量を抑えるかが重要なポイントとなる。このようなプロアクティブ型のプロトコルは一般的にノード数が多く、移動が少ないネットワークにおいて有効となる。このように、リアクティブ型とプロアクティブ型ではそれぞれが相反する特性を持つ。

- OLSR (Optimized Link State Routing)

OLSR[5]の最大の特徴は、マルチポイントリレー (MPR) 集合と呼ばれる仕組みを利用してフラッディングを効率化している点である。従来のフラッディングは全てのノードが必ずパケットを一度だけ中継するのに対して、MPR集合を利用したフラッディングでは、必要最低限のノードだけがパケットを中継する。各ノードは、自分がどのノードのMPRになっているのかを知っており、そのノードからのパケットだけを中継する。MPRの選択にあたっては、まず各ノードは近隣ノードとのHelloメッセージの交換により、2ホップ先までの経路情報を収集する。1ホップで接続するノードを集合 $N_1$ とし、2ホップで接続するノードを集合 $N_2$ とする。そこで、各ノードは集合 $N_2$ の全てのノードに情報を伝えるには、集合 $N_1$ の中のどのノードが最低限、情報を中継すれば良いのかを考える。その手順としては、まず最初に集合 $N_1$ の中から、このノードを経由しない限り、集合 $N_2$ の特定のノードに対して情報伝達が不可能となるノードをMPRとして選択する。次に最も多くの集合 $N_2$ ノードに対して、情報を伝達可能なノードを集合 $N_1$ の中から選択する。この動作を繰り返すことで、最終的に集合 $N_1$ の中からMPR集合が選択される。各ノードが送信するHelloメッセージには、自分のMPR集合のリストも含まれている。このリストから、各ノードは自分がどの近隣ノードのMPRであるかを認識することができる。このMPRは、フラッディングを行う際に利用され、自分がMPRとして選択されているノードからのパケットだけを中継する。実際にこのMPRを利用したフラッディングは、経路表の作成に利用されている。MPRとして選択されたノードは、自分がどのノードのMPRであるかというリストをフラッディングする。あるノードが選択しているMPR集合がわかるということは、そのノードへの経路の1つ手前のノードがわかるということである。さらにその1つ手前のノードのMPR集合

もわかるため、最終的に各ノードはネットワーク全体のトポロジ構成を認識することが可能となる。実際にデータパケットを転送する際には、この経路表に従って送信される。

- TBRPF (Topology Broadcast Based on Reverse-Path Forwarding)

プロアクティブ型のプロトコルは、ノードの移動によってトポロジが変更された場合、ネットワーク内の全てのノードの経路表に影響を与える可能性がある。TBRPF[4]では、このようなトポロジの変更処理コストを最小限に抑えるために、安定したリンクの選択やトポロジの差分情報を積極的に利用している。TBRPFでは、最終的にネットワーク内の全てのノードに対する最短経路である送信者木 T を作成する。全てのノードが同じトポロジから作成された T を保持することで、どのノードとも通信ができることになる。同じトポロジを共有する最も簡単な方法は、各ノードが自分の隣接ノードのリストをネットワーク内の全てのノードに伝えればよいというものである。しかし、この方法ではトラフィック量が膨大になるため効率が悪い。そこで、Tの部分集合となる報告部分木 RT と呼ばれる情報を定期的を送信する。さらに、より細かな時間粒度で、RTに追加・削除の変更があった場合の差分情報を送信する。RTの全体情報を送信するのは、新規にネットワークに加わったノードには差分情報だけでは充分ではないからである。このRT情報は、隣接ノードにのみ送信され中継はされない。しかし、この情報が隣接ノードのRTに変更を加えた場合、隣接ノードは自らのRT情報をさらに隣接ノードに送信する。TBRPFのこのような処理が、純粋なフラッディングとは異なる点が特徴的といえる。

### 3.1.3 ハイブリッド型プロトコル

リアクティブ型の経路制御プロトコルとプロアクティブ型の経路制御プロトコルの両方の特性を持つものが、ハイブリッド型の経路制御プロトコルとなる。そのため、2つのプロトコルの利点と欠点を有することになる。ハイブリッド型のプロトコルであるZRPでは、近隣ノードとの通信経路は、プロアクティブ型の動作により定期的に更新するが、数ホップ先のノードとの経路については、通信を行いたい場合に限りリアクティブ型の動作により、通信経路を確保する。このようにハイブリッド型のプロトコルは経路制御が複雑になることから、一般的に性能が悪く、現段階では実用に耐えうるものは存在しない。

## 3.2 その他の経路制御プロトコル

従来型のネットワークとは異なり、アドホックネットワークでは、ノードが移動することが想定されているため、アドレスに依存した経路制御とは異なる形の経路制御の仕組みが必要となる。そこで、各ノードの位置情報を利用した経路制御方法が考えられる。この経路制御の基本的な仕組みは、まず各ノードが近隣ノードに対して自分の位置情報を定期的に通知する。データを転送する場合、宛先ノードの位置に最も近い近隣ノードに対して

データを転送する。データを受信したノードは、さらに宛先ノードの位置に最も近い近隣ノードにデータを転送する。このような処理を繰り返すことにより、最終的に宛先ノードまでデータが届くことになる。このようなデータ転送方式を Greedy Forwarding[15] と呼ぶ。Greedy Forwarding の仕組みでは、たとえ周辺に通信可能なノードが存在したとしても、必ず自分より宛先に近いノードに対してデータを転送しようと試みる。そのため、この条件に適合するノードが存在しない場合はデータの転送は行われず、通信が途絶えることになる。このような方法を避けるために、いくつかの提案がなされている。GPSR[16] では、Greedy Forwarding により中継先がなくなった場合には、一時的に Perimeter Forwarding と呼ばれるモードに切り替わる。Perimeter Forwarding に切り替わったノードは、自分の位置情報をデータパケット内に記録し、このノードを出発点として、近隣ノードで作られた多角形を右回りで周回するようにデータが転送される。このデータパケット内に記録された座標よりも、宛先ノードの位置に近い近隣ノードを保持するノードに到着すると、そこで再度 Greedy Forwarding に切り替わり転送される。

### 3.3 ノード同士の位置関係を把握するためのハードウェア

前節で示したノードの位置関係を利用した経路制御を行う場合には、各ノードはあらかじめ自分の位置情報を把握しておく必要がある。ノードが位置情報を知る方法の1つとしては、人間が各ノードに対して直接位置情報を与える方法が考えられる。しかし、ノードが多数存在する場合は、手間のかかる作業となる。また、ノードが移動する際には、位置情報を更新する必要がある。そのため、ノードが自律的に位置情報を認識する方法がいくつか提案されている。もっとも一般的な方法としては、GPS が考えられる。ノードが GPS システムを搭載することで、正確な位置情報を取得することが可能となる。しかし、ノードの高価格化や、GPS 電波の到達性の問題から屋内での使用が難しいと言った問題がある。また、各ノードが三角測量を行うことで、互いの位置情報を認識する方法も考え出されている。三角測量を行う際には、超音波を使用する方法や指向性を持つアンテナを用いる方法が提案されている。超音波を用いる場合は、電波と同時に送信することで、電波と音波の速度差を利用してノード間の距離を検出することができる。その距離情報をもとに各ノードが三角測量を行うことで、自分の位置を認識することができる。しかし、超音波を用いる際には、ノード間に遮蔽物がなく、ノードが見通せることが必要となる。また、指向性を持つアンテナを利用する際には、各ノードが電波の到来角を検出することで三角測量が行われる。この方法を用いるには、アンテナを回転させるモーターや、1つのノードに対して複数本のアンテナを搭載する必要がある。

これらの位置情報を特定する方法は、すべて特殊なハードウェアを使用することで成り立っている。そのため、大量のノードによって構成されるセンサネットワークの場合は、特に負担の大きなものとなる。



### 3.4 センサネットワークに適したルーティングアルゴリズム

この章では、無線ネットワークにおける経路制御プロトコルの概要を示した。ここでは、センサネットワークの特性に適合するプロトコルについて考察を行う。まず、センサネットワークの特徴としては、ノードの数が多く、移動が無い、もしくは少ないため、プロアクティブ型のプロトコルが適当であると考えられる。また、最終的な宛先となるノードは、データ収集を行うノード(観察者)に限られるため、各ノードは、観察者に対してデータを送信する際には、どのノードを次ホップとしてデータ転送を行えば良いのかを知るだけで良く、経路表のサイズを抑えることが可能となる。

従来のネットワークとは異なり、センサネットワークでは、ノードの位置情報を経路制御に利用することができる。そこで、PIC等に搭載されているA/Dコンバータを利用して受信信号強度から簡便にノード間の距離を推定することができる。次の章では、この受信信号強度を用いたプロアクティブ型のルーティングアルゴリズムの構築を行う。

# 第4章 センサネットワーク用経路制御プロトコル

前章では、無線ネットワーク上で用いられる経路制御プロトコルについて、その概要を示した。現在までに多数の経路制御プロトコルが提案されているが、どれも決定的なプロトコルにはなり得ていない。その大きな原因は、ネットワークの性質によって求められるプロトコルの特性が大きく異なるためであり、どのような状況下であっても最適な性能を発揮するプロトコルは存在しないからである。そこで本章では、センサネットワークが用いられる環境とネットワークに求められる特性を想定し、センサネットワークの性質に特化した経路制御プロトコルを構築する。

## 4.1 想定する環境とセンサネットワークに必要な特性

2.3節で示したように、センサネットワークはアドホックネットワークで想定されているノードに比べてハードウェアの性能が低く、また、ノードが設置される密度も高いと言える。図 4.1 では、一辺が 30m の 2 次元空間に 200 個のセンサノードがランダムに設置されている。各センサノードの通信可能な距離を 10m と設定した場合、図 4.2 に示すように、非常に多くの通信経路が存在することになる。そこで、これら多数の通信経路の中から、通信効率の良い経路を選択することがプロトコルに求められる。さらに、センサネットワークに求められる特性としては、

- 通信遅延が少ない
- 通信信頼性が高い

点が挙げられる。これは、火災などのイベントが発生した場合には、素早く、しかも確実にデータ収集ノード (以下、DCN) に情報を伝える必要があるためである。そこで、これらの要件を満たすために、ハードウェア資源が制限されたセンサノードにおいても、比較的簡単に検出できる信号受信強度を用いて経路制御プロトコルの構築を行う。

## 4.2 受信信号強度を用いた経路制御プロトコル

前節で明らかになった、センサネットワークに必要な要件を満たすために、経路制御プロトコルを notification フェーズと communication フェーズの 2 つのフェーズに分けて

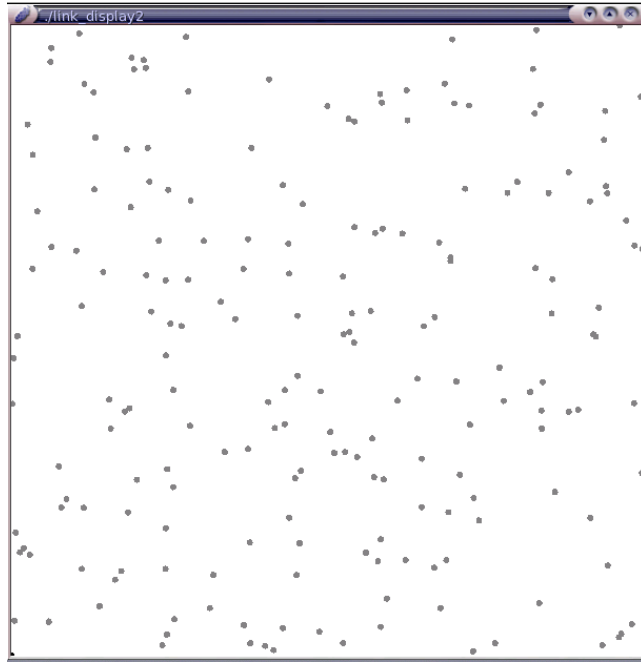


図 4.1: センサノードの設置

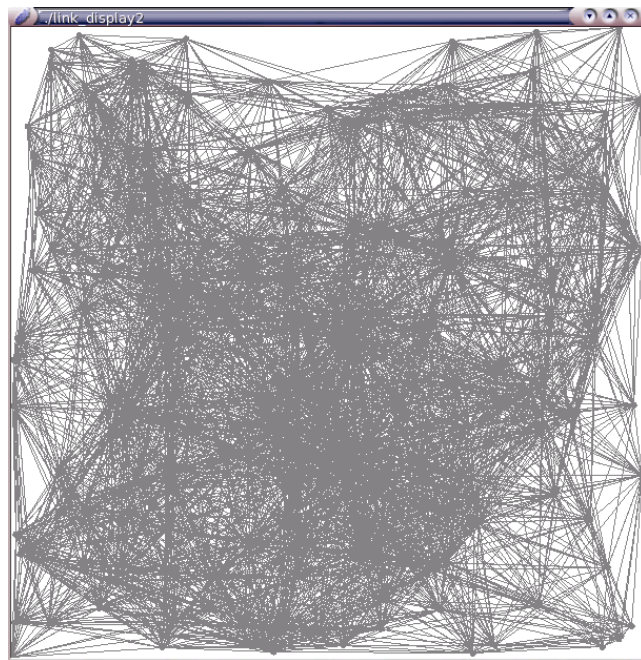


図 4.2: 通信可能な全ての経路

実現する。notification フェーズでは、各ノードが受信信号強度を利用して DCN への通信経路の選択を行う。その際に、各ノードは通信距離が異なる 2 種類の経路を確保する。また、communication フェーズでは、各ノードが検出した情報を、notification フェーズで確保した通信経路を利用して DCN へと送信を行う。このようなプロアクティブ型のプロトコルにすることで、制御パケットとデータパケットが送信されるタイミングをずらすと同時に、通信距離の異なる経路を使い分けることで低遅延・高信頼型の経路制御プロトコルが実現可能となる。具体的なプロトコルの動作について以下に示す。

#### 4.2.1 NOTIFICATION フェイズ

notification フェーズでは、各ノードは notification 信号を一度だけ送信する。各ノードが、notification 信号を送信するタイミングを決定する方法について図 4.3 に示す。notification 信号は [自身の ID, DCN までの距離, DCN までの Hop 数] によって構成され、この信号を送信することで周囲のノードに自分の存在を告知することができる。センサネットワークを構築するにあたり、まず始めに DCN が notification 信号を送信する。これにより、DCN の周囲に配置されたノードが DCN からの notification 信号を受信することができる。図 4.3 では、ノード B と C が DCN からの notification 信号を受信している。ここで、ノード B と C は、受信した信号強度をもとに、DCN との距離を推定することができる。この推定された距離をもとにノード B, C は自身が送信する notification 信号の送信タイミングを決定する。送信タイミングは DCN との距離が近いノードほど早くなる。また、一定の距離ごとに複数のタイムスロットが設定されており、各ノードがランダムにスロットを選択することで、同一距離区間に存在する複数のノードが同じタイミングで notification 信号を送信することを抑制する。しかし一方で、ノード D のように DCN からの notification 信号を直接受信することができないノードが存在する。そこで、ノード D はノード B, C が送信する notification 信号を利用して DCN との距離の推定を行う。その推定方法について図 4.4 に示す。DCN からの notification 信号を受信したノード B, C は、hop 数を 1 とし、推定した DCN との距離情報を格納した notification 信号を送信する。この 2 つの信号を受信することができるノード D は、信号受信時に計測される信号強度から推定されたノード間の距離と、その信号パケット内に格納された距離情報を加算し、最も値が小さくなるものを自身と DCN との距離として決定し、hop 数を 2 として、notification 信号を送信する。hop 数が 3 以上となるノードにおいても同様の方法で、DCN との距離を推定するとともに、自身の notification 信号を送信する。

次々と notification 信号が送信される中で、各ノードは周囲のノードから送信される notification 信号を利用して、Long hop route と Short hop route の 2 種類の経路を確保する。Long hop route とは、隣接するノード間の通信距離が長い経路を意味し、パケットを中継する回数を減らすために用いられる。一方、Short hop route は隣接するノード間の通信距離が短い経路をさ意味し、ノード間の通信信頼性を高めるために用いられる。Martin Haenggi と Daniele Puccinelli は論文 [20] の中で様々な角度から Long hop route

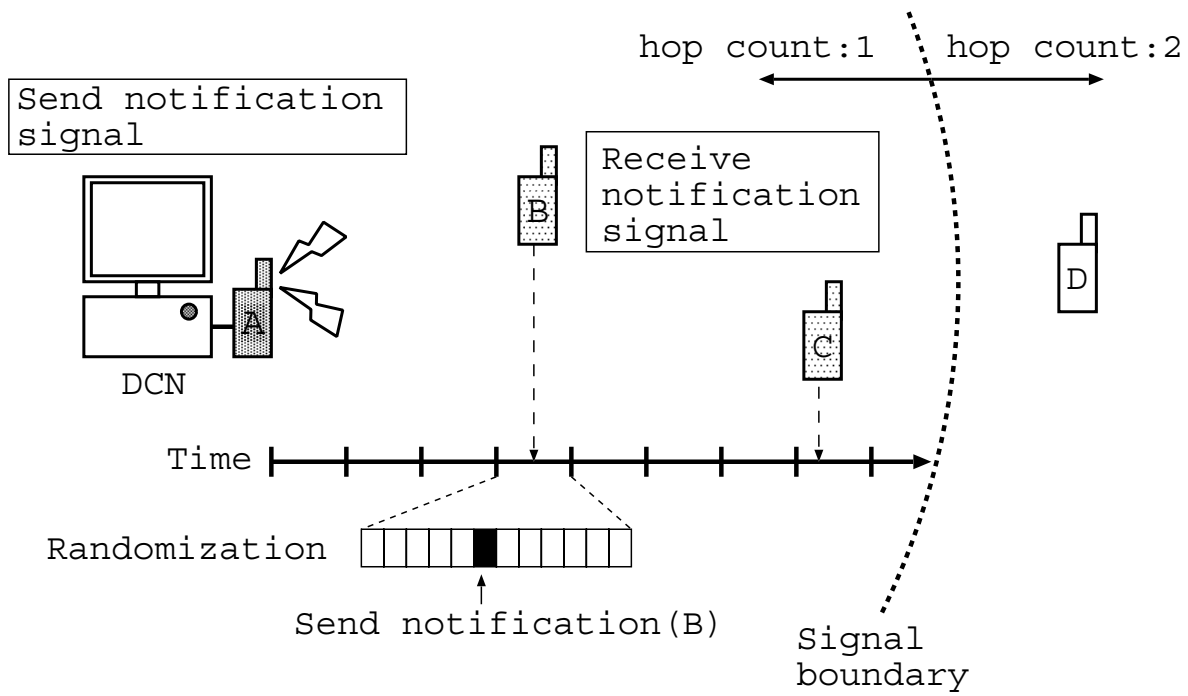


図 4.3: notification 信号送信タイミング

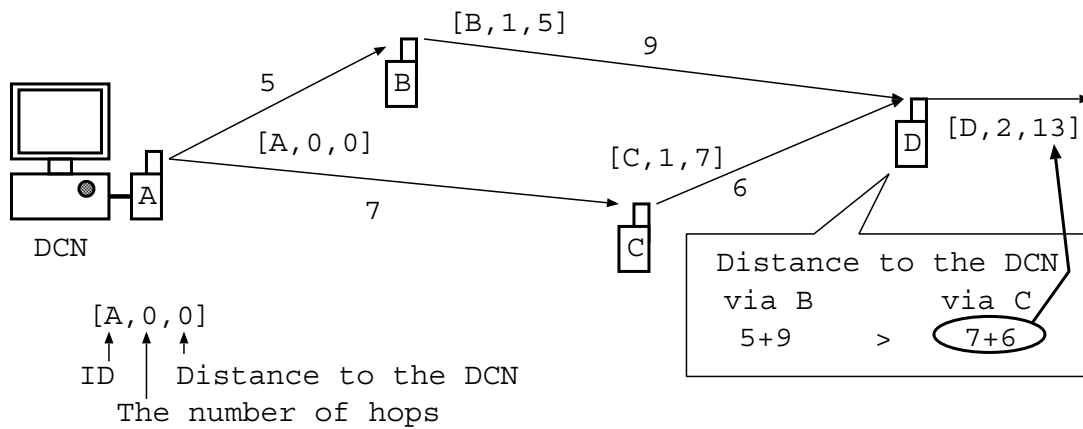


図 4.4: DCN との距離を推定する方法

の優位性を示しているが、本研究では、両方の経路を適宜使い分けることにより、遅延が少なく信頼性の高い通信の実現を目指す。そこで、ネットワークの中に多数存在する通信経路の中から、各ノードは表 4.1 に示すルールに従って経路の選択を行う。

このルールにより、自身のセンシング領域の中で、最も DCN に近いノードが Long hop route として選択 (図 4.5) されるとともに、自ノードより DCN との距離が近いノードの中から、通信時に高い信号強度が得られるノードが Short hop route として選択 (図 4.6) され、ルーティングテーブル内に格納される。これらの図では、DCN がシミュレーション空間の真ん中に設置されており、経路が DCN に向かって張られていることがわかる。ここでは、各ノードは Long hop route、Short hop route とともに 1 つだけ経路を表示しているが、実際には、複数の経路をルーティングテーブル内に確保することが可能である。

各ノードが notification 信号を送信し、その情報をもとに DCN への経路を確保することで、notification フェーズは終了する。各ノードが notification を行う際にはアルゴリズム 1,2 に従って動作する。communication フェーズでは、図 4.9 に示すように、Long hop route と Short hop route の 2 種類の経路を用いて各ノードが検出した情報を DCN へと転送を行う。

表 4.1: 経路選択ルール

Long hop route
・自ノードのセンシング領域にあるノードの中で DCN に最も近いノード
Short hop route
・自ノードより DCN との距離が近いノード かつ
・通信時に高い信号強度が得られるノード

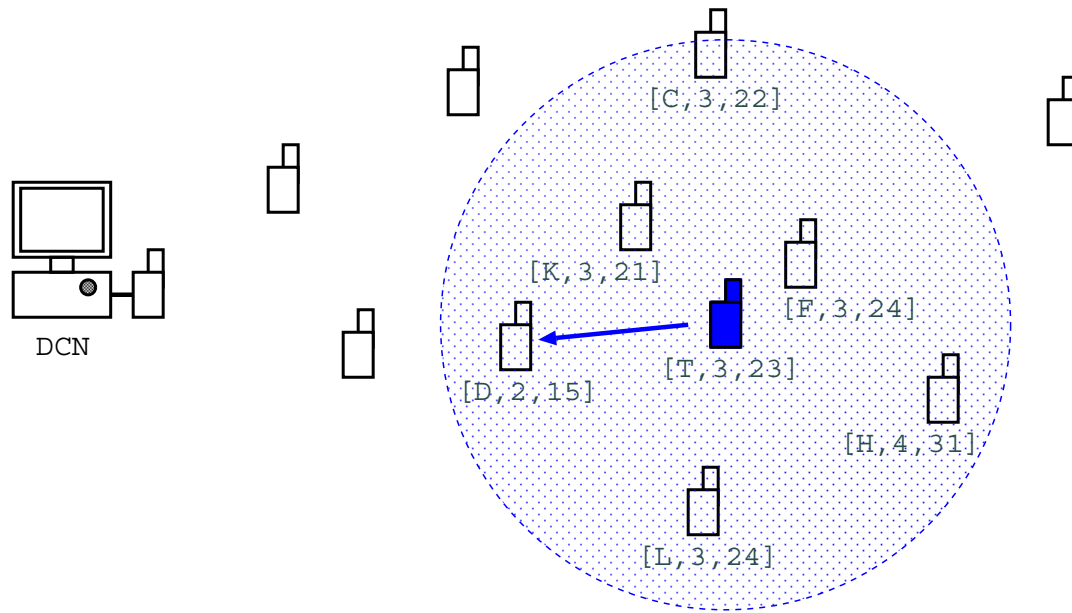


図 4.5: Long hop route の選択

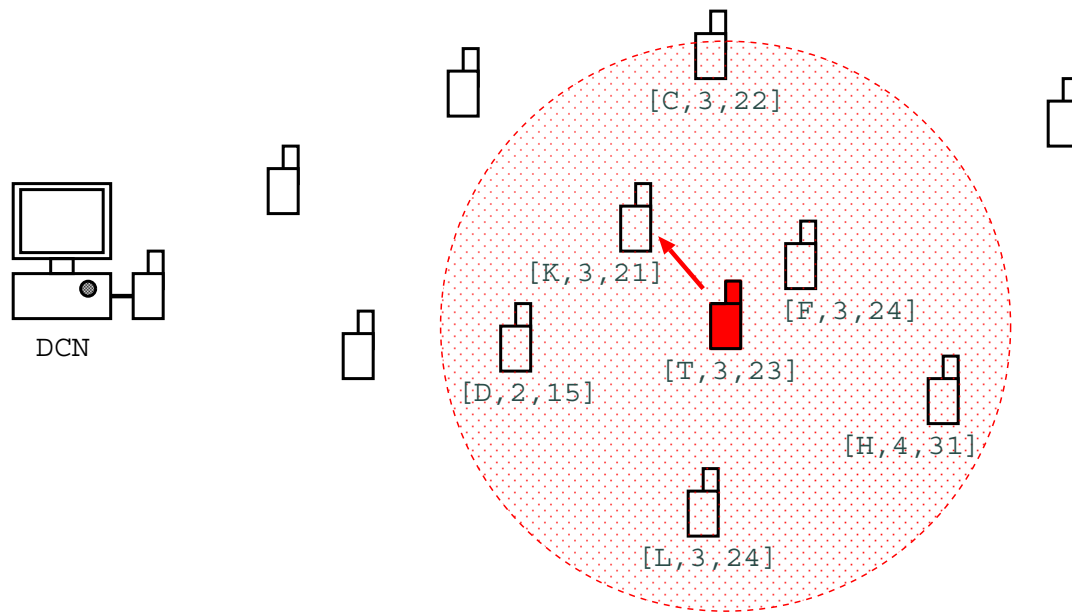


図 4.6: Short hop route の選択

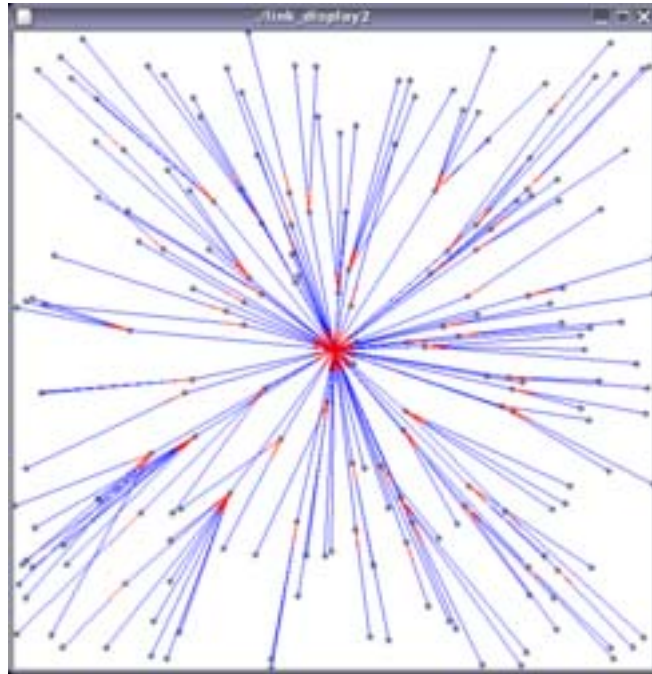


图 4.7: Long hop route

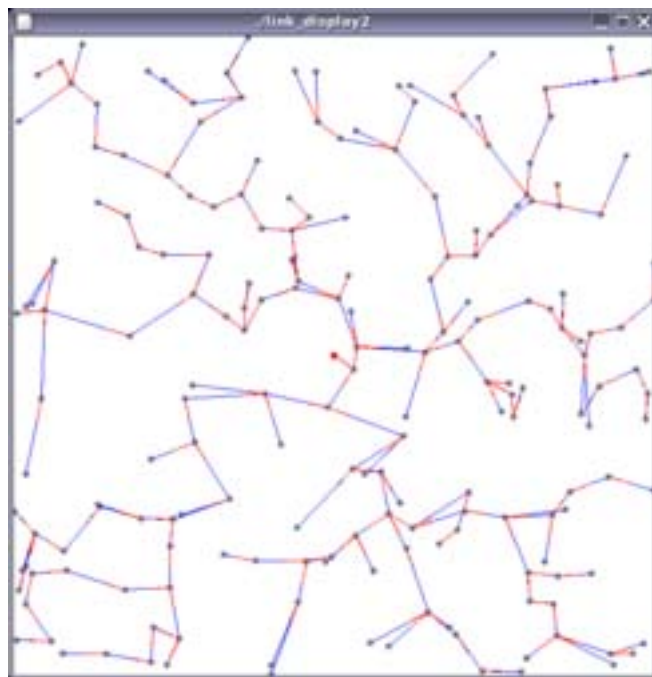


图 4.8: Short hop route



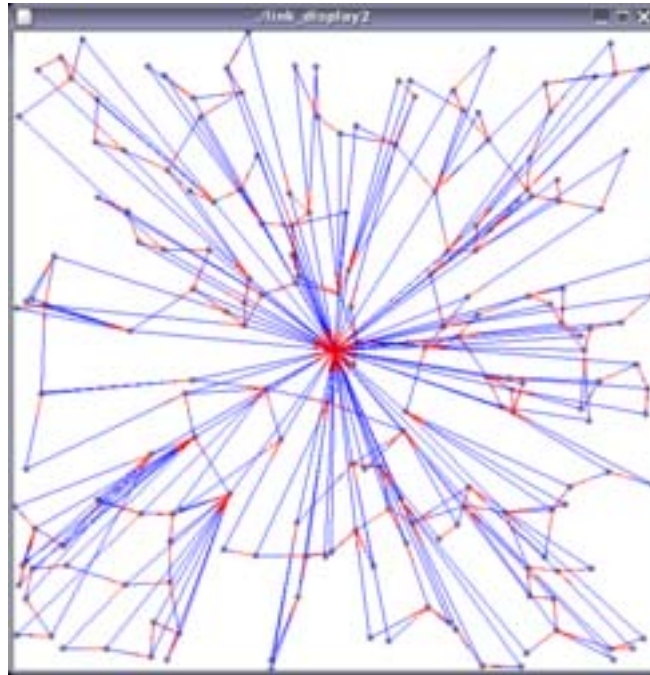


図 4.9: notification フェーズで得られる通信経路

---

**Algorithm 1** Sensor node operation in Notification phase

---

```

1:  $myhop \leftarrow$  ▷ Substitute large figure
2:  $mydist \leftarrow$ 
3: repeat
4:   Wait some signals from internal hard ware
5:   if Get signal then
6:     if Get notification signal then▷ Receive notification signal from neighbouring
       node
7:        $SETSTATUS(ID, hop, dist, estimated\_dist)$ 
8:     end if
9:     if Get timer signal then ▷ It's time to send my own notification signal
10:      Send my own notification signal
11:    end if
12:  end if
13: until Send my own notification signal

```

---

---

**Algorithm 2** Set status algorithm

---

```
1: procedure SETSTATUS(ID, hop, dist, estimated_dist)
2:   if myhop > hop + 1 then
3:     myhop  $\leftarrow$  hop + 1
4:   end if
5:   if mydist > dist + estimated_dist then
6:     mydist  $\leftarrow$  dist + estimated_dist
7:     time_slot  $\leftarrow$  notification_signal_sending_time * 30
8:     delay  $\leftarrow$  time_slot * estimated_dist + Random[0 : time_slot]
9:     Set send notification timer to be activated after delay seconds  $\triangleright$  timer set or
    update
10:  end if
11:  if dist is nearest to the DCN then
12:    Select as long hop route
13:  end if
14:  if dist is nearer than mydist and obtain high level signal strength then
15:    Select as short hop route
16:  end if
17: end procedure
```

---

## 4.2.2 COMMUNICATION フェイズ

communication フェーズでは各ノードが検出した情報を DCN に向けて hop-by-hop で送信を行う。notification フェーズでは、2種類の通信経路を確保したが、通常は通信遅延が小さくなるよう、中継する回数の少ない Long hop route が用いられる。しかし、電波強度は距離に比例して弱くなるため、通信が不安定になる場合が考えられる。そこで、Long hop route での通信が不可能な場合には、Short hop route に切替えて通信を行う。図 4.10 に、DCN へのデータパケット送信例を示す。ノード B が検出したデータを DCN に送信する際には、Long hop route であるノード F に対してデータパケットを送信する。ノードの内部では、mac 層として IEEE802.11 を想定し、データパケットを受信した際には、ACK 信号を返信する。ここで、複数回データパケットを送信したにもかかわらず、F からの ACK 信号を受信できない場合には、経路制御プロトコルが Short hop route であるノード C へと経路の変更を行う。Short hop route を用いることで、通信距離が短くなり、通信の信頼性が向上することになる。ノード C はノード B に対して ACK 信号を返信すると、自身が保持するルーティングテーブルの中から経路を選択し、DCN へとデータパケットを転送する。この転送動作を各ノードが行うことで、最終的に DCN にデータが到達することになる。なお、各ノードはアルゴリズム 3,4,5 に従って動作を行う。

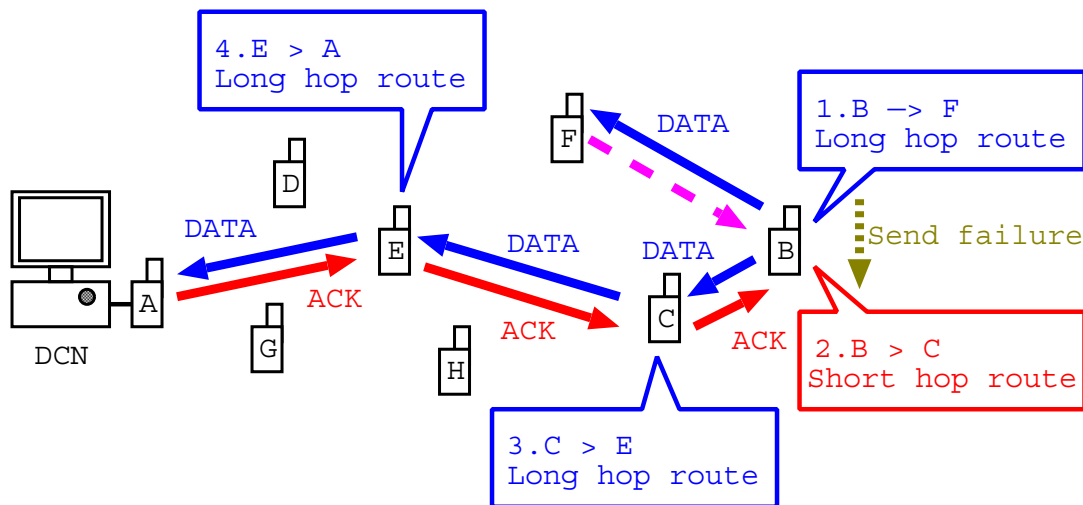


図 4.10: DCN へのデータパケットの送信

---

**Algorithm 3** sensor node operation in Communication phase

---

```
1: repeat
2:   Wait some signals from internal hard ware
3:   if Get signal then
4:     if Receive data packet then
5:       TRANSFERDATAPACKET(DataPacket)
6:     end if
7:   end if
8: until node dies or something
```

---

---

**Algorithm 4** Transfer data packet algorithm

---

```
1: procedure TRANSFERDATAPACKET(DataPacket)
2:   Send back ack signal
3:   Select next hop route from long hop route
4:   SENDDATAPACKET(DataPacket, LHroute)
5:   if Succeed at sending data packet then
6:     return
7:   else
8:     Select next hop route from short hop route
9:     SENDDATAPACKET(DataPacket, SHroute)
10:  end if
11: end procedure
```

---

---

**Algorithm 5** Send data packet algorithm

---

```
1: procedure SENDDATAPACKET(DataPacket, route)
2:    $i \leftarrow 32$ 
3:    $slottime \leftarrow 0.0002$ 
4:   while  $i \leq 1024$  do
5:      $backoff\_time \leftarrow \text{Random}[0 : slottime * i]$ 
6:      $i \leftarrow i * 2$ 
7:     Send packet after  $backoff\_time$ 
8:     if Get ack signal from route then
9:       return success
10:    end if
11:  end while
12:  return failure
13: end procedure
```

---

# 第5章 シミュレーションによるプロトコル性能評価

この章では、前章で提案したルーティングアルゴリズムについて、シミュレーションにより、その性能を評価する。現在、複数のネットワークシミュレータが開発され、実際にネットワークプロトコルのシミュレーションが行われているが、本研究ではシミュレータとして SENSE[6](SEnsor Network Simulator and Emulator) を用いて以降のシミュレーションを行う。センサネットワークは、その利用が想定される環境が多岐にわたるため、柔軟なプロトコルの設計が求められる。SENSE ではプログラムの拡張性、再利用性に優れており、プログラムをコンポーネント化することで、従来のプロトコルスタックにとらわれることなく自由度の高いプロトコル設計が可能となる。

## 5.1 シミュレーション条件

表 5.1 に、シミュレーション条件の概要を示す。一辺が 30m の 2 次元空間にランダムに配置されたノードは通信半径を 10m とし、パケット長が 15byte のデータパケットを 19.2kbps の通信速度で、DCN へと転送を行う。

表 5.1: シミュレーション条件

シミュレーション領域	30m × 30m
センサノードの数	49,99,199
データ収集ノードの数	1
ノードの初期配置	ランダム
ノードの移動	固定
通信半径	10m
通信速度	19.2kbps
パケット長	15byte
シミュレーション時間	800 秒

4 章で提案した経路制御プロトコルの性能を比較をするため、受信信号強度 (RSS : Received Signal Strength) を用いずに、ホップ数の情報のみで経路を選択した noRSS プロ

トコルについてもシミュレーションを行う。図 5.1 に noRSS プロトコルで得られる通信経路の例を示す。この図においても、RSS の場合と同様に、DCN はシミュレーション空間の中心に設置されている。noRSS がこれらの経路を形成するにあたっては、notification フェーズにおいて、DCN から送信される notification 信号を合図に各ノードはランダムなタイミングで、自身の notification 信号を送信する。また、各ノードは、隣接ノードから送信される notification 信号のなかから、最初に受信した 2 つの信号をもとに DCN との通信経路を決定する。

受信強度が利用できない noRSS では、Long hop route と Short hop route を区別することができないため、communication フェーズにおいては、どちらか一方の経路を利用し、その経路での通信が不可能となった場合には、もう一方の経路に切替えて通信を行う。

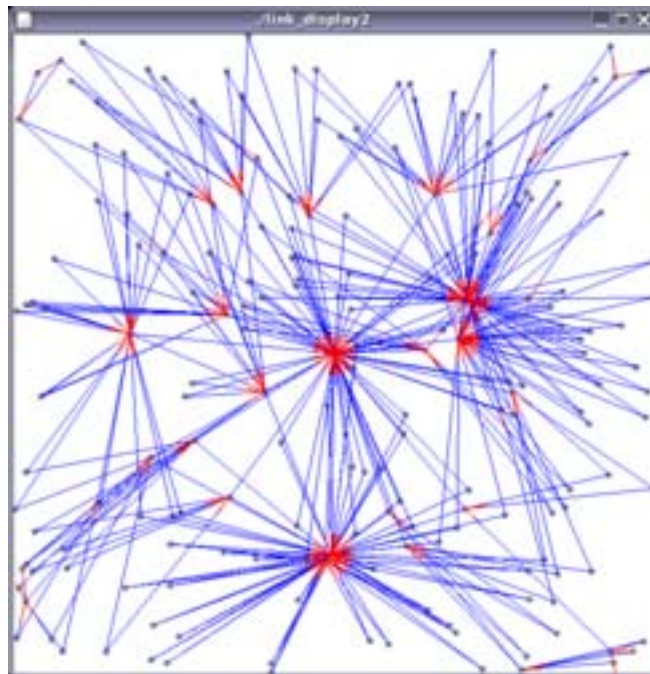


図 5.1: noRSS によって形成される通信経路

### 5.1.1 CBR 法によるパケット生成

各ノードは、Constant Bit Rate (CBR) 法に従い、データパケットを生成する。図 5.2 には、各ノードにおけるデータパケット生成タイミングを示す。各ノードは指定された  $T$  秒間のなかで、最初はランダムにパケットを生成するタイミングを選択する。その後は、 $T$  秒ごとにパケット生成を繰り返す。ここでは、パケット生成間隔  $T$  を 8 秒、12 秒、16 秒と設定し、それぞれについてシミュレーションを行った。

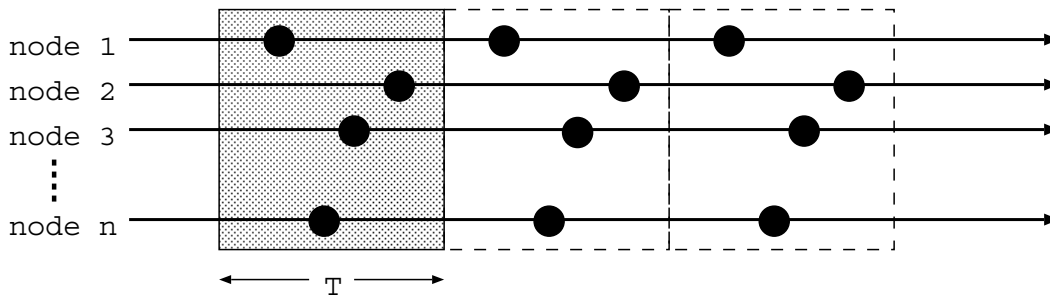


図 5.2: CBR 法によるデータパケット生成

### 5.1.2 電波強度モデル

シミュレーションを行う際には、Free Space と呼ばれる電波モデルを使用する。

$$S_s = T_x \left( \frac{c}{4\pi f d} \right)^2 \quad (5.1)$$

ここで、 $T_x$  はトランスミッターからの電波送信出力、 $c$  は光速、 $f$  は信号周波数、 $d$  はトランスミッターとの距離を意味している。このモデルでは式 (5.1) に示されているとおり、距離の 2 乗に比例して電波強度が減衰する。このモデルを利用するにあたっては、各ノードから送信される信号出力  $T_x$  を同一のものとし、ノードの通信可能な距離が 10m となるように、受信閾値を設定する。各ノードにおいては、受信した信号強度  $S_s$  がこの閾値を越えた場合にのみ、信号が受信できたものとみなす。

### 5.1.3 通信距離半減エリアの設定

信号強度が変化する要因として、シミュレーション内に電波信号の通信距離が半減するエリアを設定する。このエリアは一辺が 10m の正方形で構成され、シミュレーション開始時には、ランダムな位置に存在する。このエリアに電波信号が触れた時点で、信号強度が 3 割減少することにより、この地点以降における通信可能な距離が半減する。このエリアは秒速 0.5m でランダムに設定された地点まで移動し、そこで 20 秒間停止する。その後はまた同様に移動と停止を繰り返す。

このようなテストケースについて、合計 5 回、DCN とノード、通信距離半減エリアの位置を変更してシミュレーションを行う。

## 5.2 シミュレーション結果と考察

ここでは、前節で設定した条件をもとに、実行したシミュレーションの結果を示す。DCN が notification 信号を送信することでシミュレーションが開始され、notification フェーズ

により、各ノードは通信経路を確保する。実際にこのフェーズは10秒程度で終了し、その後各ノードはデータパケットをCBR法にもとづきDCNへと送信する。

### 5.2.1 データパケット到達率

図5.3には、データパケットのDCNへの到達率を示している。ここでは、シミュレーション環境に配置するノードの数を49,99,199に設定し、cbr法でのパケット生成間隔Tを8,12,16秒と変化させている。この図から、信号受信強度を用いるRSSを用いることで、ほぼすべてのデータパケットをDCNへと到達可能なことがわかる。また、受信信号強度を使用しないnoRSSの場合は、隣接するノードとの距離を確認することができないため、信号強度半減エリア内に入った場合は、高い信号強度が得られるshort hop routeを選択することができず、信号到達率の減少につながっている。しかし、RSSにおいてもノード数の増加やデータパケットの送信間隔が短くなることで、ネットワーク全体のトラフィックが増加してくると、すべての通信が到達できるとはいえなくなる。特に最も条件の厳しい、ノード数:200, データパケット生成間隔T=8の場合では、5回の試行うち、一度しか100%の到達率を示すことはなかった。また特に5回目の試行での結果が悪化したのは、DCNの配置による影響が大きい。図5.6には、2回のシミュレーションを行った際に形成された通信経路を示す。図5.6(a)では、DCNが右下に設置されているため、DCNへ送信されるデータパケットは、主に左上から、右下のほうへと流れることになる。一方、図5.6(b)では、DCNが中央下に設置されているため、全ての方向からデータパケットが転送されることになる。そのため、データパケットの衝突が発生しやすくなり、DCNが正しくデータパケットが読み取れなくなることでパケットの到達率が低下することになる。

また、ノード数が49個の場合には、RSSにおいて、他の全ての試行の際にはデータの到達率が100%を示しているにもかかわらず、2回目の試行(Trial number:2)の場合にのみ、到達率の低下がみられる。これは、ノードの密度が低いために、Long hop routeとShort hop routeの選択基準に適合するノードが周辺に存在せず、経路を確保することができなかったためである。図5.7と図5.8を比較すると、RSS, noRSSともに、このシミュレーションにおいて各ノードは経路を2つだけ確保することになっているが、RSSのほうが確保された経路が少ないことがわかる。この傾向はすべての試行の際にみることができ、Trial numberが2の場合に影響が生じた。当然ではあるが、各ノードは他のノードのセンシング領域の中に設置するとともに、適切な経路を提供してくれるノードが周辺に存在しなければならない。

### 5.2.2 パケット平均遅延時間

図5.4には、パケット平均遅延時間を示す。ここでは、DCNに到達したパケットのみを対象とし、それらすべてのパケットの遅延時間の平均値を求めている。図5.4から、RSS, noRSSともに、ほぼ数秒以内でDCNへのデータパケット送信が完了していること



がわかるが、ノード数が 199 個の場合は RSS のほうが遅延時間が短いと言える。しかし、パケット生成間隔  $T$  が 8 秒の場合は、試行番号の違いによっては RSS における平均遅延時間が大幅に悪化した。これは、データトラフィック量が増加することで、信号の衝突が発生している場合にも、RSS では通信経路を Long hop route から Short hop route に変更を行う性質を持つためである。このため、通信ホップ数が増加し、それがパケット遅延を増大させる原因となっている。

### 5.2.3 パケット最悪遅延時間

図 5.5 には、パケット最悪遅延時間を示す。最悪遅延時間とは、パケットが生成された後に、それが DCN に到達するまでに必要とした時間が最も長くなった値を表す。この時間は、DCN に到達したパケットのみを対象とし、DCN に到達しなかったものについては無視する。図 5.5 から、ノード数が 49 個の場合においては、RSS を用いることで、ノードの配置に問題のある場合 (Trial number:2) をのぞいて、最悪でも数秒程度のうちには DCN にパケットが到達していることがわかる。しかし、ノードの数の増加やパケット生成間隔  $T$  が短くなることにより、最悪遅延時間は 250 秒近くにまで増加することになる。これを解決する方法としては、ネットワーク容量に対するパケット通信量を抑えるか、各ノードがパケット生成時にタイムスタンプを付加することで、各ノードが送信すべきデータパケットの中から、優先的に時間の経過しているパケットを送信する、などの方法が考えられる。

### 5.2.4 シミュレーションのまとめ

ここでは、受信信号強度を用いた RSS と、それを用いることなくホップ数の情報のみで経路を形成した noRSS について、cbr 法にもとづいた DCN へのパケット送信シミュレーションを行った。その際には、電波信号強度の変動要因として、通信距離が半減するエリアをシミュレーション空間内に設定した。このエリアについては、現実世界においてこのような電波強度変化が起こるとは限らず、その妥当性については全く不明であるが、もし、電波強度が変動した場合には、強い信号強度が得られる経路に切替えて通信を行うという、RSS の優位性を示すことはできた。

これは、すべてのネットワークについて言えることではあるが、ネットワークにおける通信トラフィックがある点を越えると、ネットワークの通信性能は急激に悪化する。そのため、ネットワークを構築する際には当然、ネットワークの要件に適合したシステムを構築する必要があると言える。

なお本来ならば、他の一般的なルーティングアルゴリズムとの性能を比較したいところではあるが、SENSE に実装されているデモ用ルーティングアルゴリズムの実装が不完全な点と、ns-2 などの他のシミュレータに比べてマイナーであるため、他の研究者から

SENSE用のプログラムが提供されていない点から、SENSE上での性能比較を行うことは難しい状況にある。

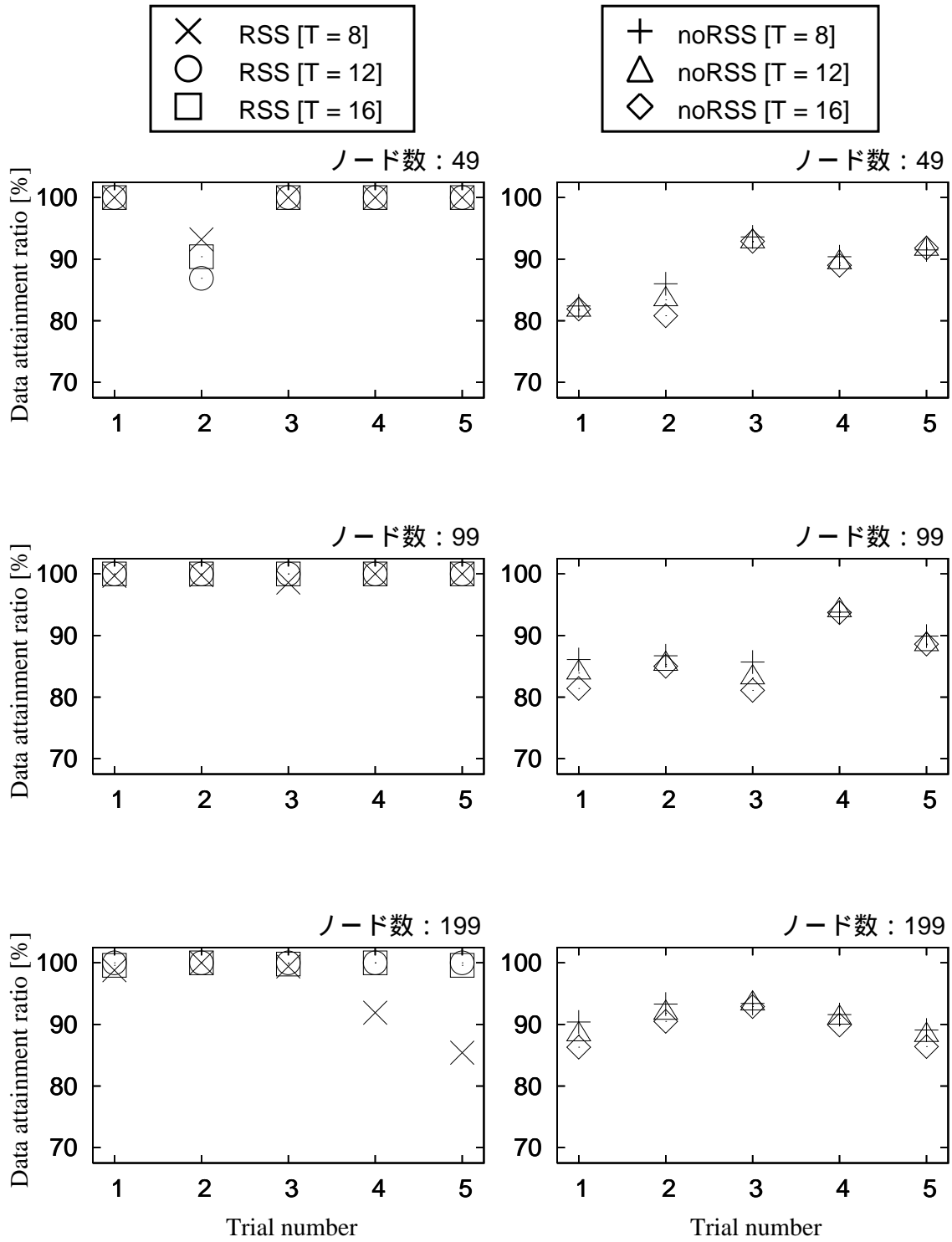


図 5.3: データパケット到達率

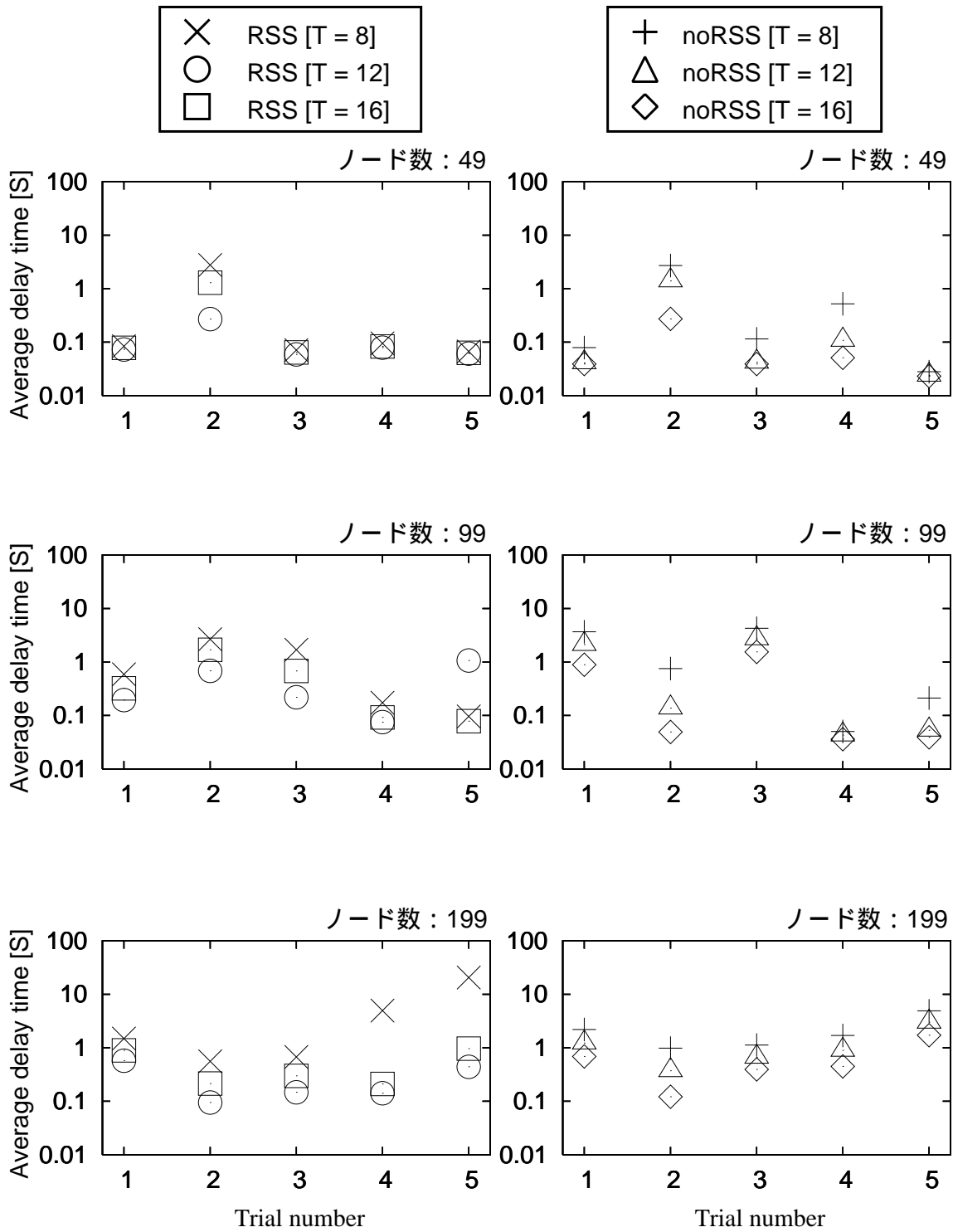


図 5.4: 平均遅延時間

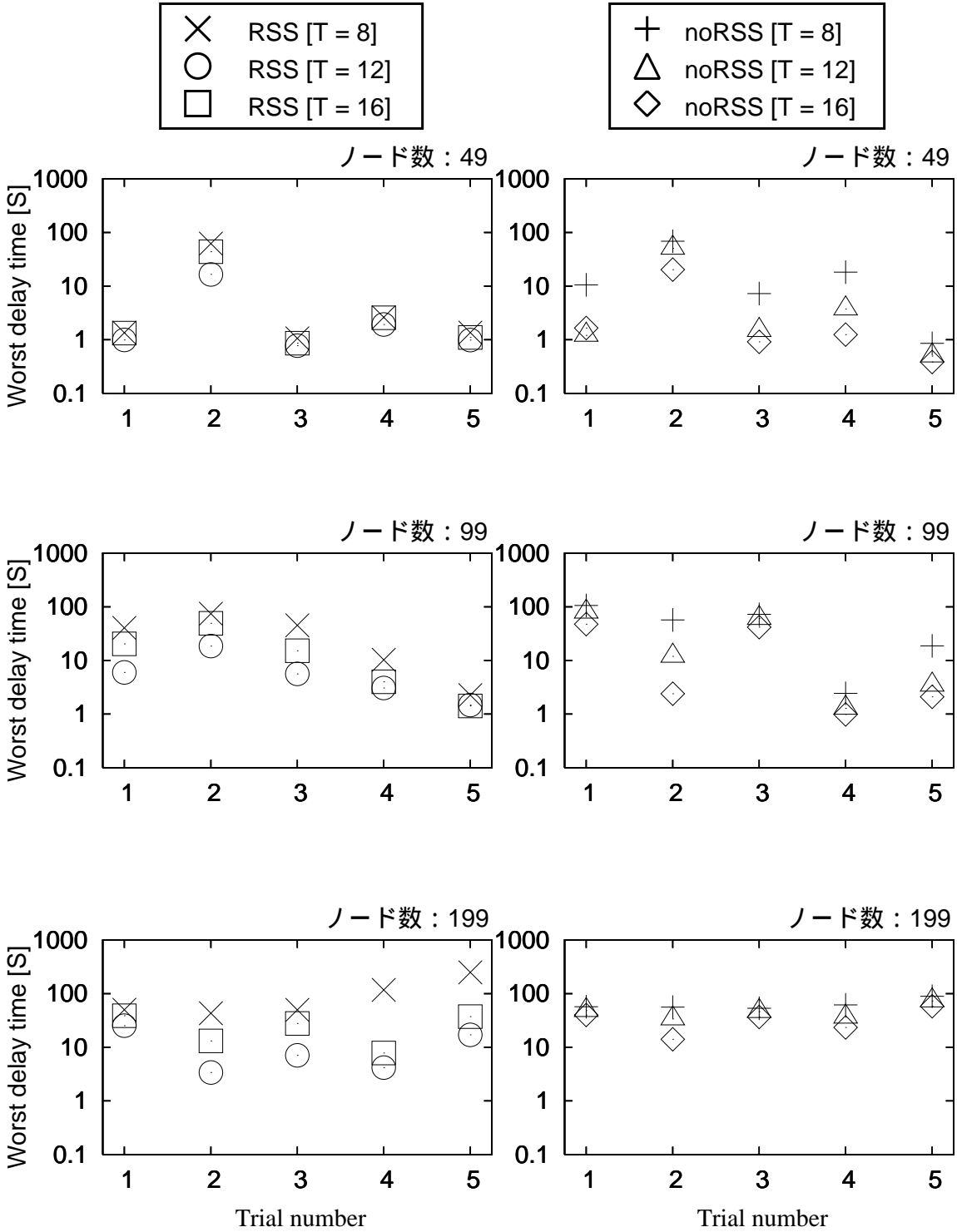
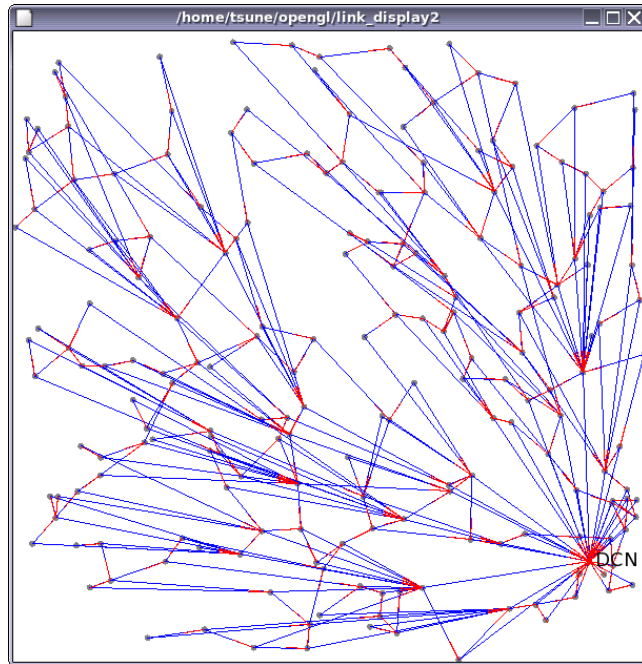
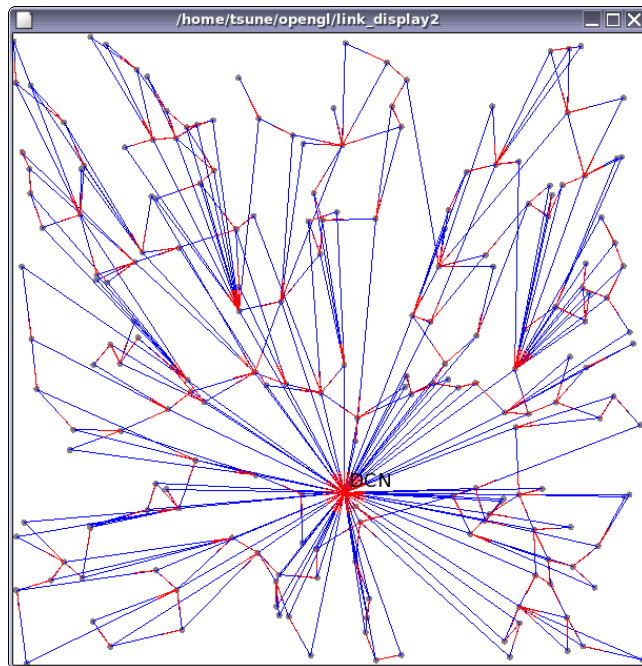


図 5.5: 最悪遅延時間

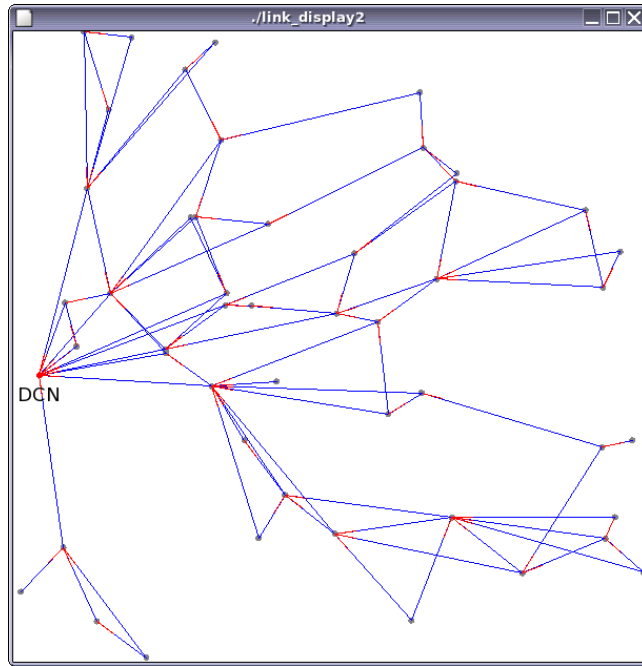


(a) RSS, ノード数:199, Trial number:2

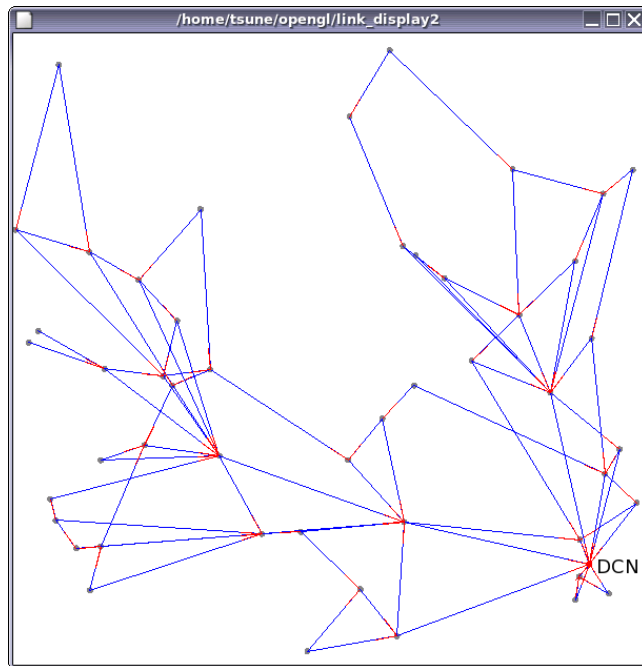


(b) RSS, ノード数:199, Trial number:5

図 5.6: RSS で形成された通信経路

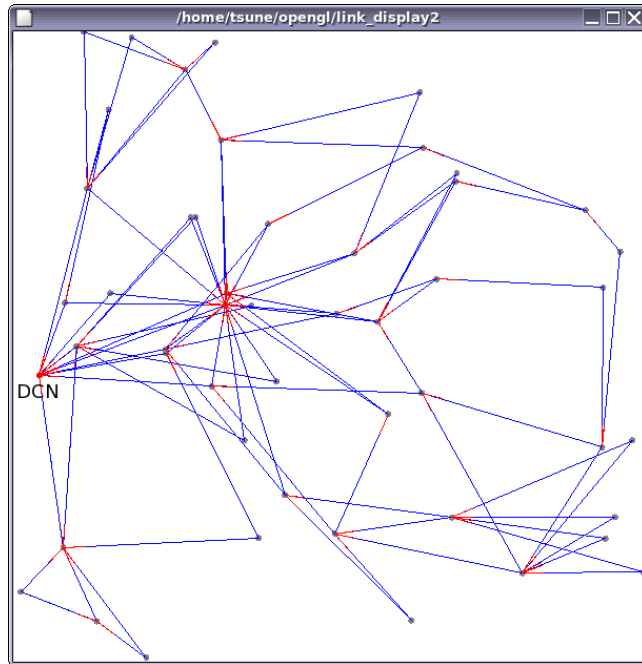


(a) RSS, ノード数:49, Trial number:1

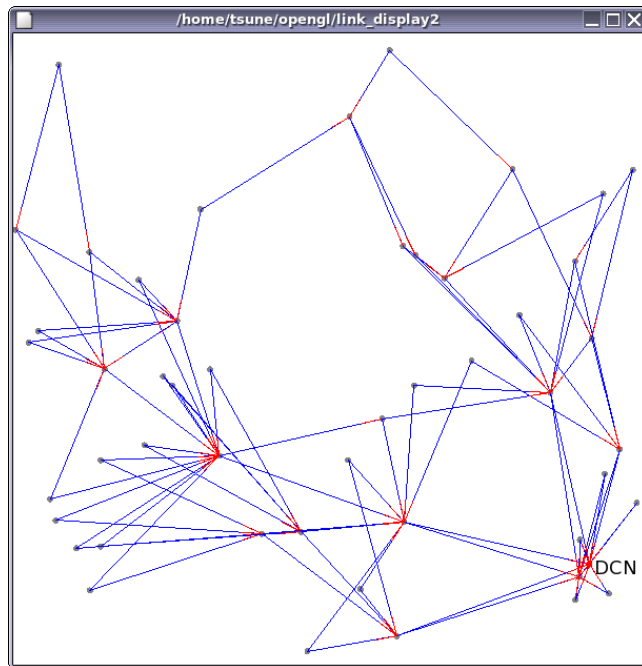


(b) RSS, ノード数:49, Trial number:2

図 5.7: RSS で形成された通信経路



(a) noRSS, ノード数:49, Trial number:1



(b) noRSS, ノード数:49, Trial number:2

図 5.8: noRSS で形成された通信経路



# 第6章 結論

## 6.1 本研究のまとめ

本研究では、受信信号強度 (RSS : Received Signal Strength) を用いて低い信号遅延と高い信頼性を持ったセンサネットワーク用ルーティングアルゴリズムの構築を行った。本稿では、センサネットワークの概要について表し、一般的な無線ネットワーク形態の1つであるアドホックネットワークとの違いについて比較を行った。そこから、センサネットワークの特徴として、データパケットは観察者と呼ばれるデータ収集ノードにのみ集められる点や、アドホックセンサネットワークと比較して、ネットワークを構成するノードの性能が低い点や、高い密度でノードが配置される点が異なることが示された。そこで、これらの特徴をもとに、ルーティングアルゴリズムを notification フェーズと communication フェーズの2つのフェーズに分割を行った。そのなかで、notification フェーズでは、各ノードが自分自身の存在を notification 信号をもちいて告知するとともに、これらの信号の内容と、その受信信号強度をもとに Long hop route と Short hop route の2種類の経路を確立する。また、communication フェーズでは、通常、Long hop route を用いたデータパケット転送を行う。このルートは隣接ノード間の通信距離が長く、少ないホップ数での通信が可能であるため、信号遅延を抑えることができる。もしここで、通信が不可能となった場合には、高い信号強度が得られる Short hop route に切替えることで、信号強度の変動にも対応することができる。この2つの経路を使い分けることによって、低い信号遅延と高い信頼性を持つ通信が可能となった。

## 6.2 今後の課題

今後の課題として、以下の項目があげられる。

- 実センサノードへのプロトコル実装
- ノードが移動する場合への対応

本研究で提案した受信信号強度を用いたセンサネットワーク用ルーティングアルゴリズムを実センサノードに搭載し、その動作検証を行う必要がある。その際には、電波強度の距離による変動がどのようなモデルで表現することができるのか、電波の反射、屈折、減衰などの影響は環境によってどの程度変動するのか、また、電波強度の時系列変化につい

て観察実験を行う必要がある。これらのデータをもとにアルゴリズムの妥当性の検討や実環境に適合したアルゴリズムへの改良が期待される。さらに、ルーティングアルゴリズムをノードが移動する場合について対応できるよう改良することが今後の課題として挙げられる。これにより、アルゴリズムの実用可能性が向上するとともに、電波強度の変動に対しても柔軟に対応することが期待される。

## 参考文献

- [1] Mark Weiser, The computer for the twenty-first century, *Scientific American* 265(3), pp94–104, September 1991.
- [2] B. Brumitt, B. Meyers, J. Krumm, A. Kern, and S. Shafer, Easy Living : Technologies for Intelligent Environments, *Proc. of the 2nd International Symposium on Handheld and Ubiquitous Computing (HUC2000)*, pp12–29 2000.
- [3] S. Tilak, N. B. Abu-Ghazaleh, and W. Heinzelman, A taxonomy of wireless micro-sensor network models, *ACM SIGMOBILE Mobile Computing and Communications Review* 6(2), pp28–36, 2002.
- [4] R. Ogier, F. Templin, and M. Lewis, Topology Dissemination Based on Reverse-Path Forwarding (TBRPF), RFC 3684, 2004.
- [5] T. Clausen, and P. Jacquet, Optimized Link State Routing Protocol (OLSR), RFC 3626, 2003.
- [6] SENSE: Sensor Network Simulator and Emulator  
<http://www.cs.rpi.edu/~cheng3/sense/>
- [7] The Network Simulator - ns-2  
<http://www.isi.edu/nsnam/ns/>
- [8] S-NODE Wireless Sensor Networks Node  
<http://ymatic.co.jp/>
- [9] J. Elson, and D. Estrin, Fine-grained Network Time Synchronization Using Reference Broadcasts, *ACM SenSys*, 2003.
- [10] R. Lyenger, K. Kar, and S. Banerjee, Low-coordination Topologies For Redundancy in Sensor Networks, *ACM MobiHoc*, pp332–342, 2005.
- [11] E. Biagioni, and S.H. Chen, A Reliability Layer for Ad-Hoc Wireless Sensor Network Routing, *Proc. HICSS Track 9*, p90300, 2004.

- [12] M. Varshney, and R. Bagrodia, Detailed Models for Sensor Network Simulations and their Impact on Network Performance, Proc. ACM MSWIM, pp70–77, 2004.
- [13] C. Perkins, E. Belding-Royer, and S. Das, Ad hoc On-Demand Distance Vector (AODV) Routing, RFC 3261, Internet RFC/STD/FYI/BCP Archives, 2003.
- [14] D.B.Johnson, D.A. Maltz, and Y-C. Hu, The Dynamic Source Routing Protocol for Mobile Ad hoc Networks (DSR), draft-ietf-manet-dsr-10.txt, July 2004.
- [15] G. Finn, Routing and Addressing Problems in Large Metropolitan-Scale Internetworks, Tech. Rep. ISI/RR-87-180, Information Sciences Institute, 1987.
- [16] B. Karp, and H. T. Kung, GPSR : Greedy Perimeter Stateless Routing for Wireless Networks, Proc. of ACM/IEEE MOBICOM'00, 2000.
- [17] C. Intanagonwiwat, R. Govindan, D. Estrin, J. Heidemann, and F. Silva, Directed diffusion for wireless sensor networking, ACM/IEEE Transactions on Networking 11(1), pp2–16, February 2002.
- [18] 安藤繁, 田村陽介, 戸辺義人, 南正輝, センサネットワーク技術, 東京電機大学出版局, 2005.
- [19] Klaus Finkenzeller, RFID ハンドブック - 第2版 -, 日刊工業新聞社, 2004.
- [20] M.Haenggi, and D. Puccinelli, Routing in Ad Hoc Networks: A Case for Long Hops, Communications Magazine, IEEE Volume 43, Issue 10, pp93–101, October 2005.