

Title	ワイヤレスセンサーネットワークにおけるデータ伝送のQoSに関する研究
Author(s)	金, 永光
Citation	
Issue Date	2007-03
Type	Thesis or Dissertation
Text version	author
URL	http://hdl.handle.net/10119/3584
Rights	
Description	Supervisor:丹 康雄, 情報科学研究科, 修士

修 士 論 文

**ワイヤレスセンサーネットワークにおける
データ伝送の QoS に関する研究**

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

金 永光

2007 年 3 月

修 士 論 文

ワイヤレスセンサーネットワークにおける データ伝送のQoSに関する研究

指導教官 丹 康雄 助教授

審査委員主査 丹 康雄 助教授
審査委員 篠田 陽一 教授
審査委員 日比野 靖 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

510031 金 永光

提出年月: 2007 年 2 月

概要

ワイヤレスセンサーネットワークのデータ伝送 QoS に関する研究は主に各リンクやパスの latency 値の縮小か reliability 値を上昇させる研究がされてきた。だがワイヤレスセンサーネットワークでは単一のパラメータを考慮するのは不十分であるのは明らかである。本研究では、このような問題に対して、ワイヤレスセンサーネットワークのデータ伝送において、もっとも重要視される両パラメータ：latency と reliability を取り上げ、両パラメータをともに考慮してネットワークの各リンクおよびパスに対して latency と reliability をトレードオフする計算法を提案する。

目次

第1章	はじめに	1
1.1	背景	1
1.2	目的	1
1.3	本論文の構成	1
第2章	ワイヤレスセンサーネットワークにおけるデータ伝送の QoS とは	3
2.1	はじめに	3
2.2	ワイヤレスセンサーネットワークとは	3
2.3	アドホックネットワークとの違い	4
2.4	QoS とは	5
2.5	ワイヤレスセンサーネットワークでの QoS とは	6
2.6	まとめ	7
第3章	L-R 式ルーティングアルゴリズムの提案	8
3.1	はじめに	8
3.2	パラメータ	8
3.2.1	Latency	8
3.2.2	Reliability	8
3.2.3	時間 T_r	9
3.2.4	時間 $T_{infor-change}$	9
3.3	プロセス	9
3.3.1	ノードの配置	9
3.3.2	Sink ノードへの隣接ノード情報の集め	12
3.3.3	コストの計算	16
3.3.4	コストによるルーティング	19
3.4	ルーティングにおける主な問題への対応	23
3.4.1	ノードの脱落問題への対応	23
3.4.2	ノードの密集問題への対応	32
3.5	時間 $T_{infor-change}$ の決め方	32
3.6	まとめ	33

第 4 章 評価	34
4.1 はじめに	34
4.2 L-R 計算法に対する検証	34
4.3 オーバーヘッドに対する検証	41
4.4 まとめ	43
第 5 章 討論	44
5.1 はじめに	44
5.2 on-demand ルーティングアルゴリズムへの適用	44
5.3 まとめ	46
第 6 章 結論	47
6.1 まとめ	47
6.2 今後の課題	47
謝辞	48
付 録 A 用語一覧	49
参考文献	50

目 次

2.1	ワイヤレスセンサーネットワークの構成	4
2.2	ワイヤレスセンサーネットワークとアドホックネットワークの関係	5
3.1	ルーティング全体図	10
3.2	ルーティングの方向	11
3.3	ルーティングの方向-その2	12
3.4	ネットワークのトポロジー	13
3.5	隣接情報を送るためのツリー	14
3.6	伝送ツリーのアルゴリズム	15
3.7	ダイクストラ法 (Dijkstra 法) のアルゴリズム	17
3.8	最適シングルパスルーティング	22
3.9	マルチルーティングパスの探索アルゴリズム	28
3.10	最適マルチパスルーティング	31
4.1	ノード 15 個に対しての Latency の検証	35
4.2	ノード 15 個に対しての Reliability の検証	36
4.3	ノード 15 個の L-R 計算法の検証	37
4.4	node50 個の各最適パス	39
4.5	ノード 50-150 の latency のトレードオフ検証	40
4.6	ノード 50-150 の reliability のトレードオフ検証	40
4.7	主な ad hoc network のプロトコルの比較その 1	41
4.8	主な ad hoc network のプロトコルの比較その 2	42

表 目 次

3.1	各ノードの隣接情報	14
3.2	ノード 1 (sink) が持つ最適パスのルーティングテーブル	20
3.3	ノード 2 が持つ最適パスのルーティングテーブル	20
3.4	ノード 3 が持つ最適パスのルーティングテーブル	20
3.5	ノード 7 が持つ最適パスのルーティングテーブル	21
3.6	ノード 11 が持つ最適パスのルーティングテーブル	21
3.7	マルチルーティングパスの探索アルゴリズム	26
3.8	ノード 1 (sink) が持つ最適パスのルーティングテーブル	27
3.9	ノード 2 が持つ最適パスのルーティングテーブル	29
3.10	ノード 3 が持つ最適パスのルーティングテーブル	29
3.11	ノード 1 (sink) が持つ最適パスのルーティングテーブル	29
3.12	ノード 2 が持つ最適パスのルーティングテーブル	30
3.13	ノード 3 が持つ最適パスのルーティングテーブル	30
4.1	実験データ	38
4.2	実験データの比較	38
4.3	主な Proactive プロトコルとのオーバーヘッドの比較	42

第1章 はじめに

1.1 背景

現在、社会のユビキタス化の進行につれ、さまざまなモノがつながるが、それらの設定を状況に応じてその都度手動で変更するのは困難である。特に、在宅患者の健康管理、オフィスや自宅の電気機器のコントロール、農作物栽培における施設管理、交通量モニタリング、自然災害の監視など多くの分野での活用において、ワイヤレスセンサーネットワークの実現化が求められるようになっている。ワイヤレスセンサーネットワークが実現すると、利用者やモノの状況を自動的に認識し各種設定を人を介することなく変更することができ、さらには利用者の状況に応じたさまざまなサービスをリアルタイムに提供することも可能になる [1][2]。

1.2 目的

ワイヤレスネットワークではデータの伝送中のデータの損失率がほかのネットワークと比べると大きい。高いデータ伝送の reliability（信頼性）を保つためにデータの再伝送がひとつ方法であるが、頻繁の再伝送処理は大きい latency（遅延）をもたらす。そこでいかに効率よく sink に伝送できるようにするかはワイヤレスセンサーネットワークのデータ伝送のひとつ大きな課題になっている。

ワイヤレスセンサーネットワークのデータ伝送 QoS に関する研究は主に各リンクやパスの latency 値の縮小か reliability 値を上昇させる研究がされてきた [3][4]。だがワイヤレスセンサーネットワークでは単一のパラメータを考慮するのは不十分であるのは明らかである。本研究では、このような問題に対して、ワイヤレスセンサーネットワークのデータ伝送において、もっとも重要視される両パラメータ：latency と reliability を取り上げ、両パラメータをともに考慮してネットワークの各リンクおよびパスに対して latency と reliability をトレードオフする計算法を提案するのがわれわれの研究の目的である。

1.3 本論文の構成

本論文は以下のような構成になっている。

- **第一章** 本研究の背景と目的および論文の構成について述べる。

- **第二章** ワイヤレスセンサーネットワークやワイヤレスセンサーネットワークにおけるデータ伝送の QoS に関する研究の現状について述べる。
- **第三章** われわれが提案した L-R 計算法について説明する。
- **第四章** われわれが提案した L-R 計算法の検証および考察について説明する。
- **第五章** われわれが提案した L-R 計算法の拡張や応用について分析する。
- **第六章** 本研究の今後の課題と本研究の全体のまとめについて述べる。

第2章 ワイヤレスセンサーネットワーク におけるデータ伝送のQoSとは

2.1 はじめに

この章では簡単にワイヤレスセンサーネットワークやワイヤレスセンサーネットワークのデータ伝送のQoSについて説明する。

2.2 ワイヤレスセンサーネットワークとは

ユビキタスコンピュータはあらゆるモノの中にマイクロチップ、センサーを入れることで、コンピュータ自身を現実世界に進出させ、実世界の「状況」を認識し活用させることに主眼を置いている。従来のコンピュータ・ネットワークはインターネットに代表されるように「仮想空間」へのアクセスをするが、ユビキタスコンピューティングは現実の世界を反映したモデルと捉え、その状況を環境中に配置されたコンピュータ組込み機器群が自動認識し、サポートしていく。センサネットワークは種々のセンサを備えた小型かつ省電力の無線端末により実現され、単にセンサ端末を分散配置させるだけでシステムを構築することが可能で、ネットワークの拡張が有線ネットワークと比べて非常に簡単に行えるという利点がある [5]。一般的にワイヤレスセンサーネットワークはノードの機能により、センシングノードと sink ノードに分ける。

- センシングノードとは

センシングノードはセンサー端末であり、一般的に簡単にノードと呼ぶ。各ノードは森林や建物などのセンシングしたい対象に対して、温度や湿度などの物理情報（ここで生データと呼ぶ。）を取得し、各中間ノードを通して sink ノードへ伝送する役割を果たす。

- sink ノードとは

管理端末である。ワイヤレスセンサーネットワークではBS（base station）とも呼ぶ。ノードがセンシングして取得した情報を最終的に sink ノードに伝送される。sink ノードはそれらの情報をさまざまな目的に応じて分析して利用する。sink ノードは一般的にインターネットネットワークとつながっているのコンピューターまたは比較的データの処理能力や通信能力が高い携帯PCやPDAなどの端末を指す。センサー

ネットワークはノードがセンサー情報を取得することで、多数のセンサー情報を収集し処理する sink ノードで様々なサービスを提供することができる。

ノード自身が移動する（人がセンサノードを持って動く等）ことも可能であるが、一般的にはノードは固定されているセンサーネットワークを指す。

ネットワーク化された複数のセンサーを仮想的な1個のセンサーとして情報を観察者に提供するものがセンサーネットワークの特徴である。複数のセンサーが単にネットワークに接続されただけならば、感知されたデータがそのまま観察者に届くことになる。その場合、複数のセンサーから届いた情報を観察者自身で処理することになるが、センサーネットワークでは観察者が必要とする情報だけが最終的に届く。センサーネットワークではネットワーク上にあるセンサー同士が協調しあって、ネットワーク内部でデータの内容に基づいて冗長性をなくしたり、融合したりする。従来のインターネットのように基本的に終端端末間でデータは送受信され、途中のルータがデータを融合するといったことはない [5]。一般的にワイヤレスセンサーネットワークの構成は図 2.1 のようになっている。

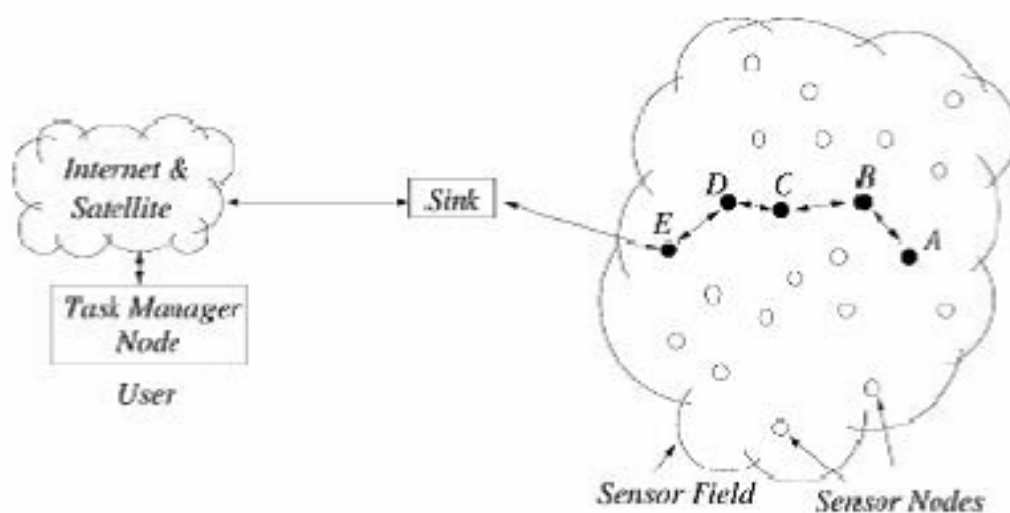


図 2.1: ワイヤレスセンサーネットワークの構成

2.3 アドホックネットワークとの違い

ワイヤレスネットワークのひとつであるアドホックネットワーク (ad hoc network) は主な特徴がある。

- ネットワークトポロジーが頻繁に変化し、(無線通信・モバイル環境が前提) 固定的なネットワークインフラが存在しない

- 集中管理機構がなくすべてのノードは対等である
- マルチホップネットワークである

アドホックネットワークに対して、センサーネットワークは主に以下のような違いがある [6][7]。

- (1) センシングして取得したデータのみを扱う
- (2) モバイル環境を前提としない
- (3) 末端のセンサー同士の通信よりも末端のセンサーと基地局に相当するノードあるいはアクセスポイントの間の階層型通信が中心
- (4) IP による通信を前提としない

これらのワイヤレスセンサーネットワークの特徴のためにワイヤレスセンサーネットワークのルーティング設計はほかのネットワークルーティングとは大きく異なる。また同じワイヤレスセンサーネットワークでもアプリケーションによってルーティングの設計も異なる。ワイヤレスセンサーネットワークはアドホックネットワークとは共通する部分はあるが、違う部分も多いのであるその関係は主に以下の図で表すことができる。



図 2.2: ワイヤレスセンサーネットワークとアドホックネットワークの関係

2.4 QoS とは

QoS とは Quality of Service である。ネットワークでネットワークあるいはネットワークサービスの提供者がユーザに対して提供するサービスのパフォーマンスのレベルを規定する基準である。QoS は一定のものでなく、ネットワークのトポロジーやアプリケーション

ンによって、評価する基準や対象が異なる。QoSは主にフロアごとやリンクごと、またはノードごとに対して優先度制御するなどによって実現する。

従来のインターネットワークでは、そこでひとつのフロアに対して、主に次の四つのパラメータにより決められる。

- Reliability

データ伝送の reliability は主にパケット損失率 (Packet-loss rate) つまり単位時間内に損失したパケット数によって決まる。パケットの損失は、伝送系の障害や、ルータのバッファオーバーフローで生じる。回線速度が異なるデータリンクが接続された状態で起こりやすい。

- Latency

データが送信側から受信側に到達するのにかかる時間。伝送路の状況に依存する伝送 latency と、中継機器の処理能力に依存する中継 latency の合計で表される。

- Bandwidth

単位時間あたりに転送できる最大データ量。ネットワーク内での伝送時間を決定する基本的な要因であり、平均レート及びピークレートの両方で表現される。

- Jitter

データの到達遅延時間の標準偏差。パケットの到着時刻のばらつき具合を表すもので、映像・音声などのストリーム品質を大きく左右する。

フロアの伝送中にデータに対する、データの優先度や各ノードの受送信の帯域幅への制御によって、ネットワークがよりよい品質のサービス (E-mail サービス、映像や音声の受送信サービスなど) を提供できるようにする手法である [8][9]。

2.5 ワイヤレスセンサーネットワークでの QoS とは

ワイヤレスセンサーネットワークではインターネットワークと同じような QoS 問題が存在する。だが、ワイヤレスセンサーネットワークの特徴により、QoS の内容や注目する重点はすこしインターネットワークの QoS と違うのである。簡単に説明すると、QoS を評価する主なパラメータが違うのである。インターネットワークと比べて、主に reliability と latency を重点において研究がすすんでいるのである。つまりいかに効率よくデータを早くそして損失が少なく伝送するのがワイヤレスセンサーネットワークでの QoS 問題だと考えられている。

ワイヤレスセンサーネットワークはワイヤレスネットワークの一種である。従来のワイヤレスネットワークとは違い、大量のセンサーノードにより構成され、あるセンシングするターゲットにセンサーノードを配置し、ターゲットの温度、湿度、ノイズ、ライトコンディションなどをセンシングし、それらのデータを sink ノードへ送る。またワイヤレスセ

ンサーネットワークの QoS サポートはアプリケーションの要求によって大きく異なる。ひとつ QoS サポートソリューションですべてのアプリケーションを満足することはできない。

今までのワイヤレスセンサーネットワークにおける QoS サポートに関する研究は大きく以下のように 2 つに分けることができる。

a. ノード間のリンクの最適化による QoS サポート

ノード間のリンクの最適化による QoS サポートを中心とするルーティングプロトコルやルーティングアルゴリズムは主にノード間のリンクの最適化、つまりノード間の伝送品質を上げることによって、ネットワークに QoS サポートを提供するのはこのタイプのコンセプトである。たとえば、代表的に Sequential Assignment Routing (SAR) と SPEED のルーティングプロトコルをあげて説明する。SAR はリンクのデータ伝送のエネルギーをリンクの重みとし、また各パケットに優先度をつけることによってパスを選び、ルーティングするアルゴリズムである [10]。また SPEED 各ノードは各隣接ノードとの間のリンクでのデータの伝送スピードを比較しながら、パスを選び、ルーティングするプロトコルである [11]。このタイプの QoS サポートの中でも、特にデータ伝送の reliability を中心において設計されたルーティングプロトコルはいくつある、たとえば ReInRorM や IASN [12][13]。これらのアルゴリズムは主にリンクでのパケットの伝送は設定とおりに保つためにパスの選択を行いながら、ルーティングするアルゴリズムである。

b. ノードの数の最適化による QoS サポート

ある特定なセンサーネットワークに対して、そのネットワークのアプリケーションに応じて、センサーネットワークのノードの配置において、ノード数や位置を最適化することにより、ネットワーク全体に QoS サポートを提供するルーティングプロトコルである [14][15]。

(a) のノード間のリンクの最適化による QoS サポートの [12][13] のようなルーティングプロトコルではデータ伝送の reliability を設計しているが実際ではセンサーネットワークのルーティングアルゴリズムの設計において、ほかのファクターも入れて、考えねばならない。そこで、われわれはこれらのルーティングプロトコルに対してデータ伝送の reliability を考慮した上でファクターのひとつであるリンクの latency を加えて、両パラメータを考慮し、両者のトレードオフされた最適パスを選び出しルーティングするのはわれわれのルーティングアルゴリズムのコンセプトである。

2.6 まとめ

ワイヤレスセンサーネットワークはアドホックネットワークと共通する部分はあるが、違う部分も多い。そのため、両者のアルゴリズムの設計も異なる。その両者の特徴から特にセンサーネットワークに対して、アプリケーションなどに応じてアルゴリズムの設計を行うべきである。

第3章 L-R 式ルーティングアルゴリズムの提案

3.1 はじめに

この章ではわれわれが提案するワイヤレスセンサーネットワークのデータ伝送におけるリンクの latency と reliability の両パラメータを対象にした latency と reliability のトレードオフする L-R 計算法について詳しく説明する。

3.2 パラメータ

ここで、われわれの計算法やアルゴリズムで使われる主なパラメータについて簡単に説明する。

3.2.1 Latency

一般的にネットワークでデータがあるソースノードから送信されて、目的地ノードまでの時間をリンクの latency と呼ぶ。またデータがソースノードから送信され、目的地ノードにつき、そして目的地ノードから返すデータがソースノードが受信するまで時間つまりソースノードにとって待ち時間は latency と呼ぶ場合もある。われわれの提案では latency は後者を指す。例を挙げると、あるデータがノード A からノード B へ送信されてノード B が受信するまでの時間は T とし、ノード B がそのデータを処理する時間は T_α とする。またノード A とノード B 間のリンクの latency は $Latency_{A-B}$ とする。この場合、 $Latency_{A-B} = Latency_{B-A} = T + T_\alpha$ となる。これから $Latency_{A-B}$ は L_{A-B} と略す。

3.2.2 Reliability

ワイヤレスセンサーネットワークではデータの伝送中には環境などにより、データの損失は激しい。そのような信号やデータの損失をチャネルエラー (channel error) と呼ぶ。ここで e と略す。とすると、リンクの $reliability = 1 - e$ とする。そして、あるノード A とノード B 間のリンクの reliability を R_{A-B} と記す。

3.2.3 時間 T_r

T_r はリンクの latency であるが、ここでの T_r はあるノード A が自分と物理的に半径の距離まで離れているノード B 間の latency つまり L_{A-B} である。つまり時間 T_r はすべてのノードの最大待ち時間である。

3.2.4 時間 $T_{infor-change}$

われわれのアルゴリズムではある一定の時間にネットワークの全体のすべてのノードが自分の隣接ノードの情報を交換して、その情報を sink に送らなければならないので、その交換周期の時間を $T_{infor-change}$ を記す。

3.3 プロセス

われわれのルーティングアルゴリズムは主に以下のプロセスによって実現される。

- プロセス 1. ノードの配置：
センサーノードを配置する。ワイヤレスネットワーク特にワイヤレスセンサーネットワークに対して、対象とする環境はノードのデータ伝送に大きく影響するので、ここで、われわれのアルゴリズムの対象とするセンサーネットワークのセンサーの配置環境について説明する。
- プロセス 2. 隣接情報の集め
ノードの配置のプロセスが終わると、各ノードは sink から infor-req メッセージを受け取って、自分のすべての隣接ノードの情報を sink に集まる。
- プロセス 3. リンクコストの計算
sink が集めてきた各リンクの情報に基づいて各リンクのコストや最適パスを計算しだす。
- プロセス 4. コストによってのルーティング
各最適パスによるルーティングを行って各ノードの生データをゲットする。

全体のアルゴリズムは以下の図 3.1 になっている。これから各プロセスについて詳しく説明する。

3.3.1 ノードの配置

ワイヤレスセンサーネットワークではノードの配置は主に二つの方法がある。

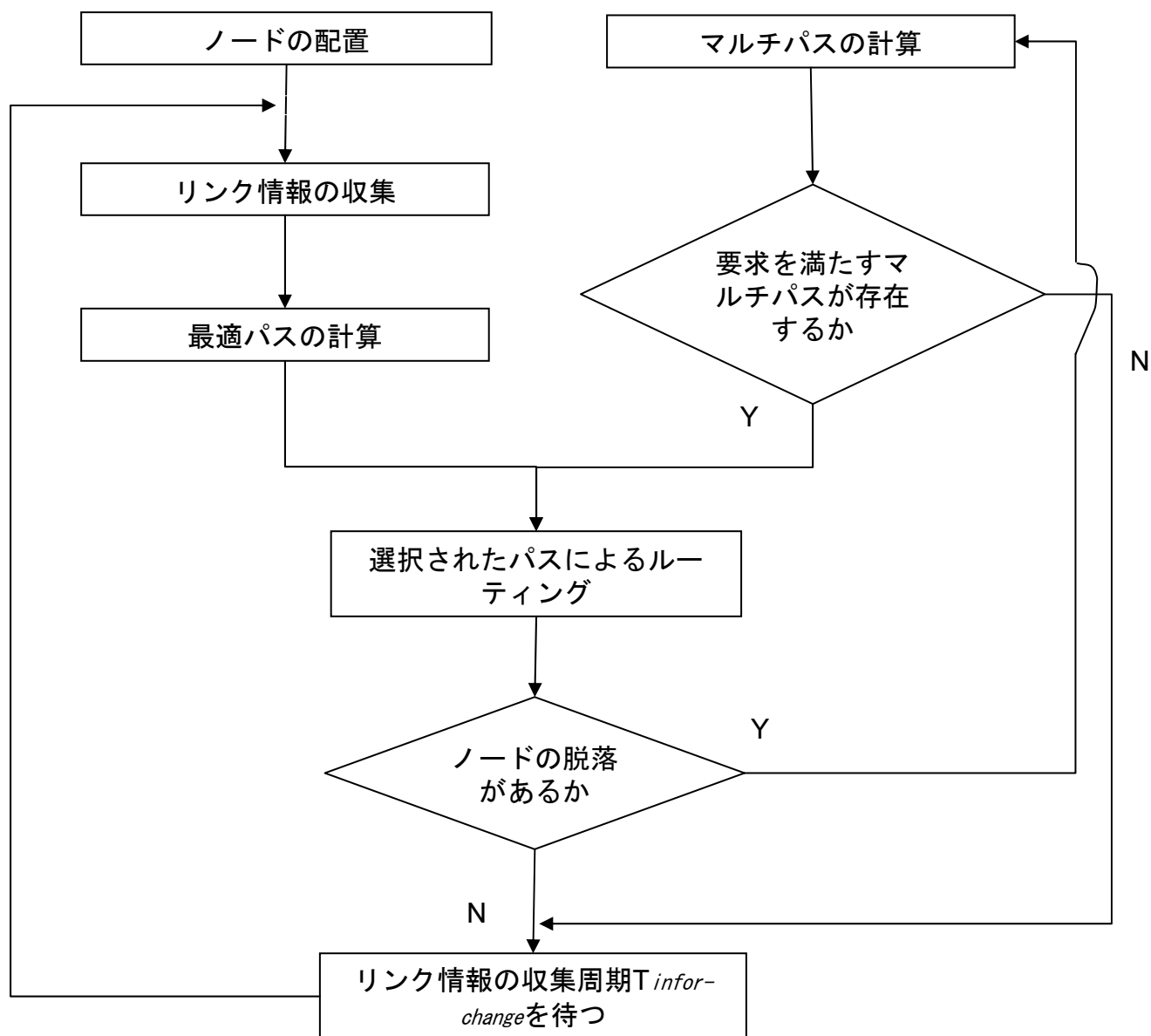


図 3.1: 全体のルーティングアルゴリズムこのようになっている。

- (1) は飛行機などの交通工具によって空中から観察目標にばら撒くという方法。たとえば、森林や山などの環境観察や探測に使う方法である。この方法では空中から目標にばら撒くことによつてのノードやノード間の初期状態が把握しにくい点や自然環境にデータ伝送が大きく影響されるなど不安定な要素が多い。
- (2) は人工によつての設置である。この方法は主に建物の温度や湿度などのデータの観測や観察に使われる方法である。この方法は実際によく使われている方法である。

本研究では主に以下のようなノード配置環境と想定する。これらは必要条件ではないが、われわれのアルゴリズムにはより適切であると考えられる。

- ネットワーク特にワイヤレスネットワークでは、ネットワークのトポロジーは各ノードに対して、隣接ノードが6個であるのは最適であると検証によつてすでに証明されているので[16]、一般的ワイヤレスネットワークのトポロジーを各ノードに対して、4個から8個の隣接ノードを持つネットワークトポロジーを対象とするので、われわれもそのようなネットワークトポロジーを持つネットワークを対象とする。
- 人工によつてセンサーノードが配置される。
- 自然環境に激しく影響されないところ（建物など）、つまり頻繁にネットワーク全体の情報アップグレードしなくてすむところ。
- エネルギーの効率を高めるために、なるべくセンサーノードが隣接ノードの信号が届く半径のところに配置する。
- ルーティングアルゴリズムの方向性から見ると主に二つのパイプがある[17][18]。(1) sink-to-node 型と (2) node-to-sink 型。sink-to-node 型はデータのルーティングの

	PSFQ (Sep. 2002)	RMST (May 2003)	ESRT (June 2003)	CODA (Nov. 2003)	GARUDA (May 2005)	SenTCP (Mar. 2005)	Reliable Transport Protocol Library
<i>Congestion Control</i>	No	No	Passive	Yes	No	Yes	Yes
<i>Congestion Detection</i>	-	-	Buffer size	Buffer size & channel condition	-	Buffer size & pkts arrival rate	Buffer condition
<i>Reliability</i>	Yes	Yes	Yes	No	Yes	No	Yes
<i>EtE/HbH</i>	HbH	HbH	EtE	-	HbH	-	HbH
<i>ACK/NACK</i>	NACK	NACK	-	-	NACK	-	ACK, NACK
<i>Seq num</i>	Yes	Yes	-	-	Yes	-	Yes
<i>Direction</i>	Sink to sensor	Sensor to sink	Sensor to sink	Sensor to sink	Sink to sensor	Sensor to sink	Sink to sensor
<i>Energy</i>	-	-	Fair	Good	Yes	Good	-

図 3.2: 各のルーティングアルゴリズムの方向

方向は sink から node へ向かつて通信を行う。つまり sink が自分がほしいデータを

各ノードからとってくるものである。(2)の node-to-sink 型のルーティングは(1)と逆で、各ノードから sink に向かって自分のデータを送る。われわれのルーティングアルゴリズムは前者の(1)の sink-to-node 型である。われわれの対象とするネットワークはメッシュ型のネットワークでもある。つまりネットワークでは sink 以外の各ノードに対しては自分のロケーションは認識できないのであるので、各ノードは単なる隣接リンクから最短距離であるリンクに沿っていくと sink から多く離れるようになる恐れがあるからである。図 3.3 にその様子を表す。

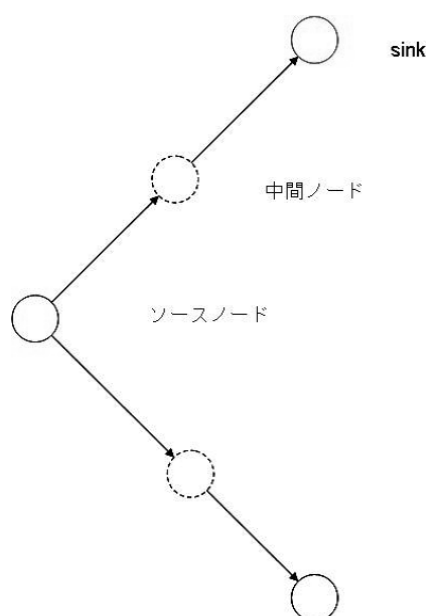


図 3.3: 真ん中にあるのノード（実線）はソースノードで右上のノードは sink とすると、ソースノードからが単なる最短距離のリンク（斜め下）の方向に沿ってルーティングすると上にある sink からより離れることになる。

3.3.2 Sink ノードへの隣接ノード情報の集め

ネットワークのセンサーノードの配置が終わると各ノードは隣接ノードの情報を集めて最終的に sink ノードへ集まる。そのステップは以下ようになる。

- ステップ 1. まず各ノードが自分初期状況をセッティングする。sink は自分の *hop-count* を 0 に、*father* を 1 にする、また各ノードは自分の *hop-count* を MAX に、*father* を 0 にする。
- ステップ 2. 各ノードは自分の *father* から *infor-req* メッセージを受け取る。そして自分の *hop-count* を $\text{hop-count}_{\text{father}} + 1$ にし、*father* を 1 にする。ここで *father*

はこのノードの状態を表す、*father* が1であれば、ノードはすでに伝送ツリーに入っていて *hop-count* は決められた状態であることを表す。また *father* が0であれば、このノードはまだ伝送ツリーに入っていない、つまり、*hop-count* が決められていない状態を指す。

- ステップ3. 自分の *father* 以外のノードに hello メッセージを broadcast する。もし隣接ノードがこの hello メッセージを受信すると自分の ID をこのメッセージにつけてメッセージを返す。各ノードは隣接ノードから返したメッセージから latency と reliability が確定することができる。
- ステップ4. 自分の *father* に自分と隣接ノード間のリンクの latency と reliability 情報を送る。
- ステップ5. 自分の *father* 値を1にする。この値は一度1に変更されると次のルーティングプロセスまでに変更されないことである。そして、自分の隣接のノードに infor-req メッセージを broadcast する。
- ステップ6. 隣接ノード情報は NULL であるまで以上の手順の繰り返し。

以上のステップで各ノードは次々と中間ノードの伝送して、最終的に sink に隣接ノード間の latency と reliability 情報を sink に送ることができる。配置されたセンサーネットワークは以下の図 3.4 のようなトポロジーを持つセンサーネットワークであるとする。各ノード

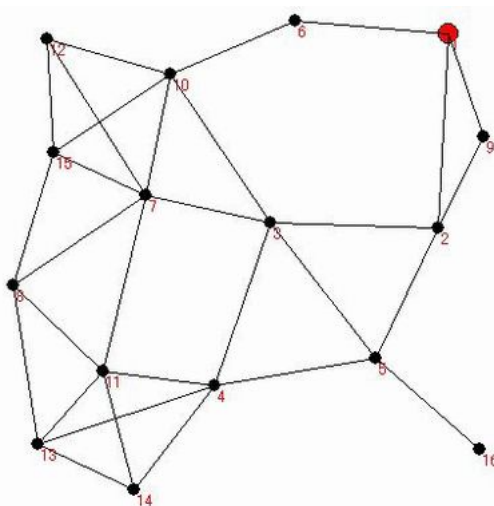


図 3.4: センサーネットワークのトポロジーである。ここで右上にあるノード 1 は sink である。各のノード下の番号は各ノードの ID である。

ドが収集した自分の隣接ノードの情報により、表 3.1 のようなテーブルが生成される。

ID	NeighborID	L_{11-j}	R_{11-j}
11	4	L_{11-4}	R_{11-4}
	7	L_{11-7}	R_{11-7}
	8	L_{11-8}	R_{11-8}
	13	L_{11-13}	R_{11-13}
	14	L_{11-14}	R_{11-14}

表 3.1: ノード 11 の隣接ノード情報である。 L_{11-j} はノード 11 と各隣接ノード j 間のリンクの latency である、同様に R_{11-j} はノード 11 と各隣接ノード j 間のリンクの reliability である。

そして、図 3.4 のネットワークに対して上のアルゴリズムによって生成された隣接情報を送るためのツリーは下の図 3.5 のようになる。伝送ツリーのアルゴリズムについては図

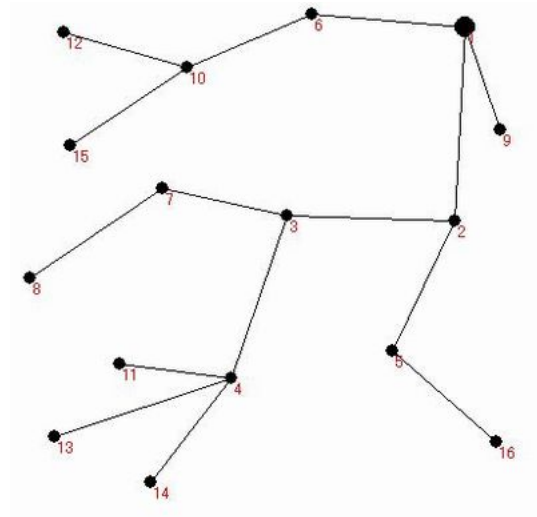


図 3.5: 隣接情報を送るためのツリーである。各のノードはひとつだけの father ノードを持つことはわかる。このツリーを沿って各ノードが自分の隣接ノード間のリンクの latency と reliability 情報を sink に送る。

3.6 を参照すること。このステップが終了すると、ネットワークにあるすべてのリンクの情報が sink が把握しているのと等しいと考えられる。これらの情報を元に次のプロセスで各リンクのコストを計算する。

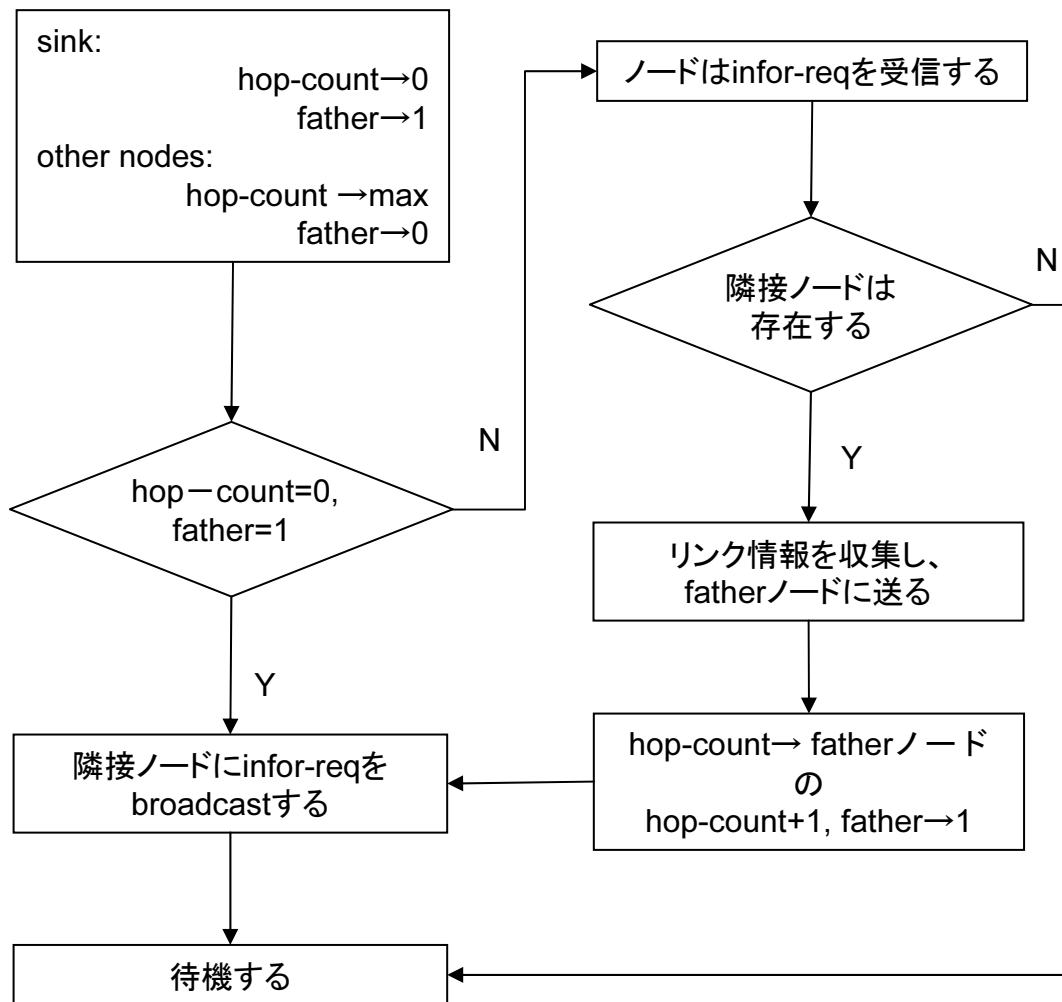


図 3.6: sink を含む各ノードの隣接ノード情報を sink へ集まるアルゴリズムである。

3.3.3 コストの計算

ここからはリンクコストの計算方法の説明である。われわれはワイヤレスセンサーネットワークにおけるデータ伝送に大きく影響するパラメータ Latency-Reliability をともに考慮し、両パラメータをトレードオフする計算方法— L-R 計算法の提案について詳しく説明する。これは上の各プロセスによって、sink に集めてきた各ノードの隣接ノード間の latency と reliability 情報を用いて、各ノード対のリンクのコストを計算するプロセスである。

ダイクストラ法 (Dijkstra 法)

このリンクコストの計算法は最短路パスの計算法のひとつであるダイクストラ法 (Dijkstra 法) を用いる。ここで、ダイクストラ法 (Dijkstra 法) について簡単に紹介する。

ダイクストラ法 (Dijkstra 法) のアルゴリズムは、出発点から到達可能なノードをまでの出発点からの距離を求める操作を、出発点から円形に範囲が広がる形で行っていき、最短経路を探索する方法である。

具体的には、頂点集合 V を 2 つの部分集合 VP と $VT (=V-VP)$ に分ける。両集合を分ける辺—カットを入れ、集合 VT の中から、最短パスになる経路を見つけ、集合 $VP=V$ まで広げることによって、最短パスを見つかるアルゴリズムである。より正確には以下のように書ける [19][20]。

(1) スタート点 s を選び、

$VP := \{s\}; \text{distance}[s] := 0; \{\text{path}[s] := 0; \}$
 $\text{distance}[v] := w(e); \{\text{path}[v] := e; \} (e = (s, v) \in E)$
 $\text{distance}[v] := \infty; \{\text{path}[v] := -1; \} ((s, v) \notin E)$
とする。

(2) $VP \neq V$ である限り、(a) , (b) を繰り返す。

(a) $V-VP$ の頂点の中で、 distance の値の最小な頂点 w を求める。

(b) $VP := VP \cup \{w\}$ とする。さらに w を始点とする各辺 $e = (w, v)$ に対して、
 $\text{distance}[v] > \text{distance}[w] + w(e)$
ならば、
 $\text{distance}[v] := \text{distance}[w] + w(e) \{; \text{path}[v] := e \}$
とする。

ここで例を挙げて、上のダイクストラ法 (Dijkstra 法) のアルゴリズムを説明する。図 3.7 のように頂点 A からほかの各頂点までへの最短パスは以上のアルゴリズムに基づいて、たどり着く経路を計算しだし、定めることができる。

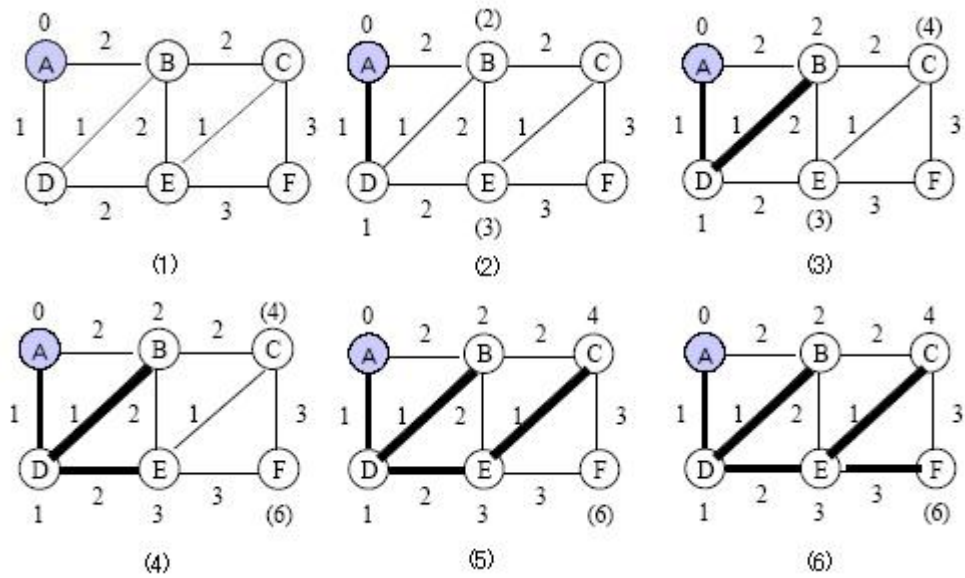


図 3.7: ダイクストラ法 (Dijkstra 法) のアルゴリズムを用いて、頂点 A から各頂点までの最短経路のプロセスである。太線は最短経路を表す、また細い線は一時パス表す。各数字はノード間の距離を表す。

ダイクストラ法（Dijkstra 法）では各頂点間のリンクは重みをつけることによって、各リンクを評価し、最適パスをみつけるアルゴリズムである。例ではリンクの重みを距離で表したが、水の流量などさまざまな目的に応じて重みを置き換えることができる。簡単に目的に応じて、ある点から各他の点までの最小値を求めるアルゴリズムであるとも言える。

L-R トレードオフ計算法

この L-R トレードオフ計算法（これから L-R 計算法と呼ぶ）は主なコンセプトとしてはワイヤレスセンサーネットワークの各リンクに対して、Minimum Latency（最小遅延）を持つパスより reliability の高い、また Maximum Reliability（最大信頼性）を持つパスより latency を小さいパスを見つけることによって、すべてのパスの中で、簡単に比較的 reliability や latency とともによいパスを計算し出す方法である。

またワイヤレスセンサーネットワークにおけるデータ伝送では latency と reliability また hop-count 三者は以下で述べる関係があるのはわかる。

- リンクの latency は小さければ小さいほどデータの伝送によい。
- リンクのデータ伝送率は大きければ多きほどデータの伝送によい。
- *hop - count* は小さければ小さいほどデータの伝送によい。

上のコンセプトに基づいて、L-R トレードオフ計算法は以下のように求め得られる。

$$P_{single} = \min_h \left\{ \min \left\{ \frac{L_{i-j}}{R_{i-j}} \right\} \right\} \quad (3.1)$$

- P_{single} : トレードオフされた最適ルーティングパスである
- L_{i-j} : ノード i とノード j 間のリンクの latency である
- R_{i-j} : ノード i とノード j 間のリンクの reliability である
- h : ノード i からノード j までの *hop - count* 数である

あるセンサーネットワークに対して、 L_{i-j} は隣接ノードのノード i とノード j をつながるリンク $i-j$ の Latency である、同様に R_{i-j} は隣接ノードのノード i とノード j をつながるリンク $i-j$ の Reliability である。そして、各リンクのリンクコストを $\frac{L_{i-j}}{R_{i-j}}$ となる。ここで、ダイクストラ法（Dijkstra 法）のアルゴリズムを用いて求める。上のリンクコストをダイクストラ法（Dijkstra 法）の重みとみなし、すべてのノード間のリンクに対して、最短パスつまりここでの最適ルーティングパス P_{single} となる。またもし最適値が同じなパスが存在するのであれば、最適パスは最小 *hop - count* 数を持つパスになる。またその上に同様な *hop - count* を持つパスが複数存在すれば、任意な一通パスを選択する。

3.3.4 コストによるルーティング

上の L-R トレードオフ計算法で求められたパスが決められれば、そのパスを sink が記憶しておく。そして、そのパスによって、sink がほしいデータを各ノードから取ってくるようにルーティングする。ルーティング方法としては、パケットの伝送の確認のために ACK を用いて実現する。たとえば、あるノード A があるノード B の観測データつまり生データを取ってくる場合、ノード A からノード B へ L-R パスのルーティングテーブルをおくり、もしノード B がその L-R パスのルーティングテーブルを受け取ったら、速やかにノード A へ向かって、確認メッセージ ACK をノード A に送り返す。そして、ノード A がその ACK メッセージを受け取って、ノード B への伝送が成功したと確認できるのである。

ここで図 3.4 のネットワークをもって以上のルーティングアルゴリズムのプロセスを詳しく説明する。L-R トレードオフ計算法により、計算しだしたトレードオフされたパスの一部は下のようである、

node1(sink)→node2→node3→node7→node11→node14

とすると、

ルーティングは以下ようになる。

まずノード 1 (sink) が最適パスを計算して、以下のようなルーティングテーブル (表 3.2) を生成される。

NodeID	NextNodeID
node1	node2 node3 node7 node11 node14 NULL

表 3.2: ノード 1 (sink) が持つ最適パスのルーティングテーブル

sink が 3.2 のテーブルを参照して、NextNodeID は node2 になっているので、このテーブルを次のノード node2 へおく。もし node2 がこのテーブルを受け取ると、sink へ確認メッセージの ACK 送って、次のようなテーブル (表 3.3) が生成される。

NodeID	NextNodeID
node2	node3 node7 node11 node14 NULL

表 3.3: ノード 2 が持つ最適パスのルーティングテーブル

またこのテーブルによって、次のノード node3 へこのテーブルを送る。

node3 が表 3.3 のテーブルを参照して、NextNodeID は node3 になっているので、このテーブルを続けと次のノード node3 へおく。もし node3 がこのテーブルを受け取ると、ACK を node2 へ送って、次のようなテーブル (表 3.4) が生成される。またこのテーブル

NodeID	NextNodeID
node3	node7 node11 node14 NULL

表 3.4: ノード 3 が持つ最適パスのルーティングテーブル

によって、次のノード node7 へこのテーブルを送る。

同じくこのパス上にあるすべてのノードがこのようなステップで、テーブルにある自分の次のノードを見つけて、テーブルを次のノードへ伝送する。

NodeID	NextNodeID
node7	node11 node14 NULL

表 3.5: ノード 7 が持つ最適パスのルーティングテーブル

NodeID	NextNodeID
node11	node14 node14 NULL

表 3.6: ノード 11 が持つ最適パスのルーティングテーブル

そして、ノード node14 が次のテーブルを node11 から受け取ると、次のノードは NULL であるので、自分がそのパスの目的地ノードであるとわかるので、自分の生データを直前ノードへ送る、そして、パス上の各ノードも同じく自分の直前ノードへその生データを送ることによって、最終的に sink へたどりついて、sink がその生データを受け取ることになるのである。

ここで、sink 側から、node14 のデータがほしいという場合、sink から node14 へのルーティングは以下ようになる。図 3.8。

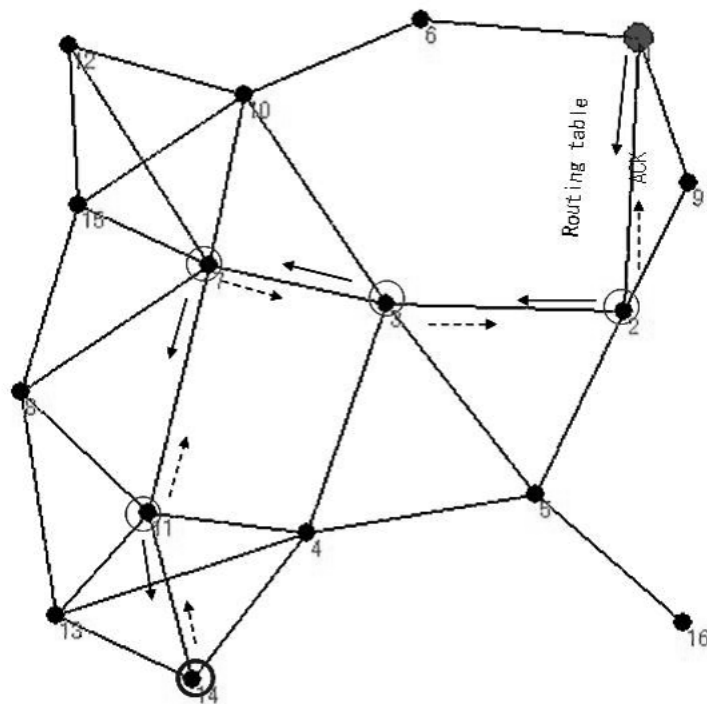


図 3.8: ここでノード node14 が目標ノードとし、ノード node7 が脱落ノードとする。実線はルーティングテーブルの伝送方向を表す。点線はルーティングテーブルの送信に対する確認メッセージ ACK を表す。

3.4 ルーティングにおける主な問題への対応

3.4.1 ノードの脱落問題への対応

ワイヤレスネットワーク特にワイヤレスセンサーネットワークではセンサーの電池切れなどによるノードの脱落はひとつ大きな問題であるので、ノードの脱落つまりリンクのオープンの問題が大きな課題である。ここでは以上の説明した L-R トレードオフ計算法と新たなルーティングアルゴリズムを用いてリンクのオープン問題を解決する方法を提案する。

われわれのルーティングアルゴリズムの中では、ノードの脱落に対しては主に2つのプロセスによって実現する。

- (1) ACK により、脱落ノードを察知する。
- (2) 同じトレードオフパスの値を持つ**マルチパス**を用いてルーティングする。
- (3) 違う**第二**の L-R パスを計算し、ルーティングパスをする。

(1) ACK により、脱落ノードを察知する。

上のルーティングはこれからシングルパスルーティングを呼ぶことにする。シングルパスルーティングでは sink から目標ノードまでたどりつくまで、各ノードからパケットの受信成立の確認のため、各ノードが自分直前のノードから受信した場合、その直前のノードへ ACK メッセージを送ることになっている。その ACK によって、直前のノードがルーティングテーブルの送信の成功が確認できるのである。もしこのとき、直前のノードが次のノードへルーティングテーブルを送信したあと $2T$ の時間つまり信号が届く最大範囲に要する時間内に次のノードから確認メッセージ ACK が送信してこない場合、次のノードが脱落したとみなし、自分がエラーメッセージには脱落ノードの ID をつけて、各自分の前のノードらを通して、つまり自分までたどってくるルートを逆の方向に沿って、最終的に sink へたどりついて脱落ノードを報告する。

(2) 最適マルチパスの計算

ルーティングする際ノード脱落が発生したら、sink がほしいデータが持っているノード、つまり目標ノードへルーティングテーブルを届いて、目標ノードが持つ生データを取ってくることでできないのである。対策として、まず上の (1) によって sink が脱落ノードが察知し、そして sink がそのノードに関するデータベースを修正する。つまり sink がその脱落ノードに関するリンクを自分が持っている各ノードのリンクデータから削除して、改めて目標ノードへのパスを再計算する。

前のシングル L-R 値をもつパスはすでに目標ノードへのシングルパス中での最適パスであるので、最適シングルパスと同じ L-R 値を持つパスはすでに存在しない。そこで最

適シングルパスと同じ L-R 値を用いて、ルーティングをするのにはマルチパスでルーティングすることしかないと考えられる。そのマルチ L-R ルーティングパスの計算法は以下のように提案する。

$$P_{multi} = \min_h \left\{ \frac{\max\{l_m\}}{Pr(r_1 \cup r_2 \cup \dots \cup r_m)} \right\} \quad (3.2)$$

$$= \min_h \left\{ \frac{\max\{l_m\}}{1 - (1 - r_1)(1 - r_2) \dots (1 - r_m)} \right\} \quad (3.3)$$

$$\leq P_{single} = \min_h \left\{ \min \left\{ \frac{l_0}{r_0} \right\} \right\} \quad (3.4)$$

ここで、

- P_{multi} : L-R マルチパスである
- l_m : 最適マルチルーティングパスの各パスの latency である
- r_m : 最適マルチルーティングパスの各パスの reliability である
- h : 最適マルチルーティングパスの合計 *hop-count* 数である
- m : 最適マルチルーティングパスのパス数である

シングルルーティングパスのなかでルーティング際パスのあるノードが脱落したとすると、最適パス上でそのノードの前後の隣接ノードの ID が sink が把握している。またそのシングル最適パス上で脱落ノードの直前ノードから sink までのルートとそのシングル最適パス上で脱落ノードの直後ノードから目標ノードまでのルートはすでに最適であるので、脱落ノードの直前ノードと直後ノードの間でのマルチパスから前のシングル最適パスと同じ最適値を持つ最適マルチパスを見つければよいのである。

ここで、ある i 通のマルチ最適パスを見つかったとすると、そのマルチパスは必ず上の式 3.2 を満たさねばならないのである。パスの reliability はリンク上でのデータやパケットの伝送信頼性というのであるが、あるソースノードから目的地ノードまでのデータやパケット伝送率ともいえる。つまり、あるデータやパケットの 100 パーセントでソースから目的地までへの確率でもある。 $Pr(r_1 \cup r_2 \cup \dots \cup r_m)$ と $1 - (1 - r_1)(1 - r_2) \dots (1 - r_m)$ は i 通最適マルチパスを通して、パケットが脱落ノードの直前ノードから直後ノードまで 100 パーセントたどり着く確率である、つまりパケットが脱落ノードの直前ノードから直後ノードまで 100 パーセントたどり着く reliability である。ここで、

- 確率の定理 (Theorem 6)
いかなる 2 つの事象 A、B に対しても、

$$Pr(A \cup B) = Pr(A) + Pr(B) - Pr(A \cap B) \quad (3.5)$$

を用いて、 $Pr(r_1 \cup r_2 \cup \dots \cup r_m)$ を計算することが可能である。例として、もしあるノード C までの reliability はノード A からの確率は 0.7 とし、あるノード B からの確率は 0.8 だとすると、ノード A とノード B から同時にノード C へ送るとするとその確率は

$$Pr(A \cup B) = Pr(A) + Pr(B) - Pr(A \cap B) \quad (3.6)$$

$$= 0.7 + 0.8 - 0.7 * 0.8 \quad (3.7)$$

$$= 0.94 \quad (3.8)$$

図 3.8 では、もし node7 が脱落したとすると、すでに sink から node3 までのパスと node11 から node14 までのパスは最適であるので、node3 と node11 のあいだのパスで最適マルチパスを見つければよいのである。もしそのマルチパスは：

- パス (1) : node3→node4→node11 :
Reliability : r_1
- パス (2) : node3→node10→node15→node8→node11:
Reliability : r_2
- パス (1) の latency を l_1 と、パス (2) の latency を l_2 とすると、 $l_1 < l_2$ 、

であるとする、式 3.2 を基づいて、以下の式が満たされる。

$$P = \frac{l_1}{r_1 + r_2 - r_1 * r_2} \quad (3.9)$$

$$\leq P_{single} = \min_h \left\{ \min \left\{ \frac{L_{1-14}}{R_{1-14}} \right\} \right\} \quad (3.10)$$

つまり、

$$P = \frac{l_1}{r_1 + r_2 - r_1 * r_2} \quad (3.11)$$

$$= \frac{\sum_{ij=1-2,2-3,3-7,7-11,11-14} L_{ij}}{R_{1-2} * R_{2-3} * R_{3-7} * R_{7-11} * R_{11-14}} \quad (3.12)$$

以上のように、われわれは式 3.2 を満たす i を脱落ノードの取り除いた各リンクの中で、そのマルチパスを沿ってルーティングすれば、必ずシングル最適パスと同じリンクコストを持つルーティングができると考えれる。

ここで、ネットワークを以下のように定義すると、

- 脱落ノードを除いたネットワークを N とする。
- 脱落ノードの直前ノードと直後ノードの間のネットワークを N_m とし、すべてのパスを P とする。
- N_m で直前ノードから直後ノードへ向かって、ルーティングする。

マルチルーティングパスの探索アルゴリズム
Multipath-searching algorithm for routing

procedure *Multipath-searching*(P_{min})

```

begin
   $P := \{P_1, P_2, P_3, \dots, P_n\};$ 
   $P_1 := \min\{P\};$ 
   $P_{min} := P_{single};$ 
   $P_{multi} := P_1;$ 
  for  $i := 1$  to  $n$  do
    if  $P_{multi} > P_{single}$  do
       $P_{multi} := P_{multi} \cup \min\{P - P_{multi}\}$ 
    else  $P_{min} := P_{multi};$ 
  end

```

表 3.7: マルチルーティングパスの探索アルゴリズム

- N_m で各パスを最適パスに近い順に最適パスを P_1 、第二最適パスを P_2 、第三最適パスを P_3 、... とする。

具体的のアルゴリズムは以下の図 3.9 ようになっている。

(3) 最適マルチパスによる再ルーティング

上の(2)で述べたように、最適マルチパスを見つければ、ルーティングをする。ここから、それについて詳しく紹介する。

ルーティングアルゴリズムは基本的に最適シングルパスのルーティング方法とは同じであるが、マルチパスでのルーティングでは複数のパスに対して、同じパケットやデータをそれに応じて、複数コピーをし、伝送しなければならないので、ここで上の例の図 3.4 のネットワークをもって、詳しく説明する。ここで、同じく node14 が目標ノードで node7 が脱落ノードとし、最適マルチパスは上のパス(1)とパス(2)を用いると sink からのマルチルーティングパスは：

- パス (M1) : node1(sink)→node2→node3→ node4→node11→node14 :
Reliability : r_1
- パス (M2) : node1(sink)→node2→node3→ node10→node15→node8
→node11→node14:
Reliability : r_2
- パス(1)の latency を l_1 と、パス(2)の latency を l_2 とすると、 $l_1 < l_2$ 、

node1(sink) から node2 へ以下の表 3.8 ようなルーティングテーブルを送る。

NodeID	NextNodeID
node1	node2
	node3
	node4
	node11
	node14
	NULL

表 3.8: ノード 1 (sink) が持つ最適パスのルーティングテーブル

またこのテーブルによって、次のノード node3 へこのテーブルを送る。node2 から node3 へ以下の表 3.9 ようなルーティングテーブルを送る。またこのテーブルによって、次のノード node3 へこのテーブルを送る。node3 から node4 へ以下の表 3.10 ようなルーティングテーブルを送る。

同じくノード node4, node11 をと次々ルーティングテーブルを運んで、最後に node14 が自分が目的地ノードであるので、自分が持つ生データをルーティングの前のノードを通して、sink へ送る。またこのパスと同じく、もうひとつのパスも同じく目的地ノード node14 へとルーティングテーブルを送って、再び、目的地ノードの node14 から生データを取ることにする。

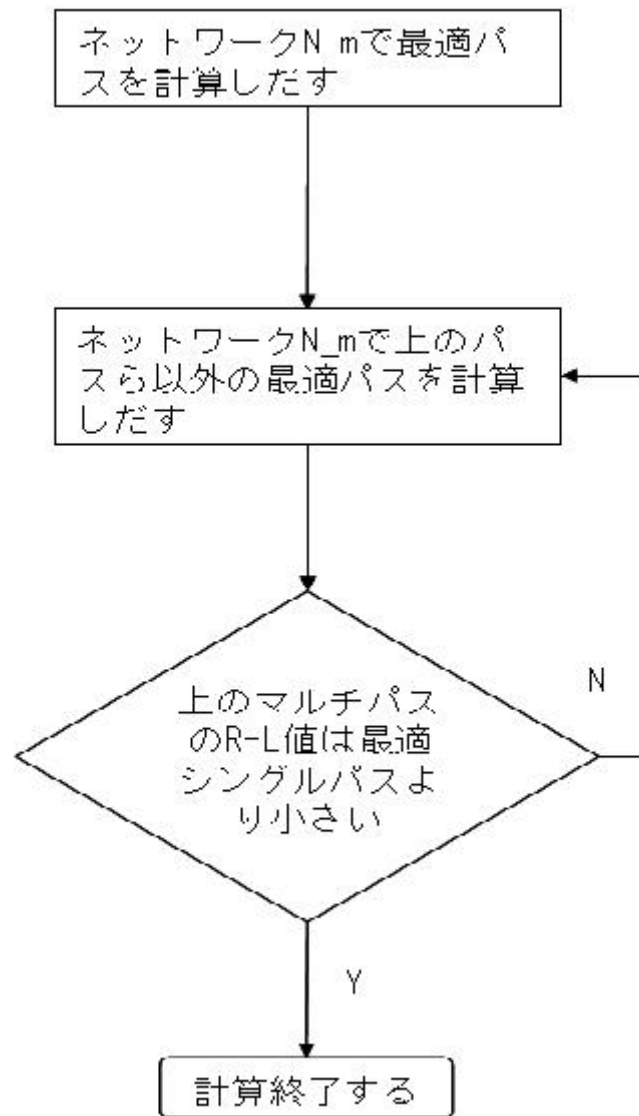


図 3.9: マルチルーティングパスの探索アルゴリズム

NodeID	NextNodeID1
node2	node3 node4 node11 node14 NULL

表 3.9: ノード 2 が持つ最適パスのルーティングテーブル

NodeID	NextNodeID1
node3	node4 node11 node14 NULL

表 3.10: ノード 3 が持つ最適パスのルーティングテーブル

NodeID	NextNodeID2
node1	node2 node3 node10 node15 node8 node11 node14 NULL

表 3.11: ノード 1 (sink) が持つ最適パスのルーティングテーブル

NodeID	NextNodeID2
node2	node3 node10 node15 node8 node11 node14 NULL

表 3.12: ノード 2 が持つ最適パスのルーティングテーブル

node1(sink) から node2 へ以下の表 3.11 ようなルーティングテーブルを送る。

またこのテーブルによって、次のノード node3 へこのテーブルを送る。node2 から node3 へ以下の表 3.12 ようなルーティングテーブルを送る。

NodeID	NextNodeID2
node3	node10 node15 node8 node11 node14 NULL

表 3.13: ノード 3 が持つ最適パスのルーティングテーブル

同じくテーブルにあるすべてのノード node15, node8, node11 を通して、再び node14 から生データを各前のノードを通して、最終的に sink へ送ってくる。ルーティングは以下図 3.10 のようになるのである。

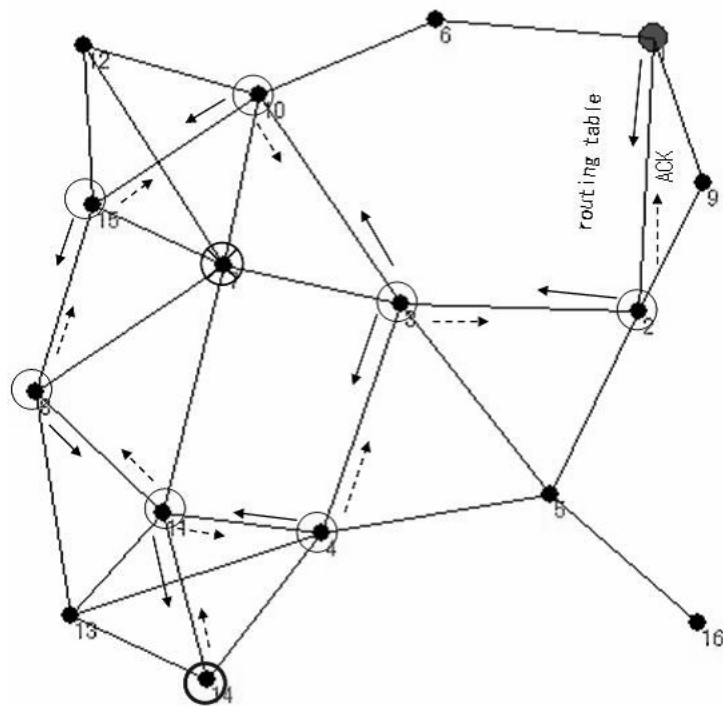


図 3.10: ここでノード node14 が目標ノードとし、ノード node7 が脱落ノードとする。実線はルーティングテーブルの伝送方向を表す。点線はルーティングテーブルの送信に対する確認メッセージ ACK を表す。

(4). 第二の L-R 最適シングルパスによるルーティング

もしマルチパスで最適シングルパスのノードの脱落問題をカバーできなくなると、第二最適パスつまりシングル最適パスを除いた最適シングルパスを選び出して、そのパスに沿ってルーティングする。ルーティングのアルゴリズムは最適シングルパスと同じであるので、3.3.4 を参照すること。ここでは省略する。

3.4.2 ノードの密集問題への対応

ワイヤレスネットワークとくにメッシュ型のワイヤレスセンサーネットワークではノードの密集問題もひとつ解決すべき問題である。われわれのルーティングアルゴリズムでは対象となるセンサーネットワーク、つまり 3.3.1 で述べたように各ノードの隣接ノードが 4 個から 8 個までであれば、そのネットワークのノード密集問題は解決できると考えられる。これからプロセスごとに説明する。

隣接ノードの情報交換する時の場合

各ノードが自分のすべての隣接ノードの latency と reliability 情報が sink に送らなければならないので、ここで各ノードに対して、信号が届く範囲内のノードをすべて隣接ノードをまなしているの、ノードは密集であるかそうでないかに関係なくわれわれのルーティングアルゴリズムに必要条件である。

ツリー構造に沿って sink に隣接ノード情報を集まる時の場合

各ノードの隣接ノードの latency と reliability 情報を sink に送るときに、ツリー構造は sink からの $infor\text{-}req$ の各ノードへの着順で決まるので、ノードは一個 father によって sink に送るので、情報の重複に伝送されることはないのである。

ルーティングする時の場合

sink から目標ノードへのルーティング時では、各直前ノードはルーティングテーブルによって、次のノードへ送るときは unicast をするので、影響はないと考えられる。

3.5 時間 $T_{infor\text{-}change}$ の決め方

時間 $T_{infor\text{-}change}$ は予め最初から、ユーザによる設定しておくのもよいが、ここでいくつかの提案をする。

- (1) ネットワークに使用するノードの仕様に対して、特にノードのバッテリーによって時間 $T_{infor-change}$ を設定する。
- (2) 各ノードが最適マルチパスのルーティングでもルーティング不可能の時、つまり各ノードに対して、ノードの配置から最適マルチルーティングパスが存在しないときまでの平均時間を $T_{infor-change}$ にする。

3.6 まとめ

この章でわれわれの L-R 計算法とそれに対応するルーティングアルゴリズムについて説明した。またその最適パスでノードの脱落が発生する場合に対する対処法についても説明した。

第4章 評価

4.1 はじめに

われわれは主に二つ項目について、われわれのルーティングアルゴリズムを検証する。

(1) L-R 計算法に対する検証

(2) オーバーヘッドに対する検証

4.2 L-R 計算法に対する検証

まず、L-R 計算法に対する検証を行う。ここでシミュレーターとして使うのは WSN Simulator というわれわれのアルゴリズムを検証するために Delphi 言語を使ったプログラミングにより作成されたワイヤレスセンサーネットワークルーティングアルゴリズムを検証するシミュレーターである。ここシミュレーターの機能として、主に二つである、(1) は指定したノードの数により、センサーネットワークのトポロジを生成する。(2) 生成されたネットワークに対して、sink から指定された目標ノードまでの Minimum Latency パス、Maximum Reliability パスと R - L 計算法によるトレードオフされたパス-最適シングルパスを生成する。最適マルチパスの原理計算法は主に最適シングルパスと同様なので、ここでは省略する。ここで、まず同じシミュレーターにより生成された図 3.4 のネットワークに対して、検証を行って説明する。それに対して、図 4.1 で Minimum Latency をもつパスを示している。

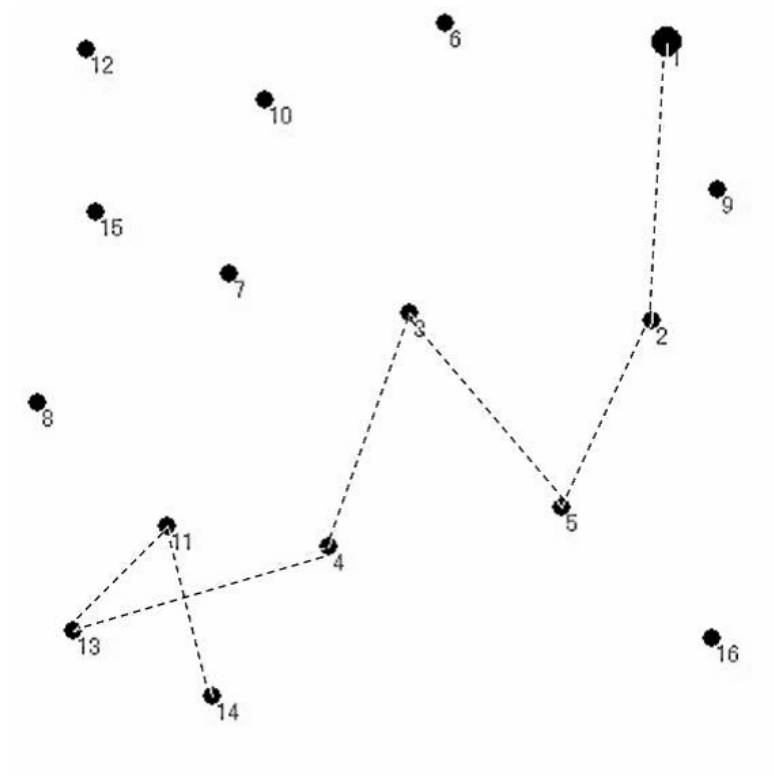


図 4.1: 図のパスは Minimum Latency をもつパスである。

そして図 4.2 は Maximum Reliability をもつパスを示している。

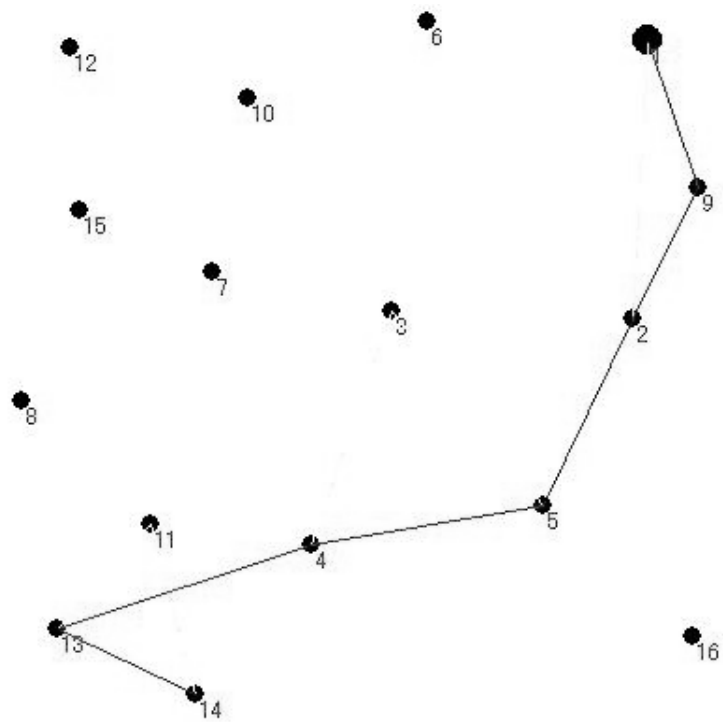


図 4.2: 図のパスは Maximum Reliability をもつパスを示している。

そして図 4.3 は L-R 計算法による R - L 計算法によるトレードオフされたパス-最適シングルパスを示している。

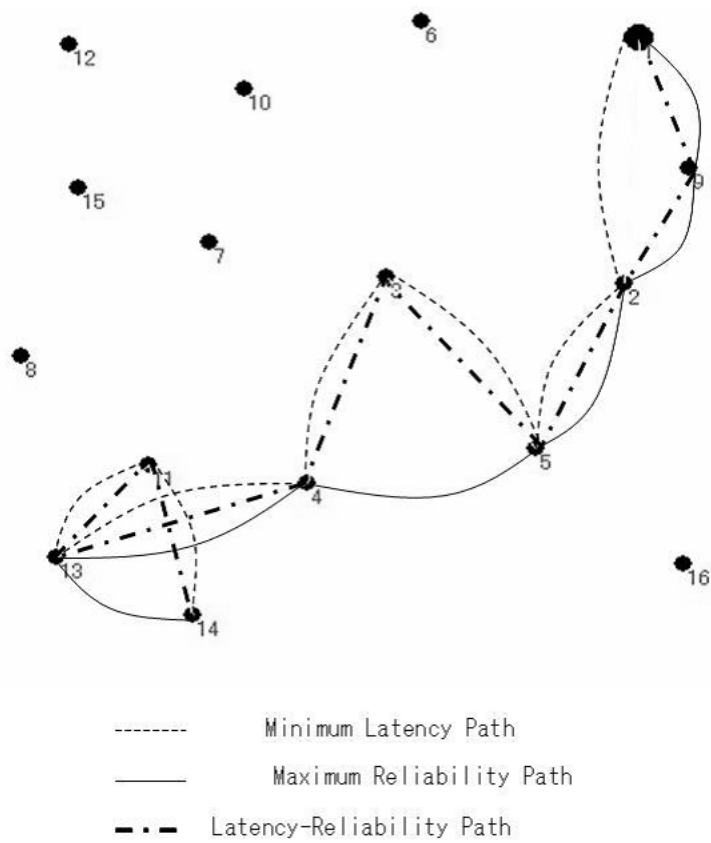


図 4.3: 図の各パスは R - L 計算法によるトレードオフされたパス-最適シングルパスと Minimum Latency パスと Maximum Reliability パスを示している。

同じようにノード数を 50 個まで増やして、検証した結果は以下図 4.4 と表 4.1 のようになっている。ここで、表 4.1 の生成された各パスの実験データを参考しながら、Minimum Latency 値、Maximum Reliability 値と L-R 計算法によるトレードオフされた最適値との関係を考察する。

	Minimum Latency(ms)	L-R 最適値	Maximum Reliability(Pr)
latency	319	413	529
reliability	0.5	0.7	0.8

表 4.1: ここで各ルーティングパス：Minimum Latency パス、Maximum Reliability 値パスと R - L 計算法による最適シングルパスの latency 値と reliability 値を示している。

この表 4.1 から Minimum Latency 値、Maximum Reliability 値と R - L 計算法によるトレードオフされた最適値の三者数字を比較すると以下表 4.2 のような関係は明らかである。

	Minimum Latency パス		L-R 最適パス		Maximum Reliability パス
latency:	319	<	413	<	529
reliability:	0.5	<	0.7	<	0.8

表 4.2: 最適の三者の数字を比較すると L-R 計算法よりルーティングパスつまり最適シングルパスはトレードオフされているのは明確である。

そしてルーティングは図 4.4 で同じように Minimum Latency をもつパスと Maximum Reliability をもつパスと L-R 計算法による R - L 計算法によるトレードオフされたパス-最適シングルパスを示している。

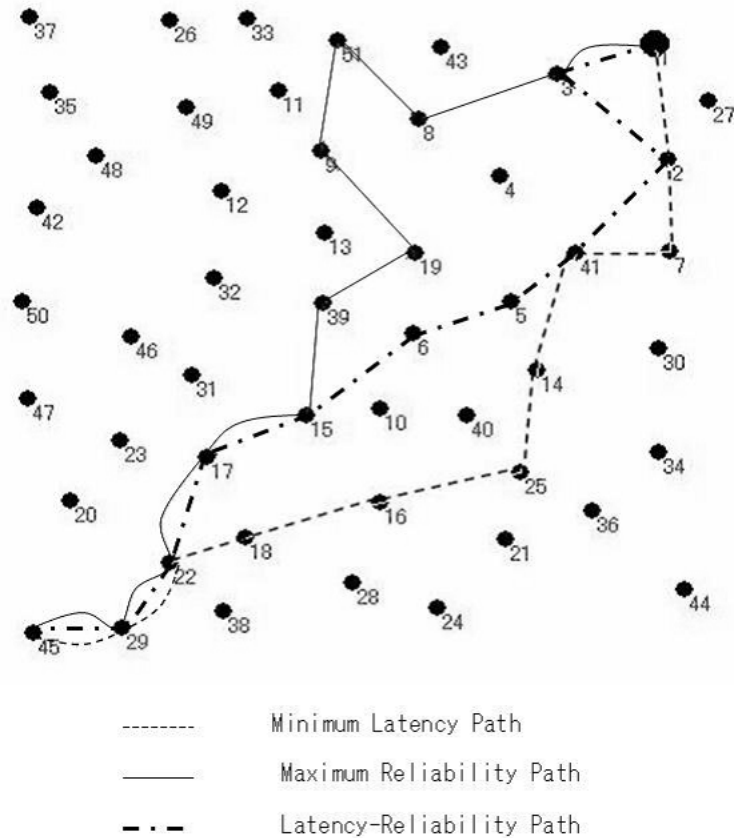


図 4.4: Minimum Latency をもつパス、Maximum Reliability をもつパスと L-R 計算法による R - L 計算法によるトレードオフされたパス-最適シングルパスを示している。L-R 最適パスはトレードオフされているのは確認できる。

これらの図と表を用いて、L-R 最適値はおおよび L-R 最適パスは Minimum Latency 値と Maximum Reliability 値の間でトレードオフされているのは明らかである、この結果によって、われわれの L-R 計算法およびルーティングアルゴリズムの提案は大いに成立すると考えられる。

そして、その上にわれわれはさらにノード数を増やして実験して考察する。その結果は以下の図 4.5 と図 4.6 のようになっている。ここでもノード 50 個からノード 150 個までの時も同じく L-R 最適パスはトレードオフされていることがわかる。上の実験結果により、われわれの L-R 計算法により、各リンクおよびパスがそれぞれの latency 値と reliability 値とトレードオフできるのを確認することができた。

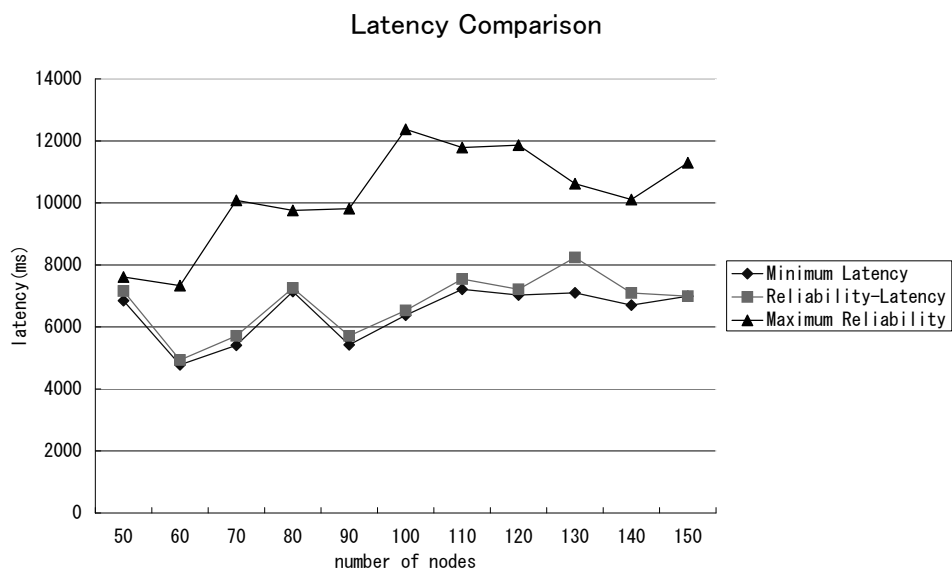


図 4.5: L-R 最適パスの結果を表す線は Minimum Latency パスと Maximum Reliability パスの中間にあり、latency 値はトレードオフされているのはわかる。

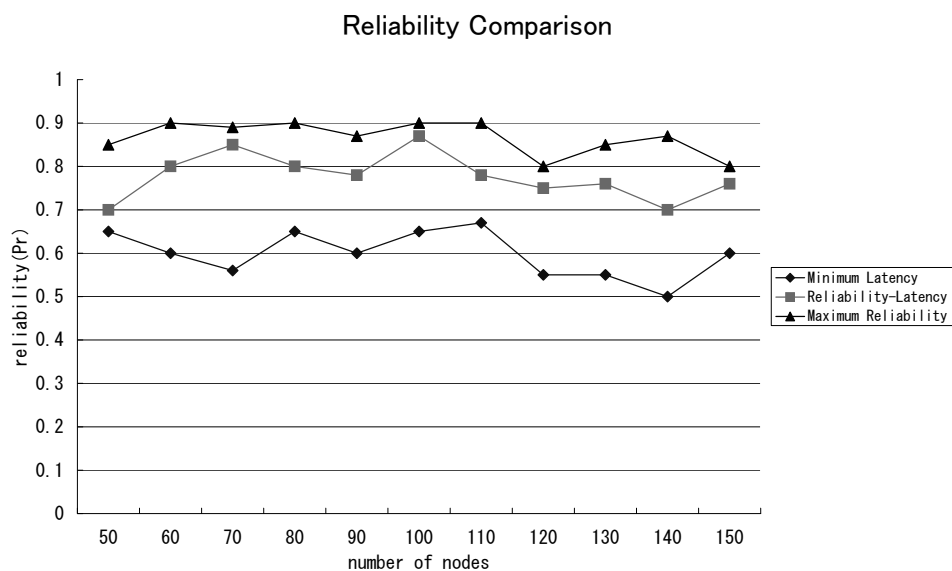


図 4.6: L-R 最適パスの結果を表す線は Minimum Latency パスと Maximum Reliability パスの中間にあり、reliability 値はトレードオフされているのはわかる。

4.3 オーバーヘッドに対する検証

われわれのルーティングアルゴリズムを用いることでルーティングでのオーバーヘッドの削減も以下の各プロトコルとの比較でわかる [21]。上の各ワイヤレスネットワークの

Parameters	DSDV	WRP	OLSR	TBRPF	GSR/FSR	STAR	FSLs
Time Complexity (link addition/failure)	$O(D)$	$O(H)$	$O(D)$	$O(H)$	$O(D)$	$O(H)$	$O(D)$
Communication Complexity (link addition/failure)	$O(N)$	$O(N)$	$O(N)$	$O(N)$	$O(N)$	$O(N)$	$O(N)$
Computation Complexity	$O(N)$	$O(N)$	$O(N^2)$	$O(N^2)$	$O(N^2)$	$O(N^2)$	N/A
Memory Complexity	$O(N)$	$O(N)$	$O(N^2)$	$O(N^2)$	$O(N^2)$	$O(N^2)$	N/A
Data Complexity	$O(N)/I$	$O(N)/I$	$O(d)/I$	$O(N)/I$	$O(d+O(N-d)/I)$	N/A	N/A
Packet Complexity	$O(1)$	$O(1)$	$O(N)$	$O(N)$	$O(1)$	N/A	N/A
Convergence Time	$O(N \cdot I)$	$O(N \cdot I)$	$O(D)$	$O(D \cdot I)$	$O(D \cdot I)$	$O(D)$	$O(D \cdot I)$
Type of Routing Information	Distance Vector	Distance Vector	Link State (MPR selectors)	Link State	Link State (fisheye)	Link State (source tree)	Link State
Destination of Routing Information	Neighbors	Neighbors	Network	Network	Neighbors	Neighbors	Neighbors / Network
Nodes relaying updates	All	All	MPR	RPF	All	All	All
Loop Free	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Routing Metric	Shortest Path	Shortest Path	Variable	Variable	Variable	Variable	Variable
Unidirectional Link Support	No	No	Yes	No	Yes	Yes	N/A
Frequency of Update Transmissions	Periodically and as needed	Periodically and as needed	Periodically and as needed	Periodically	Periodically	As needed	Periodically
Multicast Capability	No	No	Yes	No	No	No	N/A
Number of Required Tables	2	4	5	3	4	6	N/A
Multipath Routing	No	No	Yes	Yes	Yes	No	Yes
Utilizing Hello Messages	Yes	Yes	Yes	Yes	No	No	N/A

N = Number of nodes in the network, D = Network diameter, d = Degree of node connectivity, I = Routing update interval, H = height of routing tree

図 4.7: 主な ad hoc network のプロトコルの比較その 1

プロトコルでわれわれのルーティングアルゴリズムと同様の proactive ルーティングアルゴリズムをするプロトコルをピックアップして比較する。

上の表 4.3 での比較で、われわれのルーティングアルゴリズムはほかの主なワイヤレスルーティングプロトコルのルーティングアルゴリズムよりルーティングテーブルを交換するノード数とルーティングする際に要するテーブル数が少ないのはわかる。

Comparisons of Characteristics of Reactive Routing Protocols

Parameters	DSR	AODV	SOAR	LMR	TORA	ABR	SSA	LBAR	DLAR
Time Complexity (initialization)	$O(2D)$	$O(2D)$	$O(2D)$	$O(2D)$	$O(2D)$	$O(D+Z)$	$O(D+Z)$	$O(D+Z+T)$	$O(D+Z+T)$
Time Complexity (postfailure)	$O(2D)$ or 0^*	$O(2D)$	$O(2l)$	$O(2l)$	$O(2l)$	$O(l+Z)$	$O(l+Z)$	$O(D+Z)$	$O(D+T+2Z)$
Communication Complexity (initialization)	$O(2N)$	$O(2N)$	$O(2N)$	$O(2N)$	$O(2N)$	$O(N+Y)$	$O(N+Y)$	$O(N+Y)$	$O(N+Y)$
Communications Complexity (postfailure)	$O(2N)$	$O(2N)$	$O(X)$	$O(2X)$	$O(2X)$	$O(X+Y)$	$O(X+Y)$	$O(N+Y)$	$O(N+2Y)$
Unidirectional Link Supporting	Yes	No	Yes	No	No	No	No	No	No
Flooding	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Route Recovery Methodology	Erase route; notify source	Erase route; notify source	Route repair	Route repair	Route repair	Erase route; Localized query	Erase route; notify source	Notify destination	Erase route; notify source
Partial Route Recovery	No	No	Yes	Yes	Yes	Yes	No	Yes	No
Routing Strategy	Source Routing	Hop-by-hop	Hop-by-hop	Link Reversal	Link Reversal	Hop-by-hop	Hop-by-hop	Hop-by-hop	Hop-by-hop
Loop Free	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Multicast Capability	No	Yes	Yes	Yes	No	No	No	No	No
Beaconing Requirements	No	No	No	No	No	Yes	Yes	No	No
Multipath Possibility	Yes	No	No	Yes	Yes	No	No	No	No
Routing Metric	Shortest Path	Freshest and Shortest Path	Shortest Path	Shortest or Random Path	Shortest Path	Associativity and Shortest path and Load and Delay	Signal strength and Associativity	Load	Load and shortest path

* Cache hit.
L = Number of links in the network, X = Number of nodes affected by a topological change, Y = Total number of nodes forming the directed path where the REPLY packet transits, Z = Diameter of the directed path where the REPLY packet transits, T = Route selection waiting time, l = Diameter of the affected network segment

図 4.8: 主な ad hoc network のプロトコルの比較その 2

Parameters	DSDV	WRP	OLSR	GSR	STAR	L-R
Type of Routing Information	Distance Vector	Distance Vector	Link State (MPR selectors)	Link State (MPR selectors)	Link State (source tree)	Link State (L-R)
Destination of Routing Information	Neighbors	Neighbors	Network	Neighbors	Neighbors	a part of Neighbors
Nodes relaying updates	All	All	MPR	All	All	All
Routing Metric	Shortest Path	Shortest Path	Variable	Variable	Variable	Variable
Number of Required Tables	2	4	5	5	6	2
Multipath Routing	No	No	Yes	Yes	No	Yes
Utilizing Hello Messages	Yes	Yes	Yes	No	No	Yes

表 4.3: 主な Proactive ルーティングアルゴリズムをするプロトコルとのオーバーヘッドの比較

4.4 まとめ

われわれは主に提案したアルゴリズムに対して、L-R 計算法に対する検証およびオーバーヘッドに対するの検証を行った。実験結果から以下の結論が得られる。

- あるネットワークに対して L-R 計算法によるネットワークの latency と reliability の両パラメータをトレードオフすることができる。
- われわれのルーティングアルゴリズムを用いたルーティングはある程度ネットワークのオーバーヘッド問題を軽減できる。

上の実験データを参考しながら、われわれの L-R 計算法およびルーティングアルゴリズムがある程度の効果が見られるのは明確であるが、さらに他シミュレーターでの検証やリアルネットワークでの検証を行って考察する必要もあると考える。

第5章 討論

5.1 はじめに

この章ではわれわれの L-R 計算法やルーティングアルゴリズムの利用や拡張についていくつか提案して討論する。

5.2 on-demand ルーティングアルゴリズムへの適用

ワイヤレスセンサーネットワークを含め、ネットワークのルーティングアルゴリズムは伝送方式から大いに2つある。proactive routing（また table-driven routing と呼ぶ）と reactive routing（また on-demand routing や real-time routing と呼ぶ）である。われわれの L-R 計算法では pro-active つまり table-driven 式ルーティングアルゴリズムを用いているが、ここでわれわれの L-R 計算法を on-demand 式のルーティングアルゴリズムで行う場合について討論する。

われわれの L-R 計算法はすべてのノードの隣接ノード情報を式 3.1 でリンクコストを計算し、ダイクストラ法（Dijkstra 法）のアルゴリズムで最適パスを計算しだす方法であるので、各ノードの隣接ノードの情報だけでなく、すべてのノード間の latency と reliability 情報を集めなければならない。われわれの L-R 計算法をあるアルゴリズムで情報を各ノードが収集できるようにすることで実現できればよい。

ひとつ方法として flooding ルーティングアルゴリズムを利用する方法である。各ノードは隣接ノードを含め、ネットワークのトポロジーにあるすべてのノード間でリンク情報を交換し、すべてのノードがネットワーク全体のリンクの情報を取得することができる。flooding により、各ノードはネットワーク全体のトポロジーとすべてのリンクの情報を取得することができる。つまり、各ノードが従来第三章で述べたルーティングアルゴリズムでの sink ノードの役割のような働きを果たすことである。そして各リンクの情報により、各ノードが自分から各ノードに対しての目的地ノードである sink までのトレードオフされた最適パスが見つかることができる。

ルーティングアルゴリズムでは、各ノードは sink ノードからリクエストなしで自らセンシングしたデータを sink ノードへ送ることができる。つまり node-to-sink 型のルーティングをすることで、sink へセンシングして取得した生データを sink へ送ることができる。ここで、アドホックネットワークの reactive ルーティングアルゴリズムをするプロトコル AODV 類似のアルゴリズムを用いてルーティングすることで実現可能と考えられる。

欠点としては flooding を使うことによって各ノード間の情報交換には大きなオーバーヘッドが生じるのは予想される。

そこで、われわれはもうひとつ flooding を使わないつまりダイクストラ法 (Dijkstra 法) を使わない L-R 計算法を利用した on-demand ルーティングするアルゴリズムの提案を試みる。

(1) まず、各隣接ノード間のリンクの情報収集

各ノードは自分がセンシングした生データを目的地ノードである sink に送信しようとするとき、各ノードは自分の隣接ノードとの間のリンク情報を自分に収集する。第三章と違って、この場合の on-demand ルーティングでは、sink へ送信するのは各ノードから sink へ主動的送信するので、sink からの infor-req メッセージや隣接情報ツリーは不要である。そしてルーティング時に各ノードは自分の隣接ノードとだけ情報交換をする。

(2) 各ノードは自分が隣接ノードから収集した latency と reliability 情報を L-R 計算法を利用し、第三章で述べたの L-R 計算法の以下の基本的なコンセプトを用いてリンクコストを計算する。

$$P = \frac{L_{i-j}}{R_{i-j}} \quad (5.1)$$

各隣接のリンクの中から最小 L-R 値を持つパスを選び、ルーティングすることはできるが、それは第三章で述べた最適パスとは異なり、単なる局部域での最適値であるので、全体のネットワークにおいてこの局部域の最適パスに沿ってルーティングするのは必ず有効であるとはいえない。また各ノードはネットワーク全体のトポロジーの情報の取得することがないので、ネットワークすべてのリンクの情報が取得できなく、第三章で述べたように最適シングルパスは得ることはできないのである。つまりダイクストラ法 (Dijkstra 法) はここで適用できないのである。

そこでわれわれは第三章のマルチパスアルゴリズムを活用して、あたらなシングルパスとマルチパスを併用するハイブリッドルーティングアルゴリズムを用いることで、各ノードがある一定なリンクコストの要求を満足できるパスによるルーティングは可能と考えられる。

あるノードから sink までのリンクコストの条件は

$$P \leq P_d = \frac{L_{i-j}}{R_{i-j}} \quad (5.2)$$

とする、

つまり各ノードは自分の隣接ノードのリンクの中で $\frac{L_{i-j}}{R_{i-j}}$ の要求値 P_d より小さい L-R 値 P をもつリンクを選択して、ルーティングすればよい。またこのようなパスが存在しない場合はマルチパスを適用する。隣接のリンクの中では要求値 P_d より小さい L-R 値 P を持つパスは存在しないなら、各リンクから最小 L-R 値を持つリンクから要求値を達するまで順にマルチパスの本数を増やす。つまり隣接のリンクからマルチパスで reliability をより

大きくすることで P_d をより要求値に近づかせることで、ルーティングパスを発見する。そして P は以下の式を必ず満たす。

$$P = P_1 \cup P_2 \cup P_3 \cup \dots \quad (5.3)$$

$$\leq P_d = \frac{L_{i-j}}{R_{i-j}} \quad (5.4)$$

ここで $P_1, P_2, P_3, \dots$ は P の L-R 値が小さい順に並ぶ隣接ノードの各リンクを指す。そしてルーティング時に各ノードはこの式を満たすリンク先にあるノードへデータを送り、さらにこのような各中間ノードの伝送により、最終的に sink にデータを送ることができる。

このルーティングアルゴリズムの欠点としてリンクの方向性を最初から改めて、セッティングする必要があると考えられる。それはメッシュのトポロジーのセンサーネットワークで解決すべきであるもうひとつ重要な課題である location problem に深くかかわる [22][23]。

このような on-demand ルーティングアルゴリズムは災害発生時の警報などのアプリケーションに比較的有効だと考えられる。

5.3 まとめ

この章でわれわれの計算法を on-demand ルーティングアルゴリズムへ利用する場合についての提案を簡単に述べた。これらの提案や研究を深めることにより、センサーネットワークのアプリケーションはさらに広げることが考えられる。

第6章 結論

6.1 まとめ

本研究ではワイレスセンサーネットワークにおけるデータ伝送の latency とデータ伝送の reliability の両パラメータを同時に考え、そして効率よく両パラメータをトレードオフする計算法 L-R 計算法を提案した。

- まずワイヤレスネットワークのデータ伝送における主な両パラメータを選び、それらを効率よくトレードオフする計算法を研究し、ダイクストラ法（Dijkstra 法）のアルゴリズムを用いて、最適 L-R ルーティングパスの計算を提案した。
- 提案した計算法に対してワイヤレスセンサーネットワークシミュレーターを用いて検証を行い、latency と reliability のトレードオフすることができることが確認できた。

6.2 今後の課題

今後の研究課題としては以下のいくつか課題を挙げられる。

- 実環境への適用と検証
センサーネットワークの実環境を想定して、パケットの制御などによる reliability と latency のトレードオフできるアルゴリズムにするために、より実環境で効率よく高い QoS を提供できるように考察と検証を行うべきと考えられる。
- 大規模のセンサーネットワークへの実装と検証
今後われわれの L-R 計算法をより大規模なワイヤレスセンサーネットワークへの実装や実際のアプリケーションによっての大規模センサーネットワークでの実装およびそれに対する検証を行う。
- 他パラメータのトレードオフ
ワイヤレスネットワークのデータ伝送における他パラメータ、たとえば、エネルギー、帯域幅などを入れてのトレードオフ法を提案し、それに対する実験を行う。

謝辞

本研究を進めるにあたり、終始熱心なご指導、ご鞭撻を下さりました指導教官である丹康雄助教授を深く感謝申し上げます。また研究を行うにあたり多くのご意見を下さりました中田潤也さん、牧野義樹さん、Peng, Chao 博士に感謝いたします。また研究に多いご助言を下さりました張海波さんを深く感謝致します。そして、丹研究室の皆さんに感謝致します。最後に研究生活で励ましてくれました家族の皆さんに感謝します。

付 録 A 用語一覧

ad hoc networks	アドホークネットワーク
QoS	Quality of Service
WSNs	Wireless Sensor Networks
AODV	Ad hoc On-demand Distance-Vetor Routing Protocol

参考文献

- [1] J. N. Al-Karaki and A. E. Kamal. Routing techniques in wireless sensor networks: a survey. *IEEE Wireless Communications*, 2004.
- [2] Archana Bharathidasan and Vijay Anand Sai Ponduru. Sensor networks: An overview. 2004.
- [3] D. Chen and P.K. Varshney. Qos support in wireless sensor networks: A survey. *Proc. Int 'l Conf. Wireless Networks (ICWN 04)*, 2004.
- [4] Mohamed Eltoweissy Mohamed Younis, Kemal Akkaya and Ashraf Wadaa. On handling qos traffic in wireless sensor networks. *IEEE Conference Proceeding*, 2004.
- [5] 柳澤 政生. センサネットワーク. <http://www.yanagi.comm.waseda.ac.jp/sensornet/>.
- [6] 鄭 立. *ZigBee 開発ハンドブック*. リックテレコム, 2006.
- [7] 西室 洋介 川崎 光博 福井 博 坂田 史郎, 田中 成興. *ZigBee センサーネットワーク 通信基盤とアプリケーション*. 株式会社 秀和システム, 2005. pp.162-164.
- [8] アンドリュー・S・タネンバウム. *コンピュータネットワーク 第3版*. 株式会社ピアソン・エデュケーション, 1999.
- [9] Andrew S. Tanenbaum fourth edition. *Computer Networks*. Prentice Hall, fourth edition, 2002.
- [10] Sudeept Bhatnagar B.Deb and Badri Nath. Information assurance in sensor networks. *WSNA of ACM*, 2003.
- [11] Chenyang Lu Tian He, John A Stankovic and Tarek Abdelzaher. Speed: A stateless protocol for real-time communication in sensor networks. *International Conference on Distributed Computing Systems (ICDCS 2003)*, 2003.
- [12] Sudeept Bhatnagar B.Deb and Badri Nath. Reinform: Reliable information forwarding using multiple paths in sensor networks. *28th Annual IEEE conference on Local Computer Networks (LCN 2003)*, 2003.

- [13] Sudeept Bhatnagar B. Deb and Badri Nath. Information assurance in sensor networks. *WSNA of ACM*, 2003.
- [14] M. Potkonjak S. Meguerician, F. Loushanfar and M. B. Srivastava. Coverage problems in wireless ad-hoc sensor networks. *proceedings of IEEE Infocom*, 2001. pp.1380-1387.
- [15] G. Qu S. Meguerician, F. Loushanfar and M. Potkonjak. Exposure in wireless ad-hoc sensor networks. *Mobile Computing and Networking*, 2001. pp.139-150.
- [16] L. Kleinrock and J. Silvester. Optimum transmission radii for packet radio networks, or why six is a magic number. *national telecommunications conference*, 1978.
- [17] Yueming Hu Bo Li Chonggang Wang, K. Sohraby and Weiwen Tang. Issues of transport control protocols for wireless sensor networks. 2005. IEEE Conference Proceeding.
- [18] Youk Kwon. Reliable transport protocol for wireless sensor networks.
- [19] 浅野孝夫. **情報の構造 [下] — ネットワークアルゴリズムとデータ構造.**
- [20] Ronald L. Rivest Clifford Stein Thomas H. Cormen, Charles E. Leiserson. *Introduction to Algorithms*. the MIT Press, 2 edition, 2001.
- [21] Xiaobing Hou. Topology-based routing algorithms for ad hoc networks. 2002.
- [22] C-C Han A. Savvides and M. Srivastava. Dynamic fine-grained localization in ad-hoc networks of sensors,. *Proceedings of the Seventh ACM Annual International Conference on Mobile Computing and Networking (MobiCom)*, 2001. pp.166-179.
- [23] Jamal N. Al-Karaki and Ahmed E. Kamal. Routing techniques in wireless sensor networks: A survey. *IEEE Wireless Communications*, 2004.