

Title	アプリケーション特化センサーネットワークの設計と 検証に関する研究
Author(s)	安田, 真悟
Citation	
Issue Date	2014-03
Type	Thesis or Dissertation
Text version	ETD
URL	http://hdl.handle.net/10119/12100
Rights	
Description	Supervisor:篠田 陽一, 情報科学研究科, 博士

博士論文

アプリケーション特化センサーネットワークの
設計と検証に関する研究

安田 真悟

主指導教員 篠田 陽一

北陸先端科学技術大学院大学
情報科学研究科

平成26年3月

目次

1	序論	1
1.1	情報の収集, 伝達と共有	1
1.2	ユビキタスコンピューティングとセンサーネットワーク	2
1.3	センサーネットワークの設計・検証における課題	3
1.4	本論文の構成	6
第I部 センサーネットワークの現状と課題		9
2	センサーネットワークの発展	10
2.1	固定センサーノードによるセンサーネットワーク	10
2.2	移動センサーノードによるセンサーネットワーク	12
2.2.1	アドホックセンサーネットワーク	12
2.2.2	DTN センサーネットワーク	13
2.2.3	未知の環境特性を含むセンサーネットワーク	13
2.3	断続的通信環境下で観測を行うセンサーネットワークの課題	15
2.3.1	断続的通信環境下で観測を行うセンサーネットワーク設計	15
2.3.2	センサーネットワークの設計とデータ解析	17
2.3.3	ワイヤレスネットワーク実証検証環境	17
第II部 対象環境の物理特性の積極的推定とデータ解析		23
3	未知の特性を含む環境を対象とするセンサーネットワークの検討事例	24
3.1	動物観測ネットワーク	24
3.2	動物観測ネットワーク設計における技術課題	27
3.2.1	動物の行動モデルと間欠通信の時間間隔	27
3.2.2	ルーティング	28

3.2.3	ノードの時間同期	29
3.2.4	IEEE802.15.4 テストベッドによる事前技術検証	29
3.3	車車間通信ネットワーク	30
3.3.1	エミュレーション手法	31
3.3.2	車の移動データ	31
3.3.3	ネットワークに送信するデータ	32
3.3.4	実験結果	33
3.4	車車間通信ネットワーク設計における技術課題	35
3.4.1	ノードのモビリティ	35
3.4.2	ソフトウェアの検証	35
3.5	未知の特性を含む環境を想定したセンサーネットワーク設計に必要な特性	35
3.5.1	ノードの移動モデル	36
3.5.2	対象環境での電波伝播の再現	36
4	ワイヤレスネットワークにおける地形の影響	37
4.1	検証用機器による電界強度測定	38
4.2	地形データを用いた電波エミュレーション	42
4.2.1	電波伝播モデル	43
5	センサーネットワークにおける取得データの推定と解析	47
5.1	取得可能データの推定	47
5.2	展開後の実取得データの解析	48
第 III 部 対象環境の物理特性を考慮した大規模実証実験環境		49
6	ネットワーク実証検証環境の構築確度向上と効率的な構築技術	50
6.1	既存のネットワーク実証検証環境の構築手法	50
6.2	既存の実証検証環境構築手法	51
6.2.1	実ノードの作成と複製	51
6.2.2	仮想ノードのディスクイメージの配布	51
6.3	ネットワーク実証検証環境構築の精度	51
6.3.1	電源管理障害	52

6.3.2	管理サーバの障害	52
6.3.3	ハードウェアの障害	53
6.3.4	その他	53
6.4	マルチキャストを用いたデータ配布	53
6.4.1	実験制御	53
6.4.2	実ノードの起動方式	55
6.4.3	電源制御	57
6.4.4	電源制御の並列処理	57
6.4.5	PXEBoot で使用するサーバの負荷分散	59
6.5	プロトタイプによる評価	62
6.5.1	実ノードの OS 作成	62
6.5.2	管理系サーバ	64
6.5.3	Node Agent および Manager Server の実装	64
6.5.4	計測	65
6.5.5	ディスクイメージ配布のボトルネックに関する検討	68
6.5.6	実証検証環境構築におけるデータ配布手法に関する議論	73
7	汎用的な構造指示ファイルによる視覚的な実験環境構築	75
7.1	大規模実証検証環境の初期利用時の課題	75
7.2	統合実験支援ツール StarBuilder	76
7.2.1	実験ノード・ネットワークトポロジの構成記述方式	77
7.2.2	GUI による環境記述方式	81
7.2.3	SpringOS との連携	81
7.2.4	全体設計と外観, コンポーネントの共有	81
第 IV 部 アプリケーション特化センサーネットワークの設計検証		85
8	アプリケーション特化型センサーネットワークの設計検証フレームワークの提案	86
8.1	ネットワークシステム開発の設計検証におけるフェージング	87
8.2	論理的検証フェーズ	91
8.2.1	論理的検証フェーズで考慮すべき環境特性	91

8.2.2	ノードの移動特性の推定	92
8.2.3	長期生態観測システムにおける野生動物の行動推定	93
8.2.4	データ伝播モデル	97
8.2.5	ノードの移動・情報伝播モデルを用いた論理的検証	97
8.2.6	環境特性の積極的推定による設計へのフィードバック	102
8.2.7	環境特性の積極的推定による設計へのフィードバックの効果	109
8.3	機能検証フェーズ	109
8.3.1	機能検証に特化した検証環境構築支援ツール	112
8.4	実証検証フェーズ	113
8.4.1	車車間通信の地形を考慮したエミュレーションによる実証検証	113
8.4.2	アプリケーションに特化した環境特性エミュレーションによる効果	118
8.5	解析/フィードバックフェーズ	120
9	結論	122
9.1	本研究の成果	122
9.2	本研究分野の課題と展望	124
	謝辞	129
	本研究に関する発表論文	131

目 次

1.1	センサーネットワークシステムの開発・検証サイクル	4
1.2	新しいセンサーネットワークシステムの発案と技術要素の選択	5
1.3	アプリケーションに特化し、物理特性を考慮したセンサーネットワークシステムの開発・検証サイクル	7
2.1	センサーネットワークの拡大	10
2.2	有線によるセンサーネットワーク	11
2.3	有線の置き換えとしてのワイヤレスセンサーネットワーク	11
2.4	アドホックセンサーネットワーク	13
2.5	DTN センサーネットワーク	14
2.6	一般的な技術開発工程	18
2.7	StarBED の提案する技術開発工程	19
2.8	開発ステップと考慮される特性	20
3.1	首輪+ねぐらネットワーク	25
3.2	アンテナ地上高が低い場合のフレネルゾーン	30
3.3	実験のスクリーンショット	32
3.4	移動モデル毎の片道遅延累積分布 (CDF) グラフ	33
3.5	移動パターン A における車の台数毎の累積分布 (CDF) グラフ	34
4.1	白峰農業体験学習施設多目的広場:白山市	38
4.2	計測の様子	39
4.3	石川県林業試験場 (樹木公園):白山市	39
4.4	検証用送信機 (左)・受信機 (右)	40
4.5	実験機による電界強度測定結果 (平地)	42
4.6	平地と森林での電界強度の比較	43
4.7	フレネルゾーン	44

4.8	ナイフエッジ回折	44
4.9	二重ナイフエッジ回折	45
4.10	仮想単一遮蔽ナイフエッジ回折	45
6.1	マルチキャストを用いた実証検証環境構築の手順	54
6.2	Node Agent を用いた実験制御フレームワークの概要	55
6.3	Power Status 取得時の状態遷移図	58
6.4	Power ON 時の状態遷移図	58
6.5	NFS サーバ IP アドレスの問い合わせ	60
6.6	VM image file の配布	63
6.7	実ノード起動時間	66
6.8	実ノード初期化と仮想ノードのディスクイメージ配布時間	67
6.9	マルチキャストにおけるデータ転送の流れ	68
6.10	パケットのシーケンス番号とパケット損失率	69
6.11	HDD の書き込み位置毎の書き込み性能の時間遷移	71
6.12	各転送速度毎の積算書き込み量とデータ損失率	72
7.1	SpringOS で Pickup(ノードイメージの複製) 作業中のスクリーンショット	76
7.2	StarBuilder のアーキテクチャ概要	82
7.3	Omnigraffle [®] で実験環境を記述している画面	83
7.4	保存されているコンポーネント一覧画面	84
8.1	新しいセンサーアプリケーションの発案	88
8.2	既存の設計検証手法における課題	89
8.3	アプリケーション特化型センサーネットワークの設計・検証フェーズ	90
8.4	論理的検証段階における対象環境の特性とその影響	92
8.5	Levy Walk での移動軌跡	94
8.6	Homesick Levy Walk での移動軌跡	95
8.7	2つの Home を設定した場合の Homesick Levy Walk の移動軌跡	96
8.8	200 ノード (首輪/ねぐら各 100) 時の 1 日間のノードの移動軌跡	99
8.9	200 ノード (首輪/ねぐら各 100) 時の 14 日間のノードの移動軌跡	99
8.10	100 ノード (首輪/ねぐら各 50) 時の 14 日間のノードの移動軌跡	101
8.11	50 ノード (首輪/ねぐら各 25) 時の 14 日間のノードの移動軌跡	101

8.12	全ノードに占めるデータ保持ノードの増加の数値計算結果	103
8.13	200 ノード (首輪/ねぐら各 100) 時の全ノード間の推定取得接触データ . . .	105
8.14	100 ノード (首輪/ねぐら各 50) 時の全ノード間の推定取得接触データ . . .	106
8.15	50 ノード (首輪/ねぐら各 25) 時の全ノード間の推定取得接触データ	106
8.16	200 ノード (首輪/ねぐら各 100) 時の首輪ノードねぐら基地局間の推定取得 接触データ	108
8.17	100 ノード (首輪/ねぐら各 50) 時の首輪ノードねぐら基地局間の推定取得接 触データ	108
8.18	50 ノード (首輪/ねぐら各 25) 時の首輪ノードねぐら基地局間の推定取得接 触データ	109
8.19	環境特性の積極的推定による設計へのフィードバックの効果	110
8.20	単一機種での規模拡大による機能検証	111
8.21	機種混合での規模拡大による機能検証	111
8.22	QOMET のアーキテクチャ	114
8.23	ノードイメージサイズが 1GB の場合の転送時間	115
8.24	エミュレーションのスクリーンショット	116
8.25	通信機会が多かった北陸先端科学技術大学院大学付近の拡大図	117
8.26	地形情報の有無による DeltaQ の出力結果比較 (遅延)	117
8.27	Node53-93 の実験中の見通し直線と地形の交差	118
8.28	地形情報の有無による Node53-93 間の Ping による遅延時間結果比較 (遅延)	119
9.1	岩本町内山林での機器設置風景	125
9.2	送受信機設置状況	125
9.3	製作した送受信機	126
9.4	2ヶ月間の RSSI と単位時間降雨量	127

表 目 次

3.1	動物の生態，森林の環境のパラメータ	26
3.2	観測システム・デバイスに求められる要件	26
3.3	移動パターン毎の目的地選択確率	31
3.4	移動パターンと車の台数毎の平均遅延とデータ損失率	34
4.1	送/受信機詳細	41
4.2	測定実験条件	41
6.1	Node Agent と Manager Server の機能	56
6.2	実験条件	69
8.1	シミュレーションパラメータ	100
8.2	ノード数とすれ違い回数，接触時間の1日平均	102
8.3	1日に発生するデータ量	104
8.4	各移動モデルでの Node0 の他ノードとの接触回数	107
8.5	通信機会の多かった上位5ペア	116
8.6	通信量上位5ペアの地形考慮の有無による平均通信可能帯域	119

プログラム一覧

7.1	NEED の ROOT エlement例	77
7.2	NEED の NODE エlement例	78
7.3	NEED の Function エlement例	78
7.4	NEED の Parameters エlement例	79
7.5	NEED の Environments エlement例	80

第 1 章

序論

1.1 情報の収集、伝達と共有

人類の歴史は情報の収集、伝達と共有の歴史と言っても過言では無い。有史以来人々は目の前の情報を、五感を通じ収集し、言葉や文字を用いて表現し、人々の交流の中で伝達と共有を行ってきた。電子的な通信技術やコンピュータの登場以前、これらは全て人の手によって行われてきた。科学技術の発展に伴い、人類は情報の収集と共有の範囲を広げ、伝達の上を上げてきた。まず、通信技術の発達と共に、伝達手段が人の手から機械・コンピュータへと置き換わった。そして画像、音声の電子的な記録が可能になったことで、情報収集の機能も人から機械・コンピュータへと置き換わっていった。その結果、人々が得られる情報量は飛躍的に増加した。この歴史を俯瞰すると情報の収集、伝達と共有は人間の根源的な欲求であると言える。

人間に代わり、実世界の情報を自動的に収集し、コンピューティングリソースとして活用する為に、多数のセンサをネットワークを用いて結合し、様々な実環境上のデータを簡便且つ自動的に収集するセンサーネットワークシステムが提案されている。現在このセンサーネットワークシステムは、ファクトリーオートメーションシステム、気象観測ネットワーク、ホームエナジーマネジメントシステムなどの限定的な空間とデバイスにおいて実現されている。工場の生産管理においては、温度や湿度と言った情報を元に、機械の制御を行ったり、荷物に付けたバーコードやRFIDタグにより、通過した荷物の情報を収集し、荷物の仕分けが行われる等広く普及している。気象観測ネットワークとしては、多数の携帯電話基地局に気象センサを取り付ける事で、これまで気象観測所がレーダーなどで行っていた広域の気象情報収集を大規模なセンサーネットワーク群で行っている事例もある [1,2]。

スマートフォンなどのスマートデバイスの普及や、センサとなるデバイスの小型化、多様化と低コスト化によって、この様な多数のセンサから得られた情報を有線・無線ネットワークを介して収集し利活用するセンサーネットワーク技術は、これまでの限定的な空間・デバイスで構成されたホモジニアスなネットワークから、様々な場所、様々なデバイスを用いたヘテロジニアスな環境へと応用分野が飛躍的に広がりはじめている。

1.2 ユビキタスコンピューティングとセンサーネットワーク

パーソナルコンピュータが登場した 1980 年代、東京大学の坂村健らが TRON プロジェクトで全ての物にコンピュータが組み込まれる未来を考え [3]、1990 年代のインターネット創成期にアメリカのゼロックスパロ・アルト研究所の Mark Weiser らがユビキタスコンピューティングを提唱した [4]。あらゆる場所、様々なコンピュータが利用される世界を示したこの概念は、2000 年代に入って提唱されたユビキタスネットワークの概念へと昇華し、ネットワーク技術の進化、デバイスの進化と共に急速にユビキタス社会実現へと向かっている様にとらえられている。しかし、Mark Weiser はユビキタスコンピューティングの本質はカームコンピューティング (calm computing) であると表現し、ユーザによる明示的なデータ入力を求めない“さりげない”コンピューティングであるとしている。また、坂村健らはコンピュータ同士の高度な協調動作に重点を置いて、ユビキタスコンピューティングを定義している。本質的なユビキタスコンピューティングに視線を戻すと、ユビキタスコンピューティングとは情報の収集を伴う情報処理基盤であると言え、情報の収集、伝達と共有を行う基盤となる。現在のコンピューティング環境は、デバイスの多様性と規模においてユビキタスと言える状況となってきたが、各デバイスはクライアントデバイスで有り、デバイス同士の相互連携・情報収集と処理においては、ユビキタスコンピューティング環境とは言い難い。各デバイスが持つセンサー機能、通信機能を活用することで得られると期待される情報は膨大であり、これらを利活用した真のユビキタスコンピューティング環境の実現が望まれる。すなわち、ユビキタスコンピューティングの課題とはこの様な様々なデバイス、環境で利用可能なセンサーネットワーク技術との融合である。

1.3 センサーネットワークの設計・検証における課題

センサーネットワーク技術と、センサーデバイス・通信デバイスの小型化、多様化によって、これまで情報収集が困難であった、山岳・海洋地域などの地理的極限地域での環境や動物の生態観測や、無数のモバイルデバイスによる群衆の移動情報収集など、新たなセンサーネットワークアプリケーションが提案されている。

このような環境で情報を収集するネットワーク技術として、アドホックネットワークや、すれ違い通信等の遅延耐性ネットワーク (DTN:Delay/Disruption/Disconnection Tolerant Networking) が提案されている。しかし、アドホックネットワークや遅延耐性ネットワークの構築に必要な技術や最適な設計は、アプリケーションの要求、ノードの移動特性による通信環境の変化や不連続性の時間粒度によって異なり、汎用的な設計は望めない。

一般に、ネットワークシステムの開発は図 1.1 の様なライフサイクルを構成する。アドホックネットワークや遅延耐性ネットワークを利用した、センサーネットワークの開発では、それぞれのアプリケーションにおいて想定される、センサーネットワークの展開対象エリアの広さや地形、ノードの密度、移動特性システム等の特性を設計・検証の各段階で検討し、デバイスの選定、ソフトウェアの仕様定義、技術検証を行う必要が有る。しかし、アドホックネットワークや遅延耐性ネットワークで提案されているネットワーク的極限環境での情報収集を行う場合、ノードのモビリティの制御や十分なノード密度を確保出来ない場合が存在する。さらに、環境やノードの移動が推定可能であっても、一般的な技術の普及期、またはネットワークの構築開始当初は、十分なノード数が無く、徐々にノードが過密になっていく場合も考えられる。その為、この様な環境を想定した新しいセンサーネットワークシステムを提案する場合、これらの環境特性や技術の過渡期・普及期における条件など、様々な影響の検討が必要であるが、実現する為の技術要素の選択は困難である。その結果、提案したシステムの実現性の提示する事も困難となる。

一般的に、新しいネットワーク技術の提案は、汎用的な利用想定に基づいて提案される。その為、センサーネットワークシステムの設計者はどの技術やデバイスの組み合わせ、各種パラメータを適用すれば要求を実現するシステムが構築出来るのか判断する必要がある。そこで、システム設計者は先ず簡単な概念検証を行ったり、プロトタイプを作成して実験を行う等の方法で、徐々に仕様を固めシステムの設計と実装を行っていく。しかし、アドホックネットワークや遅延耐性ネットワークの利用を想定する様なシステムの論理的検証は、システム的环境、特に地形やノードの移動、デバイスの電波伝播特性と言った物

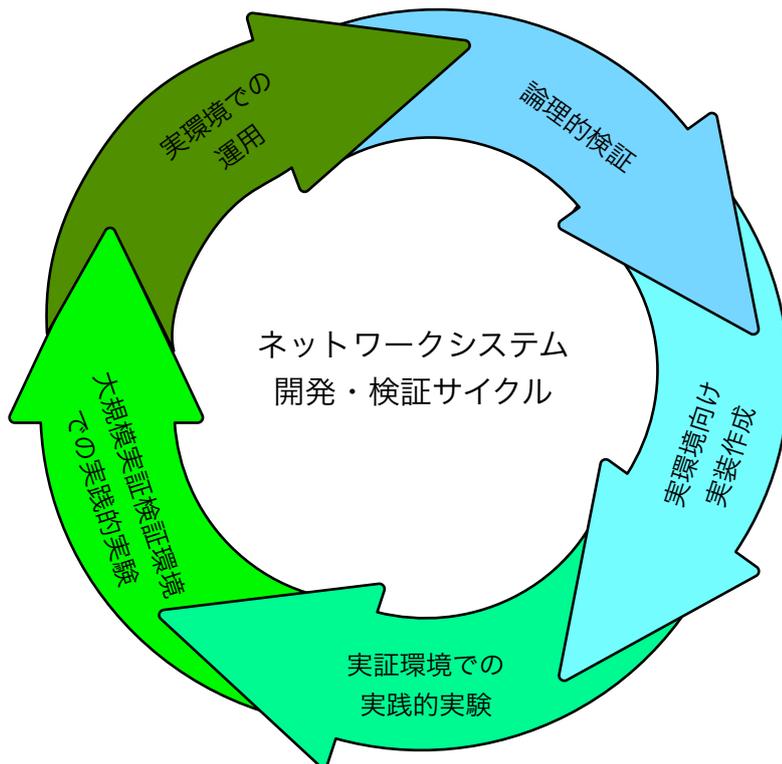


図 1.1: センサーネットワークシステムの開発・検証サイクル

ネットワークシステムの開発・検証は、新規技術の提案の論理的検証から始まり、実環境向けの実装の作成、実証環境での実践的実験、大規模実証検証環境での実践的実証実験を経て、実環境に展開・運用される。そして、運用により得られた知見および新たな機能要求に基づいて、修正や拡張のための論理的検証へと繋がるシステムのライフサイクルを構成する。



図 1.2: 新しいセンサーネットワークシステムの発案と技術要素の選択

新しいセンサーネットワークシステムを提案しても、物理的な特性による影響など、複雑な検討項目が多く、実現する為の技術要素の選択が難しく、またシステムの実現性を示すことが難しい。

理的な影響を受けるため検証のために必要なパラメータが多岐にわたる。また、センサーネットワークを設計する為には、アプリケーションの要求するデータまたはデータマイニング結果としての情報が必要十分である事が求められるが、アドホックネットワークや遅延耐性ネットワークで取得される情報からアプリケーションにとって必要な情報が取得できるか、得られた情報の意味解釈を検証可能か十分に検討する必要がある。このような検証のためのパラメータを行い、観測データとして推定されるリファレンスデータの作成、全体のデータを照合・解析を伴う詳細な論理検証を行う事は困難を極める。

StarBED Project [5] では様々なネットワークシステムの実証検証を行う為の技術開発とサポートを行っている。しかし、実証検証環境で再現される外乱をエミュレーションする為のパラメータの決定は実験者に委ねられている。その為、対象環境が複雑でパラメータの詳細な決定が困難な場合、エミュレーションの妥当性や検証結果の有効性を判断する事が困難となる。また、環境特性の推定を行い、開発工程の手戻りを減少させても、ソフトウェアの検証におけるバグや想定外の問題を根絶する事は難しい。また、システムに必要なパラメータの細かな調整はソフトウェア実装後の検証段階で行われる事が多い。その為、実証検証環境において、規模やパラメータを変えた複数回の実験を行う事となる。また、前節までに述べたようなユビキタス環境においては、ヘテロジニアスなノード・ネットワーク構成になる為、様々な規模やノードの組み合わせによる実証実験も必要となってくる。その為、アプリケーションの検証を容易に行うには、実証検証環境の規模を段階的に拡張し、効率的・再帰的実験を行う実証環境構築のフレームワークや実験支援ソフトウェアが必要となる。

これらのネットワーク的極限環境を観測するアプリケーションの開発を行う為には図 1.3 に示すような各アプリケーションに対して、物理特性をパラメタライズし、ネットワークシステムの開発と検証を行う特定アプリケーション向けの開発・検証サイクルが必要となる。本論文では、このようなネットワーク的極限環境で利用するアプリケーション向けワイヤレスセンサーネットワークの開発を行う為の開発と検証の手法について議論する。そして、特定アプリケーション向けセンサーネットワークシステムの開発検証サイクルを実現する実証検証環境を用いた開発・検証フレームワークを提唱する。

1.4 本論文の構成

本論文は以下の 4 部と序論・結論から構成される。

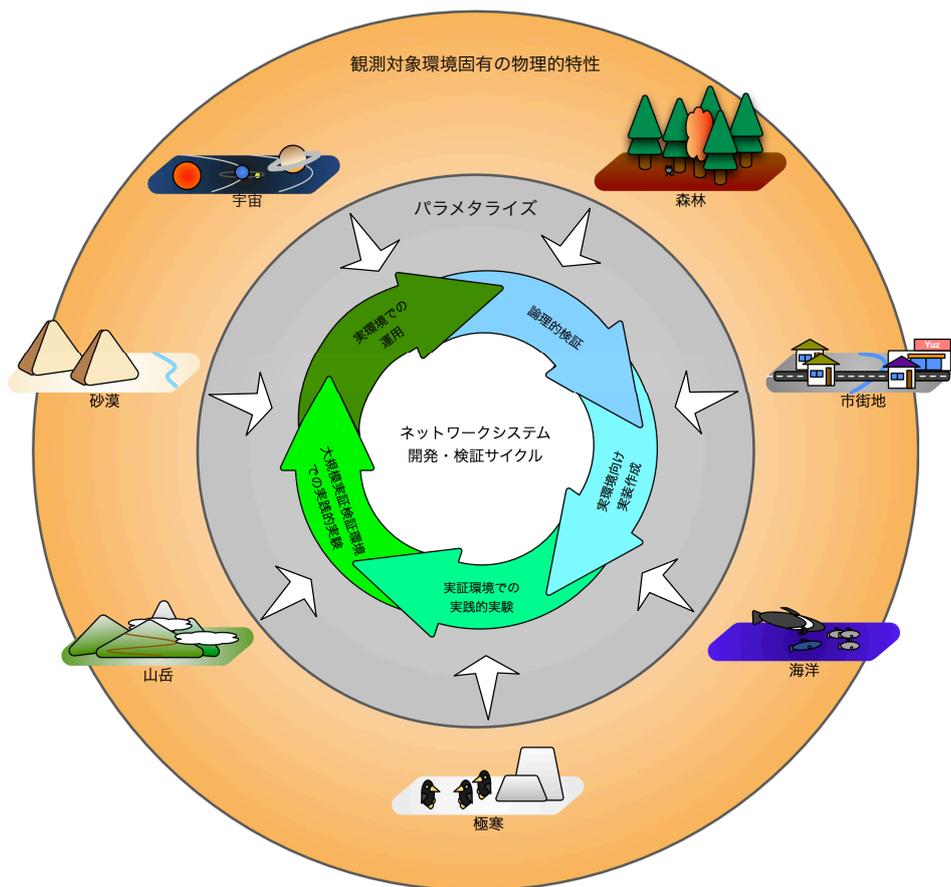


図 1.3: アプリケーションに特化し、物理特性を考慮したセンサーネットワークシステムの開発・検証サイクル

システムのライフサイクルにおける各段階で、アプリケーションが想定する環境（空間情報）やノードの移動特性等の物理特性を再現しライフサイクルの各段階における作業を最適化する。

- 第Ⅰ部 センサーネットワークの現状と課題
- 第Ⅱ部 対象環境の物理特性の積極的推定とデータ解析
- 第Ⅲ部 対象環境の物理特性を考慮した大規模実証検証環境
- 第Ⅳ部 特定アプリケーション向けセンサーネットワーク開発検証手法

第Ⅰ部では、センサーネットワークを取り巻く環境、設計における課題を述べる。第Ⅱ部では、各アプリケーション毎の観測対象環境固有の物理特性の推定の必要性を述べ、積極的に推定、パラメタライズする手法を議論する。また、アドホックネットワークや、遅延耐性ネットワークを用いたセンサーネットワークで取得されるデータの推定と解析手法についても議論する。第Ⅲ部では、対象環境の物理特性を考慮した大規模実証実験環境に求められる要件と課題について議論し、提案する実証検証環境構築手法と実装、評価を行う。第Ⅳ部では、本論文で提唱する特定アプリケーション向けセンサーネットワークの開発検証手法を述べる。

第I部

センサーネットワークの現状と課題

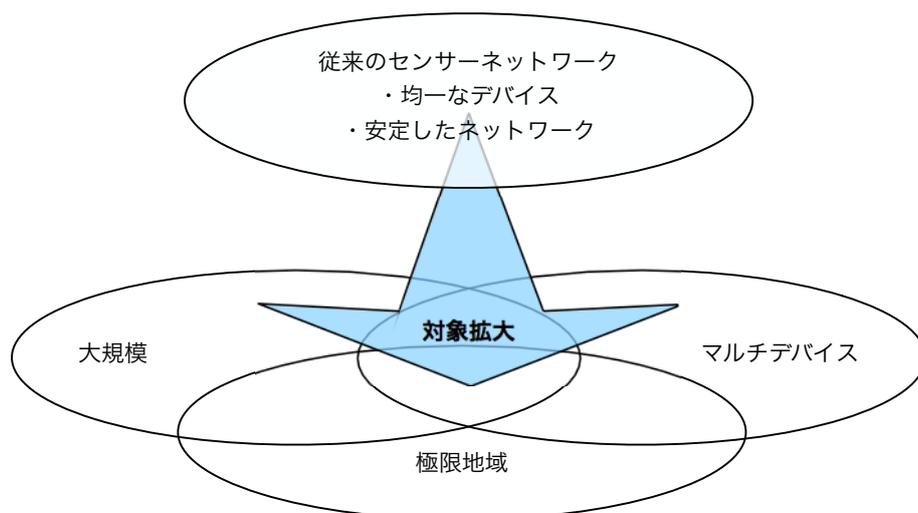


図 2.1: センサーネットワークの拡大

均一なデバイスや安定したネットワークで構成されていたシステムから、スマートフォン等のモバイルデバイスや小型で安価な IoT(Internet of Things) デバイスによる規模の拡大、多様なデバイスを受容したマルチデバイス化、山岳・海洋地域などの地理的・ネットワーク的極限地域でのネットワーク構築等、センサーネットワークが活用される対象領域は急激に拡大している。

第 2 章

センサーネットワークの発展

2.1 固定センサーノードによるセンサーネットワーク

センサーネットワークは、多数のセンサーを無線ネットワークを用いてネットワーク化する事により、様々な実環境上のデータを簡便且つ自動的に収集するネットワークシステ

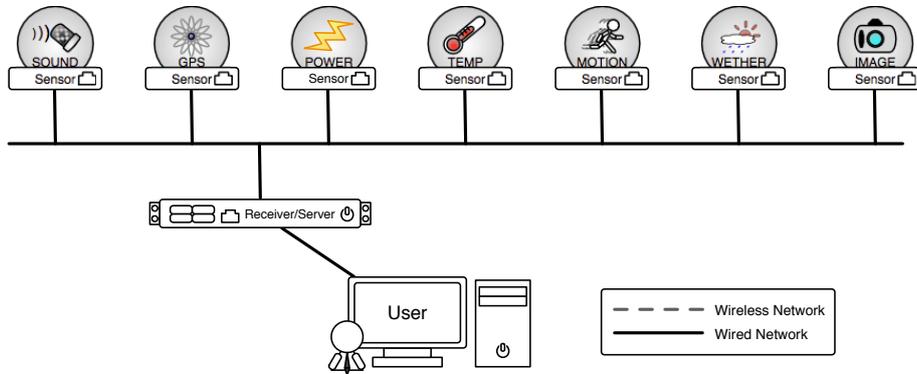


図 2.2: 有線によるセンサーネットワーク

有線ネットワークを用いたセンサーネットワークは全てのセンサーは有線による安定したネットワークを構成している。

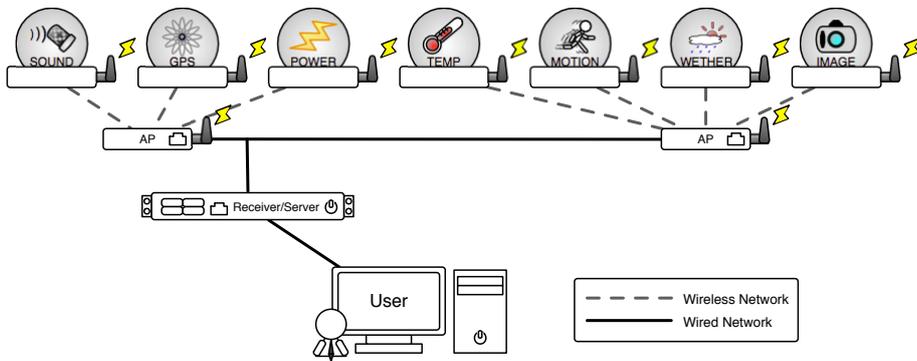


図 2.3: 有線の置き換えとしてのワイヤレスセンサーネットワーク

有線の置き換えとしてのワイヤレスネットワークを用いたワイヤレスセンサーネットワークはセンサーに繋がるラスト・ワン・ホップを無線化する。設置の自由度が向上するが、有線によるコアネットワークが存在する。持続的なワイヤレスネットワークを利用し、安定したネットワークを構成する。

ムである。センサーネットワークは図 2.1 の様にデバイスの多様性、規模、対象とする環境を広げ、急速にその適用範囲を広げている。初期のセンサーネットワークは、図 2.2 の様な各センサーデバイスを有線ネットワークで接続し、情報を収集するサーバが情報を取得する方式であった。携帯電話や WiFi と呼ばれた無線通信が普及するにつれて、末端の有線通信区間を無線通信に置き換えた図 2.3 の様なワイヤレスセンサーネットワークが利用されるようになった。無線通信を利用する事で有線ネットワークの敷設のコストが不要となり、設置面での柔軟性が向上した。この様なワイヤレスセンサーネットワークは、工場などのファクトリーオートメーションシステムその他、自動販売機や電器・ガス等の各種テレメータ、ホームエナジーマネージメントシステム (HEMS) やスマートハウスがあり、固定したセンサーを利用するシステムに応用されている。

2.2 移動センサーノードによるセンサーネットワーク

近年無線通信の機能を持つデバイス、センシング機能を持つデバイスの小型化、低価格化が飛躍的に向上したことにより、センサーデバイスの設置・ネットワークの敷設を簡便にする為に有線ネットワークをワイヤレスネットワークに置き換えた固定系ワイヤレスセンサーネットワークが利用されるようになった。さらに、これまでは実現が困難であった、有線によるコアネットワークを持たない、さらには移動するセンサーデバイスも利用したモバイルワイヤレスセンサーネットワークへと活躍の場所が広がっている。

2.2.1 アドホックセンサーネットワーク

アドホックセンサーネットワークはアメリカ U.C.Berkeley で行われている SMART DUST Project [6] によって提案された技術で、図 2.4 の様な無線通信によってセンサーデバイス同士で自立的にネットワークを構成し、情報伝送を行うセンサーネットワークである。これまでサーバに直接通信を行う事で収集していた情報を、デバイスどうして情報する事で、全く有線ネットワークのインフラを持たずに、簡便かつ短時間でのネットワーク構築が可能である。アドホックネットワークは、ネットワークインフラを持たない環境でネットワークを構築出来るが、その特性上、ネットワークが比較的安定していると共に、センサーデバイスは固定または低い移動性を想定する 경우가多く、また常時接続環境の構築を目的としているため、無線通信の到達可能な範囲でのノード密度が必要となる。

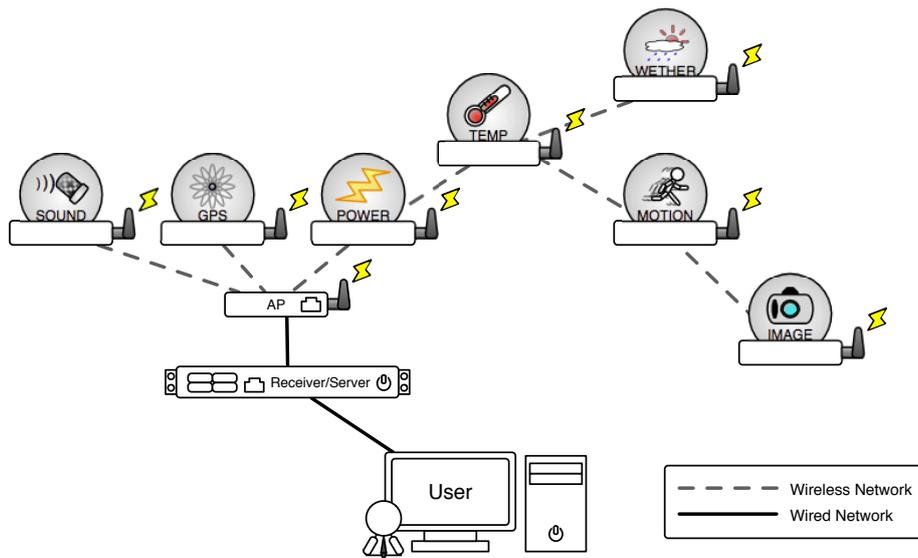


図 2.4: アドホックセンサーネットワーク

無線通信機能を有するデバイス同士が、自律的にネットワークを構成する。有線ネットワークのインフラを持たず、デバイスを適当に配置するだけでセンサーネットワークを構築する事が可能。

2.2.2 DTN センサーネットワーク

ユビキタスな環境の実現が現実味を帯びると共に、自動車の渋滞情報収集や山岳・海洋地域などの極限地域での環境情報収集や、野生動物などの生態情報の収集等、安定的な通信を望めないネットワーク的な極限地域でのネットワーク構築が注目されてきた。このような安定的な通信が困難な環境でネットワークを構築する為に、図 2.5 の様な遅延耐性ネットワーク (DTN: Delay/Disruption/Disconnection Tolerant Networking) [7, 8] を利用した情報収集・転送技術が提案されている。DTN は惑星間通信の研究に端を発する技術で、TCP の標準的なタイムアウト時間である 300 秒を遙かに超えるような長い遅延時間や、計画的または非計画的な通信リンクの断絶など、従来では想定されていなかったような劣悪な条件で通信を行う為の技術である。

2.2.3 未知の環境特性を含むセンサーネットワーク

DTN を用いた通信の研究、利用形態の提案では、これまでノードの移動、情報の収集方法において制御可能または、ある程度十分な通信を確保できる密度でノードが存在する環

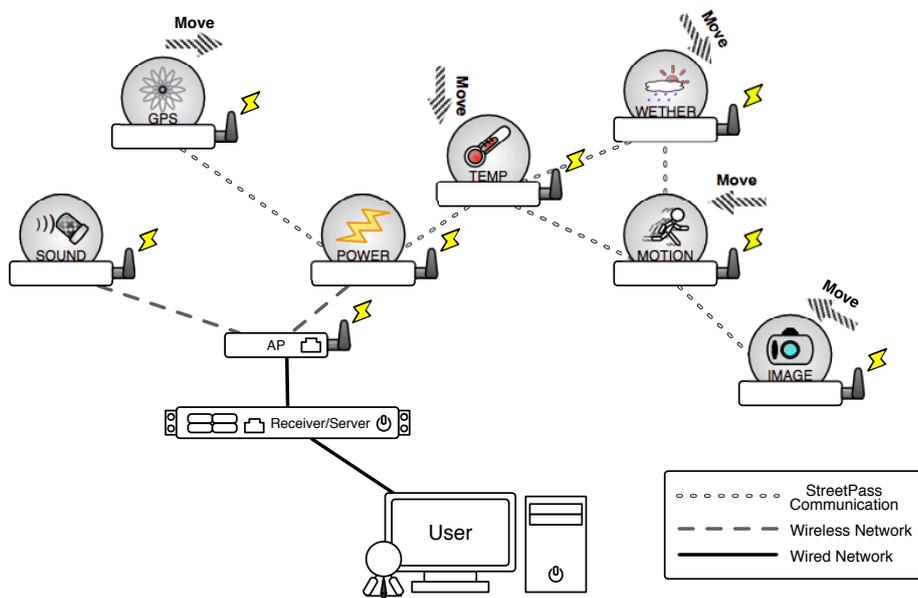


図 2.5: DTN センサーネットワーク

アドホックネットワークと同様な無線機能を有するデバイス同士による自立的なネットワーク構成に加え、計画的またはひげ威嚇的な通信リンクの断絶など、劣悪な条件でのネットワーク構築を行う。

境を想定する事が多かった。しかし、山岳・海洋地域において、低コストに情報収集を行う為にネットワーク極限地域での情報収集手段としてDTNを利用する場合、ノードのモビリティの制御や十分なノード密度を確保出来ない場合が存在する。また、環境やノードの移動が推定可能であっても、一般的な技術の普及期、またはネットワークの構築開始当初は、十分なノード数が無く、徐々にノードが過密になっていく場合も考えられる。その為、DTNの構築に必要な技術や最適な設計は、アプリケーションの要求、ノードの移動特性による通信環境の変化や不連続性の時間粒度によって異なり、汎用的な設計は望めない。このような通信環境、ノードの移動特性、ノード密度の把握が十分でない環境でネットワークを設計・利用する為にも、アプリケーションの要求を踏まえて、未知の特性を含む環境で利用するネットワークの設計指針となる設計手法が必要となる。

センサーネットワークでは、センシング・転送ソフトウェアの品質が、得られるデータの情報品質に大きく作用する。また、DTNを用いたセンサーネットワークでは、ネットワークの設計段階からの設計の検証環境が必要となってくる。その為、設計段階における、概念検証とネットワークを構成するソフトウェア群の検証が重要となる。しかし、実機を用いて実環境で大規模な実証検証を行うことは現実的ではない。本論文で議論する未知の特性を含む環境を対象とするセンサーネットワークシステムを開発する場合には、概念検証から始まるシステムの開発・検証ステップ、ライフサイクルの中で、特に概念検証と収集データからのフィードバックにおいて、如何にしてそれら未知の特性を再現し、論理的検証、実装、実証実験時の各段階の結果へ反映させて行くかが重要となってくる。

2.3 断続的通信環境下で観測を行うセンサーネットワークの課題

2.3.1 断続的通信環境下で観測を行うセンサーネットワーク設計

現在急速に普及しているコンピューティングデバイス群が即ユビキタスコンピューティング環境を実現する情報収集・伝播基盤とならない理由は、そのデバイスの多様性やネットワークに無線通信が主体となって来た事、管理主体が単一で無く、動作を制御出来ないなど、センサーネットワークとしてシステムを検討する上での要素に非決定的要素が多いことが上げられる。これは、これまで単一の管理主体による限定的な空間・デバイスで構成されていたセンサーネットワークを多様で多数のデバイスを利用して、ネットワーク的、

時空間的に自由度が高く、また管理主体すら複数にまたがるデバイスクラスタを利用して構成しようとした場合、様々な未知の特性を持つ動作環境が想定される為である。

ユビキタスな環境の実現を想定して研究提案が行われているセンサーネットワークの1つに車車間通信ネットワーク等のモバイルデバイスによるすれ違い通信ネットワークがある。この技術は、車やモバイルデバイスに搭載したセンサと通信機器を用いて渋滞等の交通情報や危険情報の共有、災害時の通信路としての活用を目的としている。しかし、自動車やモバイルデバイスに搭載されたセンサと通信機器を用いる場合、センシングとそのデータを活用する側とで管理主体が異なるため、デバイスの普及、機能の死活や移動経路などにより期待するデータ、通信路が確保出来るか概念レベルでは判断できない。また、無線による通信となるため、移動する車車間の通信がどの程度効率的に機能するかも判断が難しい。その結果、構想から技術検証に至る各開発・検証の過程で未知の特性による外乱を多く含む事となり、その未知の特性が提案されたシステムの実効性、有効性を示す上での障害となる。

この他に、近年の通信デバイスの小型化によって、山岳地や海洋地域など安定したネットワークインフラを期待出来ない地理的、ネットワーク的な極限地域で、環境観測や動物の生態観測、ノードのトラッキングなどを行うセンサーネットワークも提案されている。しかし、地理的、ネットワーク的な極限地域での通信は、ノードの移動が制御出来なかったり、ワイヤレス通信の電波状況が地理的な影響を受けやすい等概念提案時には想定が困難な未知の特性を含む。

センサーネットワークは、アプリケーションの要求により、そのシステムのネットワークの品質も含めた設計が大きく異なる。新規にデバイスを作成・展開する場合、既存のデバイス群から情報を収集する場合でも、目的の情報を収集するシステムを構築する為には、そのデバイスの特性や環境による外乱、システムによる内乱など、様々な特性を想定、検討する必要がある。しかし、ファクトリーオートメーションシステム等の、限定的な空間とデバイスによって作られていたセンサーネットワークシステムと比較し、このようなユビキタスな環境で構築されるセンサーネットワークは、ネットワークの品質、環境による通信への外乱など事前想定が困難な要素が多く、これら未知の特性を持つ環境で動作するネットワークを如何に設計して行くかが課題となる。

2.3.2 センサーネットワークの設計とデータ解析

DTN を用いたワイヤレスセンサーネットワーク、とりわけすれ違い自体が重要な意味を持つネットワークを設計・利用するためには、ノードの密度や移動モデルと行った様々な要素を検討する必要がある。ノードの密度が十分に高い場合は、得られる情報も多いが、ノードの密度が低い場合は、期待するデータが得られるのかどうか、事前に十分に検討する必要がある。DTN の最も一般的な設計はメッセージフェリーと呼ばれる、移動制御可能なノードを利用して、対象域に設置してあるセンサーノードのデータを収集して回る方法である。これは、管理された観測エリアの中でデータを効率的に収集できる反面、メッセージフェリーノードを全センサのデータが取得可能な領域まで巡回させる必要がある。これまで市街地でバスなどを利用したデータ収集の手法や、サバンナで動物に付けたセンサの情報を車で巡回して取得する研究などが提案されているが、前者は本質的には DTN である必然性は低く、後者は人が効率的に巡回可能な領域である事が条件となる。

人が効率的に巡回出来ない地理的な極限地域で情報を収集する際には、効率的なメッセージフェリーが望めない状況で情報を収集する事となる。このような条件下で情報を収集するセンサーネットワークでは、ネットワークの設計、センシング・転送ソフトウェアの品質が、得られるデータの情報品質に大きく作用する。デバイスの密度が低い場合の DTN はノードの移動や、電波の伝播距離、地形、デバイスの役割など多彩な条件の検討が必要となり、同時にそれらを踏まえた情報解析手法が必要となる。

2.3.3 ワイヤレスネットワーク実証検証環境

センサーネットワークでは、センシング・転送ソフトウェアの品質が、得られるデータの情報品質に大きく作用する。また、アドホックネットワークや DTN を用いたセンサーネットワークでは、ネットワークの実証検証環境が必要となってくる。しかし、実機を用いて実環境で大規模な実証検証を行うことは現実的ではない。これまで、ネットワークを用いたシステムの設計、検証、実運用までは図 2.6 の様な順序で開発が行われてきた。

しかし、小規模な実証環境では顕在化しない問題も多く、実環境にシステムを導入した際に、実証環境では再現されていなかった外乱などの影響を受け、期待した性能を発揮出来ない場合がある。その為、StarBED Project [5] では、大規模実証検証環境を利用した図 2.7 の検証ステップを提案している [9]。この様な大規模ネットワークシステムの実践的検証を行うために、小規模な実証環境で実装に関する最低限の検証後、実環境に近い大規

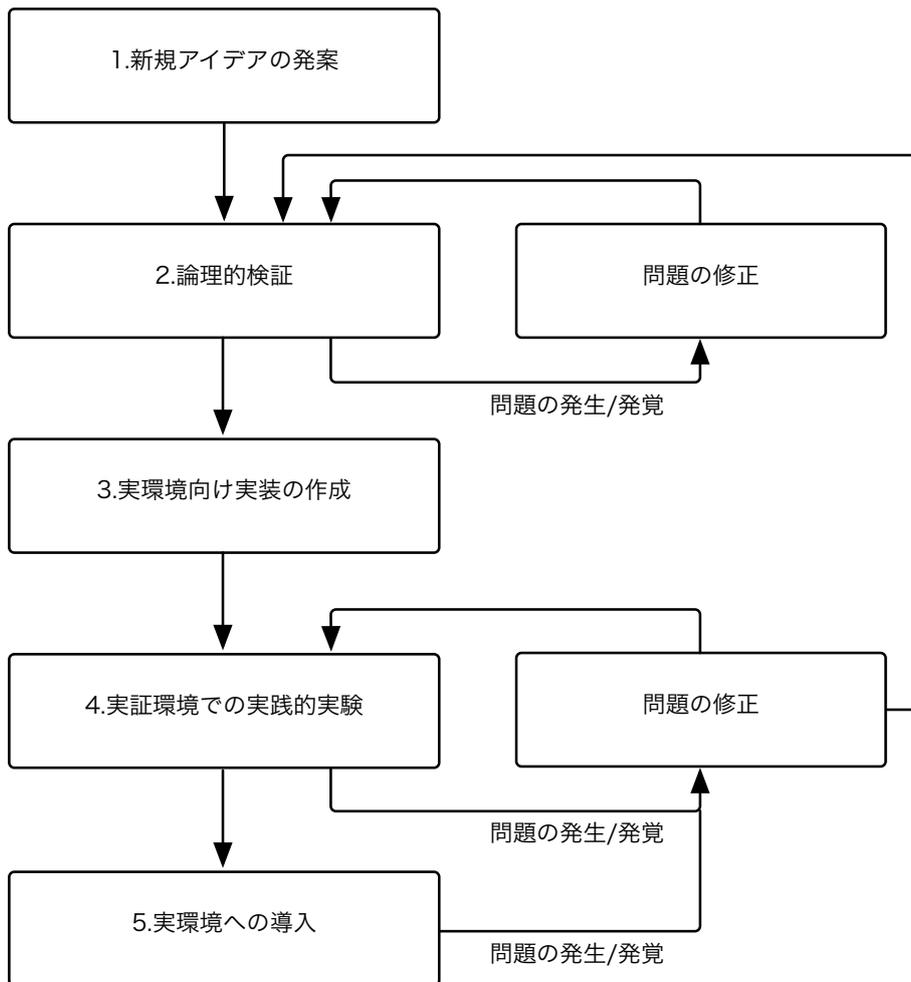


図 2.6: 一般的な技術開発工程

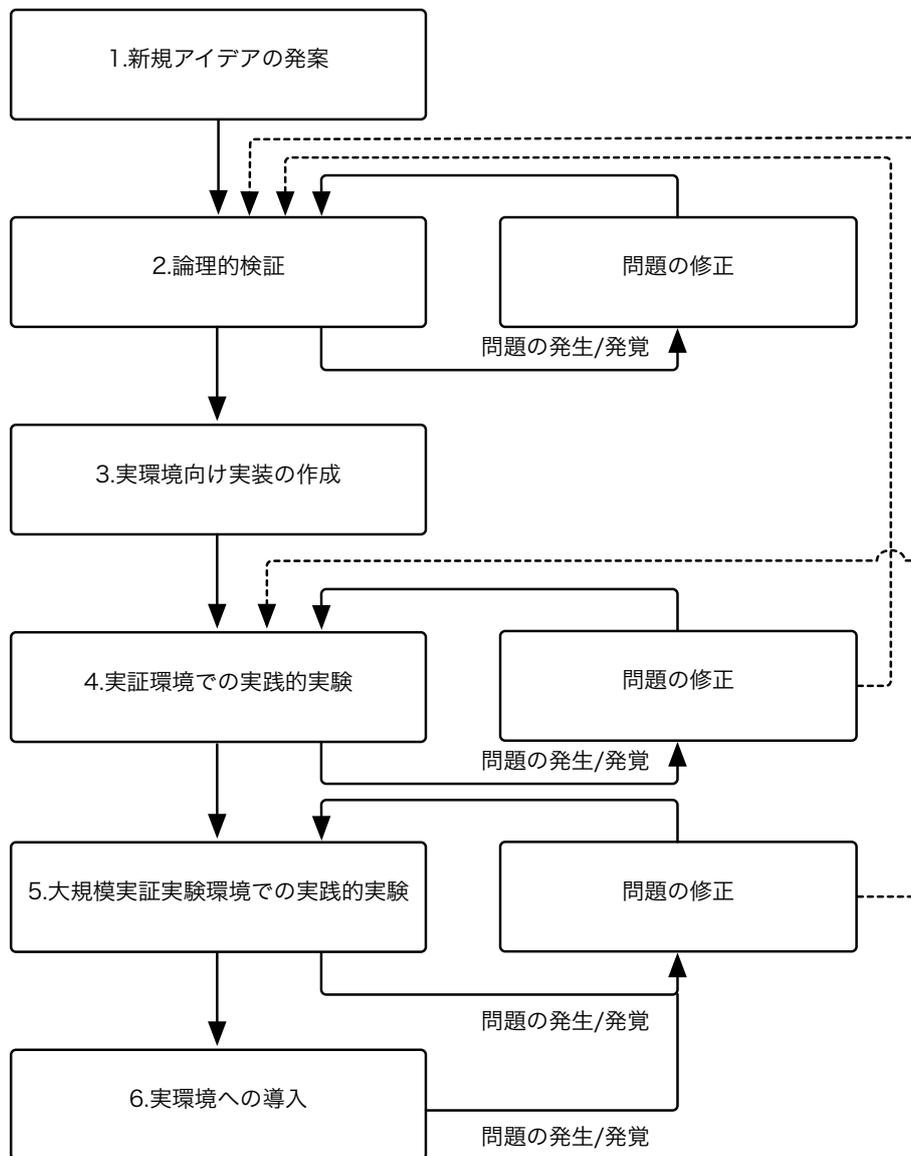


図 2.7: StarBED の提案する技術開発工程

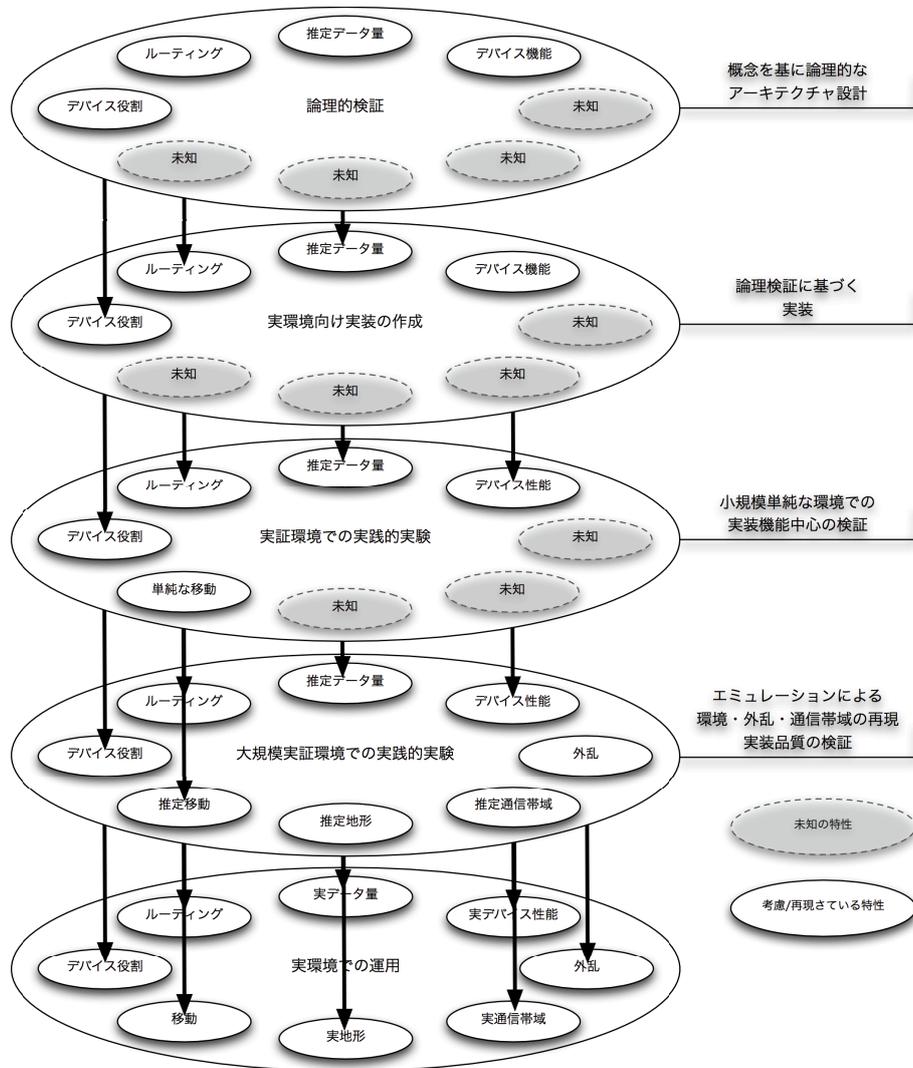


図 2.8: 開発ステップと考慮される特性

模な実証検証環境で実験を行い、規模や多様性などに起因する問題を洗い出す事が望ましい。しかし、注目すべき点は、どちらの開発ステップにも開発の段階を跨ぐ問題の修正、すなわち手戻りが有る事である。特に図 2.6 および図 2.7 内で点線で示した、論理的検証への手戻りは、論理的検証段階で十分な検討が行われていなかった結果、実装・実験の段階で解決できない問題が生じた為に発生する手戻りである。本論文で議論する未知の特性を含む環境を対象とするセンサーネットワークシステムを開発する場合には、概念検証から始まるシステムの開発・検証ステップ、ライフサイクルの中で、特に概念検証と収集データからのフィードバックにおいて、如何にしてそれら未知の特性を再現し、論理的検証、実装、実証実験時の各段階の結果へ反映させて行くかが重要となってくる。図 2.8 に大規模実証検証環境を利用したシステムのライフサイクルと考慮・再現されるシステム・環境の特性の関係を示す。概念検証から、大規模実証実験に至る開発ステップの中で早い段階で想定される特性を増やすほど、各段階での問題の発生、発見を早め、且つ手戻りによる開発コストを低減できる。

ネットワーク実証検証環境である StarBED [5,10] では IEEE 802.15.4 等の通信エミュレーション、ユビキタス環境エミュレーション技術を組み合わせたモバイルワイヤレスネットワークエミュレーションテストベット QOMB の研究を行っている [11,12]。QOMB は電波環境をエミュレーションするモジュールや通信デバイスのエミュレーションを行うモジュールなどから構成されている。そして、ノードの移動モデルやノイズ源、障害物をパラメータとして与えることでユビキタス環境での大規模なネットワークエミュレーションを行うことが可能である。QOMB を利用する事で、実証実験におけるネットワークエミュレーションにおいて、ネットワークの挙動による外乱を模倣し、検証段階においてソフトウェアの詳細な挙動の検証を行う事が出来る。道路を定義したノードの移動エミュレーションも可能であり、獣道や川沿い等の動物の移動ルートや登山道等を道路と見立てた応用も可能である。しかし、現状のシステムは人間が持ち歩くデバイスや車載デバイスを想定した市街地や屋内での環境再現を想定して設計されており、本研究で必要とする地形やアンテナ地上高による電波伝播の変化など複雑な地形を有する様なネットワーク的極限環境の再現に必要なパラメータを再現する機能はない。電波伝播モデルは、これまでに多くの研究がなされている。それらの既知の電波伝搬モデルを応用し、地形情報を組み合わせてシミュレーションすることで複雑な地形を有する環境での通信エミュレーションにも対応可能なワイヤレスネットワークエミュレーション技術を開発する必要がある。

しかし、エミュレーションの為の模倣技術だけでは改善されない問題もある。QOMB で

は、The One Simulator [13] や OpenStreetMap [14] を組み合わせて道路などの地形に沿ったノードの移動もエミュレーション可能であるが、実データに基づく移動データが無い場合、これらのデータは何らかの推定に基づくシミュレーションによる再現となる。ワイヤレスセンサーネットワーク、特にDTNの様なすれ違いによる通信を伴う場合、採用する移動モデルによって通信の結果は大きく異なってくる。その為、採用する移動モデル、そしてモデルに設定する各パラメータが検証の妥当性に大きな影響を与える。

第II部

対象環境の物理特性の積極的推定とデータ 解析

第 3 章

未知の特性を含む環境を対象とするセンサーネットワークの検討事例

本章では、アプリケーションに特化したセンサーネットワークシステムを開発する各設計・検証段階で、特に重要となる環境特性を明らかにする為、アプリケーションが対象とする環境の情報が十分に無い、すなわち未知の特性を含む環境を利用・観測するネットワークシステムの検討事例を基に設計・検証における課題を議論する。本論文では未知の特性を含むネットワークシステムの検討事例として野生動物の観測を行うネットワークと、渋滞や事故情報の共有、災害時の情報収集を行う車車間通信ネットワークの概要を説明し、未知の特性を含む環境で機能するネットワークシステムを設計する為に考慮すべき環境特性とその技術課題を述べる。

3.1 動物観測ネットワーク

WAOC Project [15] では、長期生態・環境観測を実現する為に動物指向クラウドネットワークを提唱し、図 3.1 のような首輪型のデバイスとねぐら近辺に設置する基地局デバイスによって構成される DTN を用いたネットワーク (首輪 + ねぐらネットワーク) を提案している。これまでの、ウェアラブルデバイスを用いた計測では、デバイスを回収することでデータを取得していた為、人的コストも高く、まだ同一個体の長期観測を行うためにデバイスの寿命を延ばした場合、データ取得までの遅延時間が大きくなる。また故障などでデバイスが消失した場合に、故障する迄のデータも取得出来なくなる。DTN を用いることで、デバイスを長寿命化し長期観測を行った場合でもデータの取得率を上げる事が可能と

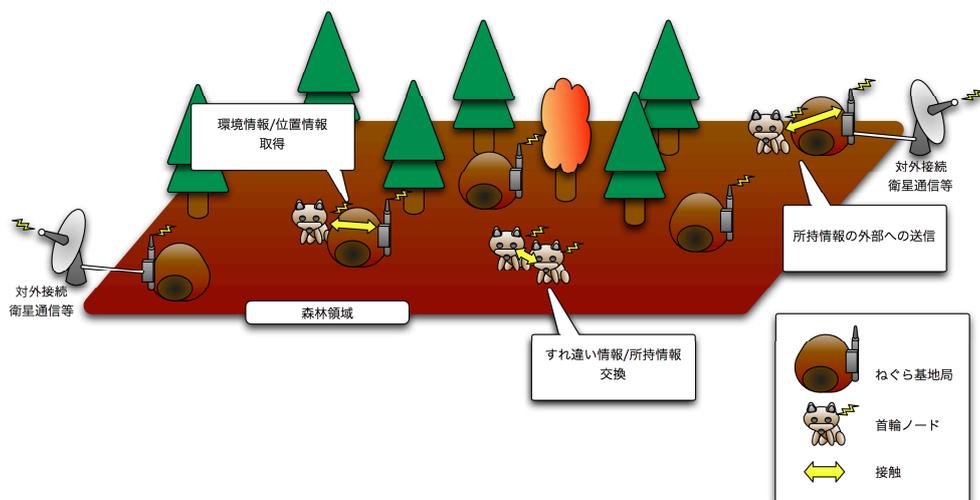


図 3.1: 首輪 + ねぐらネットワーク

森林地域において、動物の首輪に装着する首輪型デバイス(首輪ノード)とねぐら付近に設置するねぐら基地局とで構成されるネットワーク。森林領域の外縁部にあるねぐら基地局は対外接続を持ち、外部へデータを送信するゲートウェイの役割を持つ。ねぐら基地局は環境情報の収集と位置基準局としての役割を持ち、首輪ノードはねぐら基地局に接近(接触)する際に、環境情報と自身の位置情報を取得する。首輪ノード同士が接触した場合は、相互に保持している情報を交換すると共に、接触継続時間などの情報を計測し交換する。

なり、またデータの取得までの遅延時間も短縮する事が出来る。

本研究の対象動物としては、アライグマ、タヌキなどのねぐらを持つ動物を想定している。動物の生活拠点付近に固定的に設置し、環境情報のモニタリングとデータの送信・転送を行うネットワークセンサデバイスをねぐら基地局と呼ぶ。ねぐら基地局は環境情報を取得するセンサとGPSを用いた位置基準局の機能を持たせる。観測対象となる森林領域と外界の境界域にあるねぐら基地局には外部ネットワークへの定常的な接続性を確保し、森林内部に設置するねぐら基地局は外部への定常的なネットワークは確保しない。また、動物に装着し動物の移動・接触のモニタリングとねぐら基地局の観測データを含めた、データの送信・転送を行うデバイスを首輪ノードと呼ぶ。そして、ねぐら基地局と首輪ノードで構成されるネットワークを首輪 + ねぐらネットワークと呼ぶ。

動物の生態や森林の環境情報を解析するために一般的には表 3.1 のような情報を収集、解析する。様々な解析を行うためには、多様な情報の取得が必要であるが、食性や、直接

表 3.1: 動物の生態、森林の環境のパラメータ

環境情報	気温、湿度、気圧、風向、風量、 雨量、日射量、放射線量、 大気物質濃度、etc.
動物の生態	移動パターン、生体同士の接触、 繁殖活動、生存情報、食性、etc.

表 3.2: 観測システム・デバイスに求められる要件

	技術要件
デバイス	データの送受信機能 小型軽量 低消費電力
ネットワーク	耐遅延特性 最適なルーティング

的な死因などの情報は、糞尿などの収集や解剖を行う必要があり、センサ収集は困難である。一方で、環境情報全般や、動物の移動パターン、接触（すれ違い）、繁殖活動、生存情報などはセンサーによる取得、推定が可能であると考えられる。本研究では、各種センサで取得可能な情報と、センサの組み合わせや条件から、推定可能であると考えられる情報をネットワークの設計に取り込み、より多くの情報の収集を可能とするセンサーネットワークを目指す。

まず、センサと取得した情報を伝送するデバイスとネットワークには表 3.2 のような技術要件が必要となる。環境情報は、位置情報とともに定点観測する事で取得可能であるため、センサを立ち木などに固定して設置可能である。そのため、重量の制限は厳しくなくソーラーパネルによる電源供給なども行えるため、消費電力に対する制限も緩く設定できる。それに対し生態情報はそれぞれの個体を継続して観測する必要があるため、動物に首輪などで固定して観測する必要がある。動物ウェアラブルデバイスは、装着している動物

へ影響を与えないため、首輪に装着するデバイスの総重量は動物の体重の4~5%以下である必要があるとされている [16]。また、バッテリー・機器の交換のために対象動物を繰り返し捕獲する事も、捕獲作業のコストや動物への影響から好ましくない。そのため、動物に装着するセンサとネットワークデバイスは小型軽量、長寿命であることが要求される。

一般的に動物の生態観測で利用されている、FM 発信器は、発信機能のみの単機能デバイスであるため、外部観測的に対象動物の位置情報を取得することができるが、対象動物同士の接触、繁殖活動を推定することは困難である。しかし、動物に装着するウェアラブルデバイスに送受信機能を付加する事で、対象動物同士の接触情報をウェアラブルデバイス同士で自律的に取得する事が可能となる。ねぐらを持つ動物はねぐらや餌場などの拠点を中心とした生活圏を持つ。そのため、環境情報を取得する固定センサは、対象動物の拠点の近くに設置することで動物との接近頻度が高くなり、同時に環境情報の移動性が高くなると考えられる。この首輪+ねぐらネットワークによって広範囲なエリアで低コスト且つ多彩な情報を取得可能な動物指向クラウドネットワークの構築が可能であると考えられる。

3.2 動物観測ネットワーク設計における技術課題

3.2.1 動物の行動モデルと間欠通信の時間間隔

IEEE802.15.4 デバイスの多くは、間欠通信を行い、通信を行わない間はモジュールを Sleep することで消費電力を低減する機能を持っている。電力消費を抑えデバイスの駆動時間を延ばすためには、Sleep 時間を長く設定すれば良いが、デバイスを Sleep している間は、データの送信だけでなく受信も不能になり、必要な情報が取得できなくなる可能性があるため、必要な情報の特性に合わせて Sleep 時間、Wake 時間を調整する必要がある。

動物をキャリアとした DTN を設計する場合、首輪ノード・ねぐら基地局同士の接近時に通信が可能となるようにするためには、Sleep 間隔を生態観測対象またはキャリアとなる動物によって調整する必要がある。ノード同士が情報を送受信可能な距離を d 、相対的な移動速度を s 、Sleep 時間 T_s とした場合、互いのノードが送受信が可能な円を原点で交差する場合でも式 3.1 の関係でなければノードが Sleep 中に通信可能域を通過してしまうため、通信が成立する可能性が低くなる。

$$T_s < \frac{d}{s} \quad (3.1)$$

また、動物の生態を観測するためには、動物の生態を調査するために重要な情報である繁殖活動のモニタリングをする必要がある。したがって、接近を検知するだけでなく、すれ違っただけなのか、交尾が行われた可能性があるかを判別できる必要がある。そのためには、最低でも対象動物が交尾にかかる時間の $1/2$ の間隔で通信を行い連続して近距離にいることを確認する必要がある。

動物においては陰茎骨の長さや交尾の時間には相関がある [17] とされ、ネコなどは短く 1 分程度、犬などは 10 分～1 時間程度かかり、生態がよく解明されていない動物であっても推定が可能である。

これらの事から、観測対象の動物の移動速度や交尾にかかる時間からノードの Sleep 間隔を設定することが可能であり、式 3.1 または対象動物の交尾の時間の $1/2$ の短い方に合わせて設定するのが妥当であると考えられる。

3.2.2 ルーティング

森林環境で動物をキャリア (ノード) として DTN を考えた場合、構築されるトポロジ・将来の動作は非決定的である。そして、通信帯域、各ノードのデータ記憶領域に制約があることから、通信コストが下げられ、データのコピー数を抑制できる ProPHET や TCTR 等の学習型アルゴリズムに基づくルーティングが有効であると考えられる。しかし、情報伝播の速度と、端末のメモリ量、情報のライフタイムのバランスによっては、ルーティングプロトコルを単純化し、計算量を減らす方が端末の消費電力上有効である場合もある。そのため、生成される情報量、動物の移動・すれ違いによって得られる情報伝播速度を鑑みて、最適な情報のライフタイムやルーティング手法を検討する必要がある。

また、本研究が対象とする動物指向クラウド環境では図 3.1 の様に対象地域の境界地域に存在するねぐら基地局にはインターネットなどの外部への定常的な接続性を有している。この外部への定常的な接続性を持ったノードは森林の外側に対象地域を取り囲むように点在させゲートウェイの役割を担わせる。データを効率的に転送するためには、最寄りのねぐら基地局にデータを転送する必要がある。データを運搬する動物が持つ個々の生活圏、位置基準となるねぐら基地局や他の首輪ノードとの接触などの履歴から転送先を決定するアルゴリズムを、計算・電力的に低コストに実現する必要がある。

3.2.3 ノードの時間同期

首輪+ねぐらネットワークのノード同士が通信する為には、ノード同士が Sleep/Wake 間隔を同期し、相互に通信を可能な状態を保つ必要がある。ねぐら基地局は固定設置であるため GPS などのデバイスにより正確な時間を取得し時間を調整することが可能である。これに対して、首輪ノードは地上高が地表面に近く、GPS を受信しにくい。また電源をバッテリーに頼るため、時間補正用の GPS チップを搭載することは難しい。

しかし、小型のノードに搭載されるクロック IC のみの精度では温度、電圧、個体差などの揺らぎにより同期状態を維持できない。通常はリアルタイムクロックによる精度向上ははかれるが、気候などの悪条件により年単位で十分な時刻誤差に抑えることは困難であると推測される。日本をはじめとする標準電波の送信が行われている国では GPS モジュールより小型・省電力を実現可能な、電波時計による補正機能を持つリアルタイムクロック LSI 等を搭載することにより時間同期をする手法も考えられる。しかし、野生動物が行動する森林内地上数十センチといった特殊な環境は標準電波の利用環境としては想定されておらず、また、装着動物の冬眠などにより首輪ノードが長期間穴などに入る場合もあるため、このような LSI 単独で時間同期を行うことは困難であると考えられる。これらの問題から、通信時に互いの時刻同期を行い、自立的に DTN ノード間での時間同期を行う、ねぐら基地局に基準となる時刻情報を送信する機能を付加するなどの補助的な機能も必須であると考えられる。

3.2.4 IEEE802.15.4 テストベットによる事前技術検証

森林環境へのねぐら基地局の設置、動物への首輪センサの設置を行った後、ソフトウェアに何らかの不具合があったとしても、回収してソフトウェアの改修等を行うことは非常に難しい。その為、事前にネットワークを構成するソフトウェア群の検証が重要となる。しかし、実機を用いて森林環境での検証を行うことは現実的ではない。

ネットワーク実証実験環境である StarBED [5] では IEEE 802.15.4 の通信エミュレーション、ユビキタス環境エミュレーション技術を組み合わせた IEEE 802.15.4 ネットワークエミュレーションテストベット QOMB の研究を行っている [11]。QOMB は電波環境をエミュレーションするモジュールや IEEE 802.15.4 通信モジュールのエミュレーションを行うモジュールなどから構成されている。そして、ノードの移動モデルやノイズ源、障害物をパラメータとして与えることでユビキタス環境での IEEE 802.15.4 を使った大規模なネット

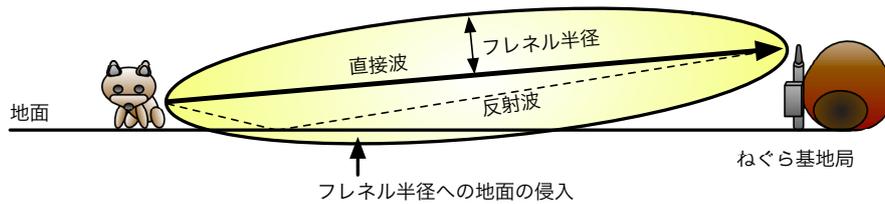


図 3.2: アンテナ地上高が低い場合のフレネルゾーン

電波の送受信端点の地上高 (アンテナ地上高) が、動物の首輪など低い場合、電波の伝播路となるフレネルゾーンに地面が侵入する。その結果、減衰などが発生する。フレネル半径は周波数が低いほど大きくなる。

ワークエミュレーションを行うことが可能である。しかし、現状のシステムは人間が持ち歩くデバイスを想定した市街地や屋内での環境再現を想定して設計されており、地形やアンテナ地上高による電波伝播の変化を再現機能はない。

首輪ノードのアンテナ地上高はタヌキやアライグマなどの小型動物では 15cm 程度になり、図 3.2 のようにノード間の電波の伝播路であるフレネルゾーンに地面が入る。その結果、減衰が大きくなり、また地形などの影響も受けやすくなる。

3.3 車車間通信ネットワーク

StarBED Project [5] では、災害時のネットワーク補完技術として、モバイルアドホックネットワークと車車間通信のエミュレーション基盤の研究を行っている [18, 19]。車車間通信ネットワークは、渋滞情報の収集や、運転危険箇所情報の共有などのセンサーネットワークとしての活用方法の研究も行われている。代表的な研究としては、DUMBO (Digital Ubiquitous Mobile Broadband OLSR) Project [20] 等が上げられる。このような実フィールドでの実験は、重要であるが、検証を行う為のコストが問題となる。また、シミュレーションベースの実験では、実アプリケーションを動作させることが出来ないため、エミュレーションによる検証が必要となる。そこで、車車間通信の実アプリケーションレベルでの概念実装検証を行う為、モバイルワイヤレスネットワークエミュレータである QOMB [11] と既存の DTN 実装である IBR-DTN [21] を用いて、車車間通信ネットワーク上でのデータ転送をエミュレーションする実験を行った。

表 3.3: 移動パターン毎の目的地選択確率

Mobility pattern	Town hall (EMA#1)	Secondary school (EMA#2)	other destinations
A	0.5	0.4	0.1
B	0.4	0.3	0.3
C	0.3	0.2	0.5

3.3.1 エミュレーション手法

本研究では、車車間通信を用いた検証環境を以下のように作成し実験を行った。自動車に搭載されたデバイスとして、組み込み Linux である、OpenWrt を用いたノードイメージを作成した。この OpenWrt には DTN Protocol の実装として IBR-DTN を、ネットワークのエミュレーション機構として QOMET のモジュールをインストールした。この仮想マシンイメージを 5 台の物理マシンを用いて、それぞれ 20VM ずつ起動し、100 台の仮想マシンを作成した。

3.3.2 車の移動データ

仮想マシンが空間上で移動するデータを作成する為、OpenStreetMap [14] から取得した道路地図データを基に、The One simulator [13] で 100 台の仮想車ノードが道路沿いに設定した目的地まで移動する移動データを作成した。表 3.3 に設定した仮想車ノードの目的地の設定確率分布 (シナリオ) を示す。

実験では、災害時の代表的な車の目的地として、避難所を想定した市役所と、中学校の 2 カ所を設定した。地図上でランダムに配置した初期値から、それぞれのノードが目的地を設定して、道路沿いに移動する。目的地に着いたノードは次の目的地を決めて移動を繰り返す。この時、目的地として、市役所、避難所を選択する確率変化させ、3 種類のシナリオを作成した。また、自動車の密度によるネットワークの性能差を見るため、3 種類の移動シナリオでノードの間引きを間引き、12 台、30 台、100 台のデータを作成、全体で 9 種類のネットワークシナリオ作成した。

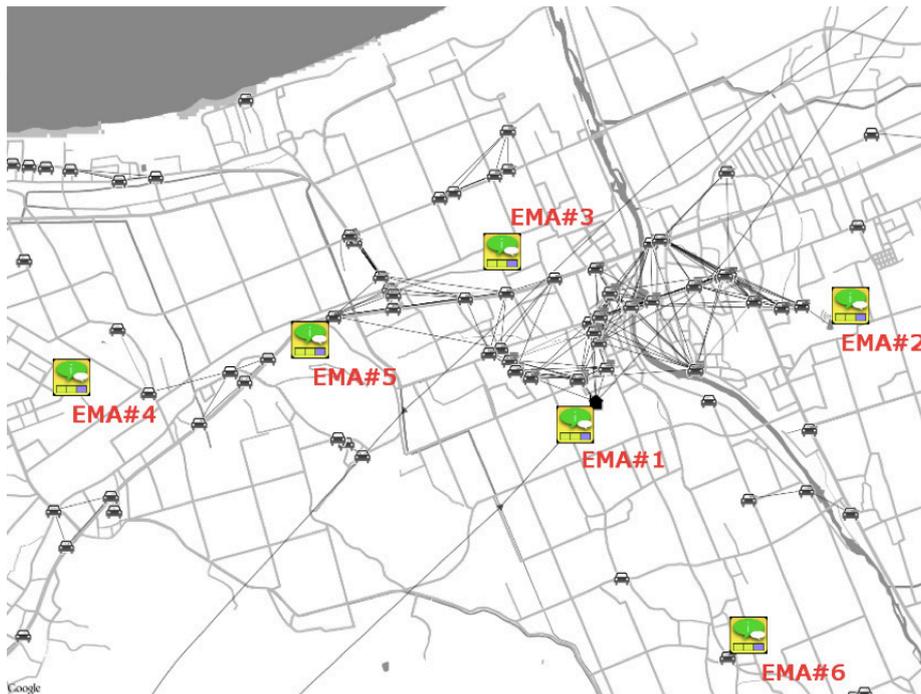


図 3.3: 実験のスクリーンショット

実験中の画面。EMA#1 と EMA#2 は固定ノード。EMA#3-#6 は車ノード。車ノードは道路に沿って移動する。図中の線は各ノード間で通信可能な状態である事を示す。

3.3.3 ネットワークに送信するデータ

ネットワークに送信するデータを生成する為、本実験では DTN 上でテキストデータを送受信するチャットアプリケーション Emergency Messaging Application (EMA) を作成した。EMA は Ruby で作成したアプリケーションで、IBR-DRN のコマンドである `dtnsend` と `dtnrecv` を利用して 20 秒に 1 メッセージ、全 EMA ノードにテキストメッセージを送信する。図 3.3 は、実験を可視化した画面のスクリーンショットである。中央にある EMA#1、EMA#2 がそれぞれ、市役所と中学校であり、この 2 つのノードは移動しない。残りのノードは車ノードの一部として移動しながら DTN ネットワークを構成する。

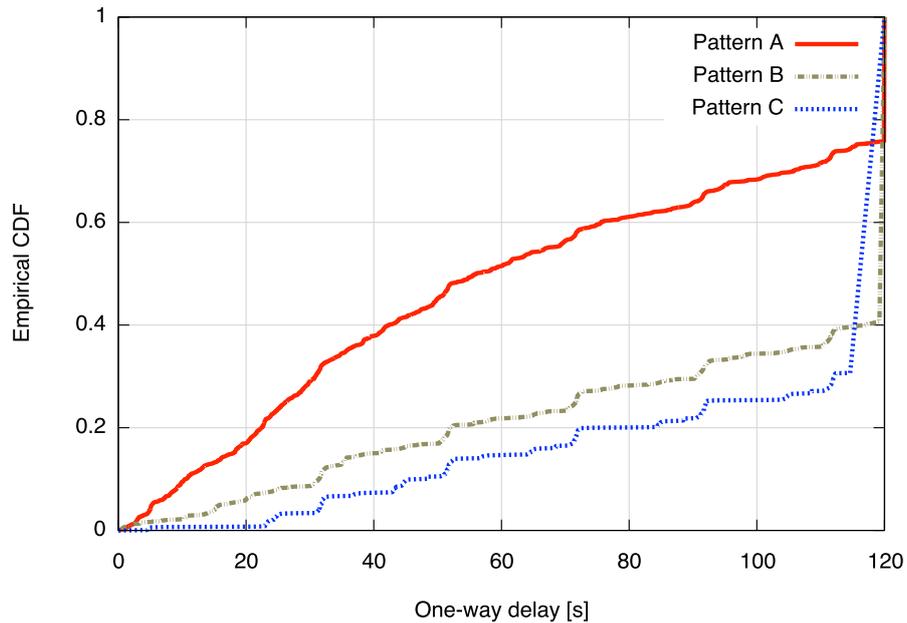


図 3.4: 移動モデル毎の片道遅延累積分布 (CDF) グラフ

本実験ではデータのライフタイムを 120[s] と設定している。グラフ中では便宜上 120[s] で 100%としているが実際には消失して届いていない。

3.3.4 実験結果

図 3.4 は 100 台の車ノードをエミュレーションした時に、EMA#2 から残りの EMA ノード 5 台への平均単方向遅延時間の CDF である。120[s] で全てのパターンで 100%になっているが、これはデータのライフタイムを 120[s] としているの為に、直前のポイントが到達レートを表す。データの送信ノードとなる EMA#1 へ多くのノードが集まる PatternA が情報の平均遅延時間が短く、到達確率も高い。

図 3.5 は PatternA のモビリティモデルの時に、車ノードの台数を 100 台、30 台、12 台と変化させた場合の CDF である。ノードの台数が増えると共に、ノードの密度が増える為、データの到達確率が上がり、データの平均遅延時間は減少している。

ノードのモビリティの変化とノードの台数の変化に対する変化の二つの結果を表 3.4 にまとめた。この 100 台の PatternB よりも 30 台の PatternA の方が平均遅延時間、ロスレート共に良好な結果となっている。この事から、台数よりもモビリティパターンが重要である事がわかった。

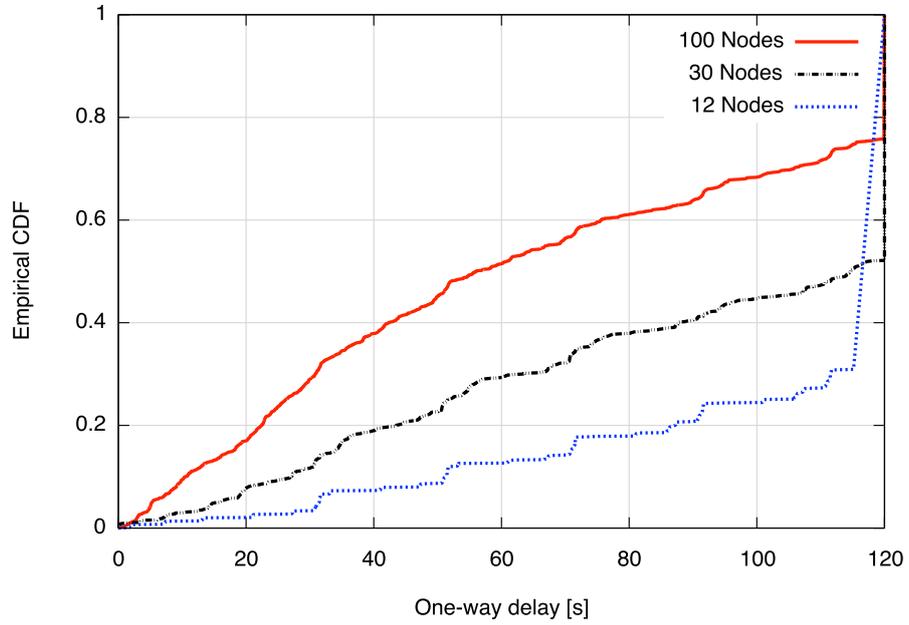


図 3.5: 移動パターン A における車の台数毎の累積分布 (CDF) グラフ

表 3.4: 移動パターンと車の台数毎の平均遅延とデータ損失率

Pattern	100 cars	30 cars	12 cars
A	47.2s/24.2%	58.0s/47.8%	67.5s/69.1%
B	58.8s/59.3%	74.8s/65.3%	70.9s/73.7%
C	64.9s/69.3%	74.8s/76.2%	71.2s/77.1%

3.4 車車間通信ネットワーク設計における技術課題

3.4.1 ノードのモビリティ

車車間通信ネットワークの設計検討によって明らかになった課題は、ノードの台数よりも、ノードのモビリティがネットワークの性能に与える影響が大きかった事である。通常時の渋滞情報共有など、十分なノード密度が期待出来る場合は問題無いが、ノード数が未知数である非常時通信を想定する場合は、その移動モデルも考慮した検討を行わないと、期待した性能が得られない事が考えられる。これは、災害マニュアルで避難所間の物資運搬車両や経路を設定するなど、施策的な支援を要する必要がある事を示唆している。そして、検討時には被害状況の想定から移動可能な車両数、災害時に発生しやすい渋滞などの考慮が必要となる。

3.4.2 ソフトウェアの検証

今回の実験では、グラフで示した最大100台・20秒間隔のメッセージでは問題が無かったが、ノード数・メッセージ数が増加した場合に、ネットワークの性能が低下する現象が観測された。詳細な解析は行っていないが、今回実験で利用したソフトウェアの高負荷時の挙動に問題があると考えられる。この事例は、エミュレーション環境を利用した実実装に基づくソフトウェアの検証が重要である事を示している。

3.5 未知の特性を含む環境を想定したセンサーネットワーク設計に必要や特性

本章では、前節までに野生動物を観測するネットワークと設計と、車車間通信の論理的検証を行う事で、未知の特性を含む環境を想定したセンサーネットワークの設計に必要な特性を議論してきた。そして、この様なセンサーネットワークでは、ノードの行動モデル、対象環境の地形情報による電波伝播の再現がネットワークを設計する上で重要となることが明らかとなった。

3.5.1 ノードの移動モデル

ノード数が変動する場合や、ノード密度が低くすれ違いの頻度が低い場合は、その移動モデルも考慮した検討を行わないと、期待した性能が得られない事が明らかとなった。自動車や人などの移動データは、近年のビッグデータに代表されるユビキタスなデバイスから取得したデータで実データを再現可能な場合もある。しかし、野生動物など、行動モデルが不明な場合もある。その場合は、他の近似の事例や、既存の移動モデルを用いて積極的な推定を行う必要が有る事が明らかとなった。

3.5.2 対象環境での電波伝播の再現

一般に電波伝播をシミュレーションする方法としては、計算式に奥村-秦式 [22,23] が用いられていることが多い。しかし、奥村-秦式は移動体のアンテナ地上高が $1m$ 以上の場合の経験的近似式であり、動物の首輪ノードでのアンテナ地上高では計算式としては不正確となる。また、自動車などのアンテナ地上高が一定程度有る場合でも、個々のノードが移動している場合は、建物や地面などの遮蔽物による影響を受ける。これまでの、センサーネットワークエミュレーションでは建物などの遮蔽物は再現していたが、想定する環境が屋内であったり、通信デバイスの電波伝播距離が短い等の理由で、地形による影響はあまり考慮されてこなかった。

より詳細な電波伝播シミュレーション手法として、SRTM-3 [24] と Longley-Rice Model [25] を用いた電波伝播シミュレーションが行われている [26]。しかし、SRTM-3 はアメリカ国内で解像度 $10m$ その他地域で $30m$ である。IEEE 802.15.4 の様な近距離通信の伝播距離は $100m \sim 200m$ であり、地形情報としての解像度が不足している。

これに対し、2007年より国土地理院が作成、公開している基盤地図情報では全国 $5m$ メッシュの標高データ (一部地域は $10m$) の地形データが公開されている。既知の電波伝播モデルと国土地理院が公開している基盤地図情報を組み合わせてシミュレーションすることで近距離通信技術にも対応可能なより詳細な電波伝播シミュレーションが行えると期待できる。

第 4 章

ワイヤレスネットワークにおける地形の影響

StarBED [5] で研究されているネットワークエミュレータである QOMB [11] は、建物等の障害物データ、屋外・屋内と言った環境パラメータやデバイスの特性を基にワイヤレスネットワークのエミュレーションを行う事が出来る。QOMB は位置情報やデバイスの特性を基に電波環境をエミュレーションする QOMET [27] や通信デバイスのエミュレーションを行うモジュールなどから構成されている。そして、ノードの移動やノイズ源、障害物をパラメータとして与えることでユビキタス環境での大規模なネットワークエミュレーションを行うことが可能である。

しかし、現状のシステムは人間が持ち歩くデバイスや車載デバイスを想定した市街地や屋内での環境再現を想定して設計されており、電波の減衰モデルは Faria [28] らの近似式を利用している。その為、本研究で必要とする地形やアンテナ地上高による電波伝播の変化など起伏に富む森林環境の再現に必要なパラメータを再現する機能はない。

市街地や、林野部を対象とした電波減衰の近似式はこの他にも、奥村/秦式等 [22, 23] が有名であるが、いずれも通信機器の地上高が $1m$ 程度の人が持つ、携帯電話など小型デバイスとしては比較的長距離の通信を行えるデバイスを想定しており、起伏に富んだ地形の影響を十分には反映できない。



図 4.1: 白峰農業体験学習施設多目的広場:白山市

4.1 検証用機器による電界強度測定

地形、そして低い地上高における電波伝播への影響がどのようにあるのか、アンテナ地上高が低い場合の影響を検討するため、検証用機器を作成しアンテナ地上高の低さが電界強度とに与える影響を計測する実験を行った。

電波は距離の二乗に反比例して減衰するが、自由空間上の送信出力と受信信号強度の差である、電波伝播損失 L は、式 4.1 で表される。

$$L = 10 \log_{10} \left(\frac{4\pi d}{\lambda} \right)^2 + G \quad (4.1)$$

ここで、 d は距離、 λ は波長、 G は送信、受信両ノードのアンテナ利得や、ケーブルなどのデバイス特有の損失などの雑損失を合算した定数である。この時 G は利用する首輪、巣穴デバイス固有の値となるため、事前に計測してシミュレーションパラメータに組み入れる必要がある。

検証用デバイスとして、802.15.4 によるデータ送信機と受信機を試作し、見通しの良い場所と森林での電界強度を測定する実験を行った。802.15.4 の通信が用いる 2.4GHz 帯は、Wi-Fi 等が混在する帯域である。そのため、他の電波から影響を避けるため、山間部に位



図 4.2: 計測の様子



図 4.3: 石川県林業試験場 (樹木公園):白山市

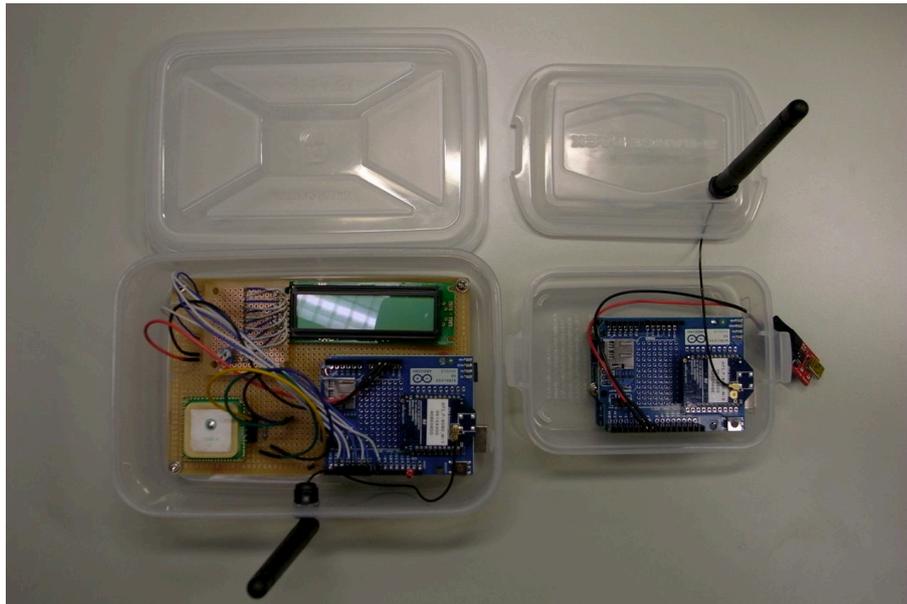


図 4.4: 検証用送信機 (左)・受信機 (右)

置する白峰農業体験学習施設多目的広場 (写真:図 4.1) で、平地を想定した実験を石川県林業試験場 (樹木公園) で実験を行った (写真:図 4.3)。実験に用いた機器の詳細を表 4.1、計測の条件を表 4.2 に示す。

平地として実験を行った多目的広場は見通しの良い平地の広場で、計測時は固い雪に覆われていた。計測は200m まで10m 間隔で計測し、各計測地点での送信機よるブロードキャストパケットを50回受信し、その平均と標準偏差を求めた。

送信機のアンテナ地上高は100cm、受信機のアンテナ地上高は送信機と同じ100cm と小型動物の首輪デバイスを想定した15cm の二種類で測定した。実験の結果とQOMBを用いてエミュレーションした電波強度を図 4.5 に示す。計測した値はXBee の受信電波強度 (RSSI) であり、空間減衰の他に送信機受信機双方のケーブル損失やアンテナ利得による減衰を含んでいる。QOMBでのエミュレーション結果は送受信機の内部損失無しでエミュレーションしているため、自由空間での理論値と同じである。受信機のアンテナ地上高が100cm の場合は200m の地点でもパケットのロスが2% でデータが送信できているが、受信機のアンテナ地上高15cm では190m 地点で、10% に上昇し200m 地点では受信不能であった。小型動物を想定したアンテナ地上高15cm の場合電波の受信可能距離は短くなり190m 以下であることがわかり、フレネルゾーンに地面が入っている影響がうかがえる。受

表 4.1: 送/受信機詳細

マイコン	Arduino Uno R3
通信モジュール	XBee Pro Series 1 Wireless SD Sheild
アンテナ	XBee 用外部アンテナ
電源	eneloop stick booster
その他 (送信機のみ)	液晶モニタ GPS モジュール

表 4.2: 測定実験条件

送信機アンテナ地上高	100cm
受信機アンテナ地上高	100cm,15cm
電波送信出力	10mW
測定距離 (10m 間隔)	平地:10m ~ 200m 森林:10m ~ 100m

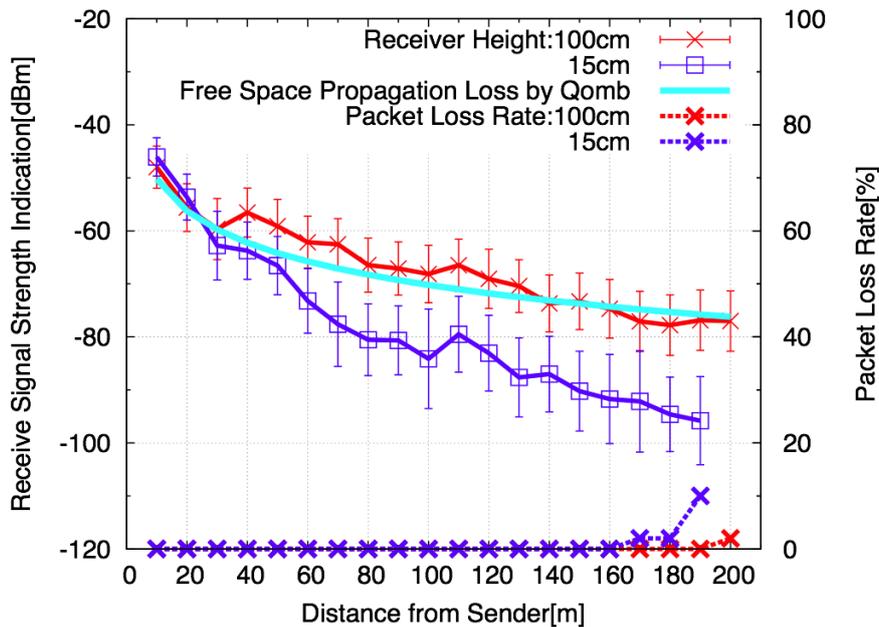


図 4.5: 実験機による電界強度測定結果 (平地)

信機のアンテナ地上高 100cm および 15cm のともに 100m の計測地点で受信強度の上昇が見られているが、送信機から 100m の計測地点付近に 10cm ほどの雪面の隆起があった為だと考えられる。受信機のアンテナ地上高 1m の時の RSSI は QOMB による理論値とほぼ同じであることから、今回作成したプロトタイプのアナテナ利得とケーブルなどの雑損値の合計はほぼ 0dB であることがわかる。

また、森林での実験結果と、平地での実験結果比較を図 4.6 に示す。森林での実験は、場所の都合から距離 100m までの実験を行った。受信アンテナの地上高が 15cm の場合、森林では平地の場合より、伝播減衰が大きく、70m で既に 20% 程度の Packet Loss が発生している。また、70m 以降では Packet Loss 率に大きな揺れが発生しているが、これは 80m の計測ポイントの近くに樹木があった為、局所的にフレネルゾーンの遮蔽率が大きくなった為と推測される。

4.2 地形データを用いた電波エミュレーション

ノードのアンテナ地上高が低い場合、ノード間の電波の伝播路であるフレネルゾーンに地面が侵入する。また、地上高が低いと、地上高が高い場合より、地形の起伏が遮蔽物と

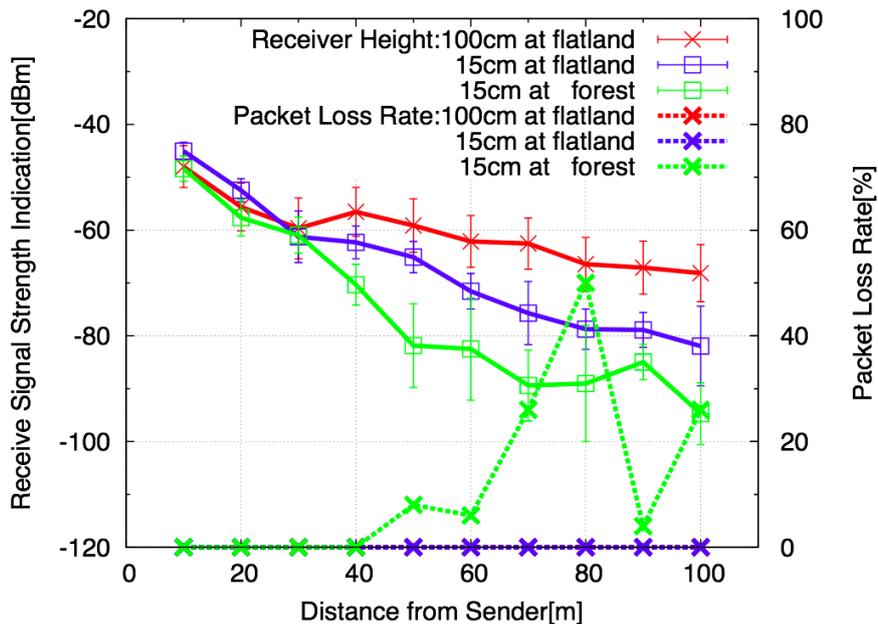


図 4.6: 平地と森林での電界強度の比較

なりやすくなり、その結果、伝播減衰が単純な距離による減衰より大きくなる。

より詳細な電波の伝播シミュレーション手法として、SRTM-3 [24] と Longley-Rice Model [25] を用いた電波伝播シミュレーションも行われている [26]。しかし、SRTM はアメリカ国内で解像度 $10m$ その他地域で $30m$ である。IEEE 802.15.4 の様な近距離通信の伝播距離は $100m \sim 200m$ であり、地形情報としての解像度が不足している。これに対し、2007 年より国土地理院が作成、公開している基盤地図情報では全国 $5m$ メッシュの標高データ (一部地域は $10m$) の地形データが公開されている。既知の電波伝播モデルと国土地理院が公開している基盤地図情報を組み合わせてシミュレーションすることで近距離通信技術にも対応可能なより詳細な電波伝播シミュレーションが行えると期待できる。

4.2.1 電波伝播モデル

電波の送受信の端点間での電波の伝播モデルおよび、障害物の影響は多くの研究と実値計測により明らかになっている。フレネルゾーンは、図 4.7 のような、電波の電波路として一般的に用いられるモデルで、 d_1, d_2 の和が電波の送受信の端点 A, B 間の最短の距離よりも半波長の $n - 1$ 倍以上でかつ n 倍以内の範囲を第 n フレネルゾーンと呼び、半波長だ

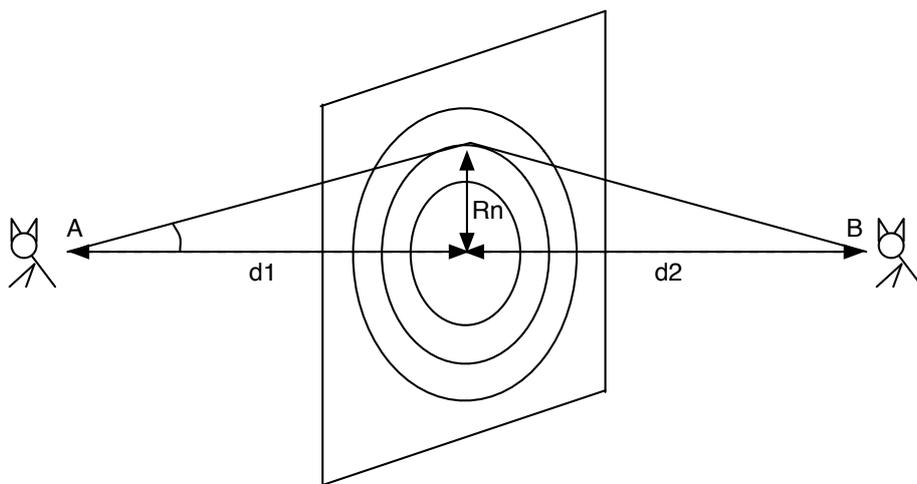


図 4.7: フレネルゾーン

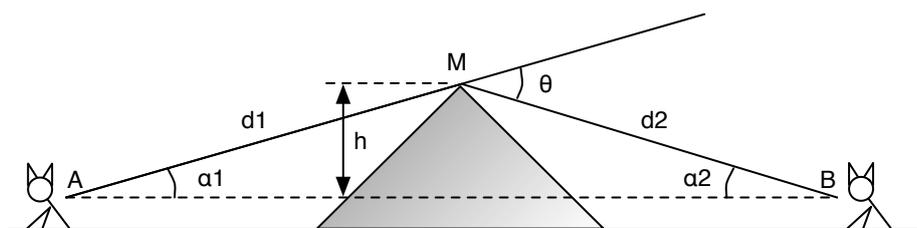


図 4.8: ナイフエッジ回折

け長い $n = 1$ の楕円 (3次元実空間上では回転楕円体) の内側の範囲を第1フレネルゾーンと呼ぶ。伝播路の途中にある遮蔽物が、その第1フレネル・ゾーンを遮蔽する程度で、遮蔽による伝播損失がほぼ決定する。

また、電波の送受信の端点 A 、 B の間に起伏などの遮蔽物が入った場合の伝播モデルは図 4.8 のナイフエッジ回折モデルとなる。

電波の送受信の端点間に複数の遮蔽物があった場合は、厳密には図 4.9 のような複雑なナイフエッジ積分を含むが、一般的には図 4.10 のように複数の遮蔽物をまとめた形で、単純なナイフエッジモデルとして計算する事が出来る。

QOMB が利用する市街地や山林地域での電波の伝播モデルに、これら地面による遮蔽の

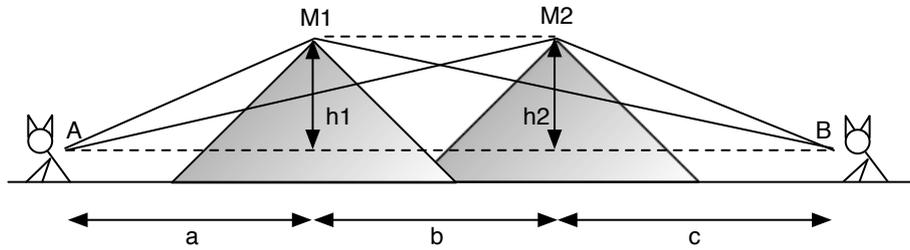


図 4.9: 二重ナイフエッジ回折

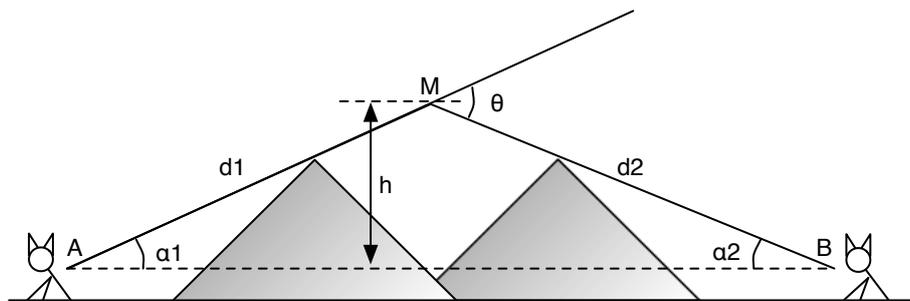


図 4.10: 仮想単一遮蔽ナイフエッジ回折

影響を計算モデルとして導入し、エミュレーション対象地域の地形データを利用することで、より詳細なワイヤレスネットワークエミュレーションが行える。

第 5 章

センサーネットワークにおける取得データの推定と解析

5.1 取得可能データの推定

ワイヤレスセンサーネットワーク特にアドホックネットワークやDTNを用いたワイヤレスセンサーネットワークで得られるデータは、商用通信サービスや、安定したネットワークで構成されるセンサーネットワークで得られるデータと異なり、非連続的で粗な情報となりやすい。また、すれ違い情報の収集を伴う場合は、情報の非連続性自体にノードの移動特性などの計測対象としての意味合いが出てくる。その為、設計したセンサーネットワークの有用性、妥当性を検証するためには、設計時に取得されるデータの推定を行う必要がある。従って、推定されるリファレンスデータの作成、全体のデータを照合・解析する技術が必要となる。シミュレーション・エミュレーション環境において、設計したネットワークで得られる（想定される）データを生成することで、設計したネットワークの概念検証を行う事が出来、また、実際に展開したネットワークから得られたデータと照合することで、様々な知見を得ることが可能となる。

しかし、移動ノードで構築されるアドホックネットワークやDTNを用いたワイヤレスセンサーネットワークでは、通常は取得データの推定は困難であるが、ノードの移動を移動モデルや実データにて推定した場合は取得可能データの推定も可能となる。

5.2 展開後の実取得データの解析

ノードの展開後、そのネットワークの設計が最適であったか、または改善の余地があるかは、実データの解析を行って、知見を得る必要が有る。しかし、アドホックネットワークやDTNではこれらのデータは時間的空間的に非連続なデータとなるため、何らかの指標に基づいて移動モデルを逆推定するなどの解析が必要となる。ノードがGPSを持っている場合、または持っているノードが混在する場合は、その履歴から行動モデルを推定する事も可能である。また、藤原らは、人が持ち歩いた Bluetooth 機器のすれ違いデータから、人の行動モデルを推定している [29]。同様の手法を用いることで、すれ違い通信の情報から、ノードの移動モデルを推定する事も可能であると考えられる。

第III部

対象環境の物理特性を考慮した大規模実証 実験環境

第 6 章

ネットワーク実証検証環境の構築確度向上 と効率的な構築技術

多数の実ノードを用いた大規模な実証検証環境の構築時には、ノードの起動不良やファイルサーバの負荷等により多数の障害が発生する。大規模な実証検証環境で起こる障害の原因特定・回避には、その実験環境での利用経験と、起こる障害についての知識が必要であり、そのことが実験者にとっての負担となっている。そのため、アプリケーションの検証を容易に行うには、実証検証環境構築時に遭遇する障害の回避・復旧手法を含めた、実証検証環境構築のフレームワークが必要となる。

本章では、実ノードを用いた大規模な実証実験を行うための施設である StarBED [30] において、仮想化技術を用いた大規模なアプリケーション実証検証環境の構築を行うためのフレームワークを検討する。

6.1 既存のネットワーク実証検証環境の構築手法

StarBED は汎用の機器で構成された実験施設であり、ここで発生する障害は他の実験施設でも起こる可能性がある。したがって、StarBED で得られる知見から設計するフレームワークは、他の実証検証環境にも適用可能であると考えられる。実験ネットワークの構築を容易に行う為に StarBED では SpringOS による実験支援ソフトウェアを運用・提供している。近年では StarBED で仮想化技術を用い、論理的に実ノードの数以上のノードを作成し実証検証環境を構築する事が行われているが [31, 32]、SpringOS の仮想化技術を用いた実証検証環境を構築する機能はまだ不十分である。

6.2 既存の実証検証環境構築手法

StarBED を利用して仮想化技術を用いた実証検証環境を構築するためには、StarBED の実験用実ノード (以下実ノード) の OS インストールと設定、仮想ノードのディスクイメージの配布、実験対象のアプリケーションの導入、実験ネットワークの構築を行う必要がある。本節では既存の実証検証環境構築で用いられている手法を説明する。

6.2.1 実ノードの作成と複製

既存の実ノードへの OS 等のソフトウェアを導入する方法は 2 種類ある。1 つは 1 台の実ノードで OS インストールと設定を行い、テンプレートとなるディスクイメージを作成する。このディスクイメージの内容をコピーし、他の実ノードに配布、起動することにより環境を構築する方法である。2 つめはネットワークブート (PXEBoot) 用の実ノードのディスクイメージを作成しネットワークブートする方法である。

6.2.2 仮想ノードのディスクイメージの配布

仮想ノードのディスクイメージを配布するためには、2 つの方法がある。個々の実ノードのローカルディスクに仮想ノードのディスクイメージを配布し起動する方法と、NFS 等でディスクイメージが置かれたディレクトリをマウントし起動する方法である。

前者ではネットワークを用いてディスクイメージを複製するため、構築に時間を要するが、起動時にはローカルディスクから起動するため、ネットワークへの負荷はなく、仮想ノードの起動時間も短く、起動後のパフォーマンスも良い。後者ではネットワークマウントの作業自体はすぐに終わるため、環境構築は早い。しかし、仮想ノードの起動時にファイルサーバに負荷が集中し、起動に時間を要する。また仮想ノード起動後のパフォーマンスも悪い。

6.3 ネットワーク実証検証環境構築の精度

実際に実証検証環境を構築する過程では、様々な障害に遭遇する。電源管理に用いる IPMI や iLO [33] 等の Remote Management Interface(以下 RMI) の信頼性の不足、実ノード起動

時の管理サーバへの負荷の集中が原因である場合が多い。本章ではそれらの障害と回避方法を説明する。

6.3.1 電源管理障害

電源管理障害は、実ノードを起動、停止、再起動する際に利用する RMI に起因する障害である。実際に起こる現象としては、電源制御の命令が実行されない。レスポンスが無い。実ノードの電源状況とレスポンスの内容が異なる等である。また、TCP でコネクション確立して通信する電源制御方式を採用するモデルにおいては、コネクションが異常切断される場合もあり、これも電源管理の障害に分類される。メッセージが実行されない、またはレスポンスが無くなる状況はネットワークの負荷によるパケットロスで起こると考えられる。この場合は処理のリトライを行う事で回避可能である。

一方で、ネットワークに負荷が高くない状況でも、レスポンスが無い場合もあり、これはハードウェアの障害であると考えられる。

実ノードの電源状況と異なるメッセージが RMI から返って来る現象は、ネットワークの負荷に関係なく、電源制御を必要とするメッセージを送信した直後に、電源の状態を問い合わせるメッセージを送った場合に発生する。この現象は StarBED で運用されている 3 種類の異なるメーカーの製品で発生している。RMI がメッセージを受け取り、電源制御を行ってもハードウェアの状態が遷移するのには時間を要するためと考えられる。したがって特に電源操作メッセージと電源状況を問い合わせるメッセージの送信間隔は十分に空ける必要がある。これまでの経験から 10[s] 以上間隔を空けてメッセージ送信を行っていれば、この現象は発生しない、

6.3.2 管理サーバの障害

管理サーバの障害は、実証検証環境に設置されている、DHCP サーバやファイルサーバのサービスが正常に受けられなくなる現象で、NFS サーバからのファイル取得失敗、DHCP サーバからの IP アドレス取得失敗等である。これは、管理サーバのハードウェア故障ではなく、ネットワークの輻輳、遮断が原因で起こるソフトウェアでの障害である。DHCP サーバ、TFTP サーバ、NFS サーバへの問い合わせが多数のノードからのリクエストが、短時間に集中しないよう実ノードを制御する事で障害発生回避が可能である。

6.3.3 ハードウェアの障害

ハードウェアの障害は、実ノードのハードウェアが故障または、故障を検知し停止する場合である。具体的にはCPU 障害、電源障害、ディスク障害等である。この障害は、修理が必要な場合と、再起動によって復旧する一時的な現象である場合がある。

障害が一時的かどうか再起動を行って判断する必要があるが、処理のリトライ、負荷分散等を行っても起動しない、または処理が正常に終了しない場合には、故障と判断して、そのノードを実ノードから除外する必要がある。故障した実ノードを除外すると、実証検証環境構築用の実ノード数が減少する。しかし、実ノード起動時にあらかじめ予備機を含めた実ノード数を起動し、起動した実ノードから想定した実ノード数を利用する事で、必要なノード数の環境構築が可能である。

6.3.4 その他

上記の他に、人為的ミスにより実験に障害が発生する場合がある。実証検証環境のリソース情報に誤りがあり、設定内容が実体と異なるため、ノードが動作不良を起こす場合等である。

6.4 マルチキャストを用いたデータ配布

6.3.3 節で検討した実証検証環境で発生する障害とその回避方法を基に、実際の実証検証環境を構築するフレームワークを設計した。実証検証環境の構築は図 6.1 の手順で行う。ネットワークの設定は SpringOS を利用して構築する。

6.4.1 実験制御

全体の制御手順を検討する。仮想ノードの起動、終了やアプリケーションの制御を行う場合、個々の仮想ノードへ命令を直接送信すると、制御メッセージを全仮想ノードに送信する必要があり、制御メッセージの送信にかかる時間が長くなる。それに伴い、実験の開始、終了などの同期制御が困難になる。したがって、実証検証環境構築のような多数の実ノードを制御する実証検証環境では、図 6.2 の様に個々の実ノードに Node Agent プログラムを設置し、個々の実ノード上で動作する仮想ノードの起動・終了や実験対象のアプリ

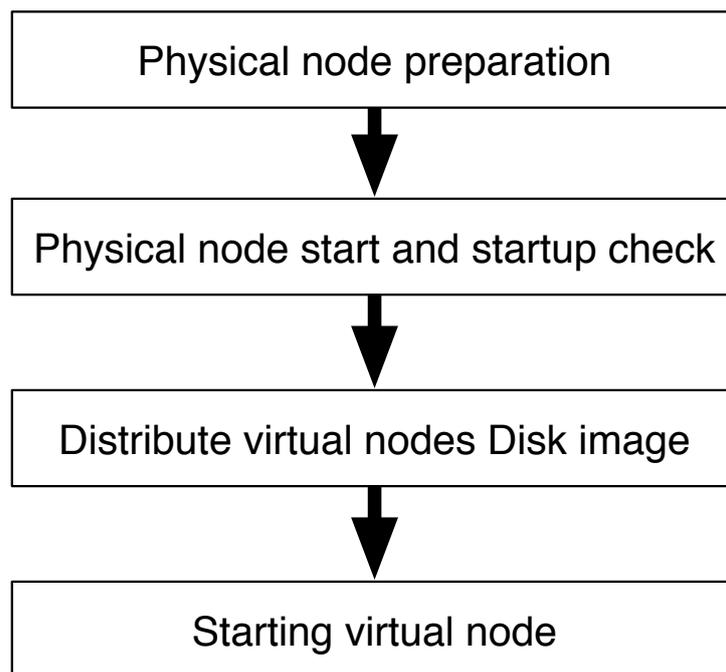


図 6.1: マルチキャストを用いた実証検証環境構築の手順

マルチキャストを用いた実証検証環境の構築では、まず物理ノードを準備し、起動を確認後マルチキャストを用いて仮想ノードのディスクイメージを配布する。実験で利用するノードの均一性が高く、ノードのディスクイメージの種類が少ない程、全体の転送にかかる時間を削減できる。

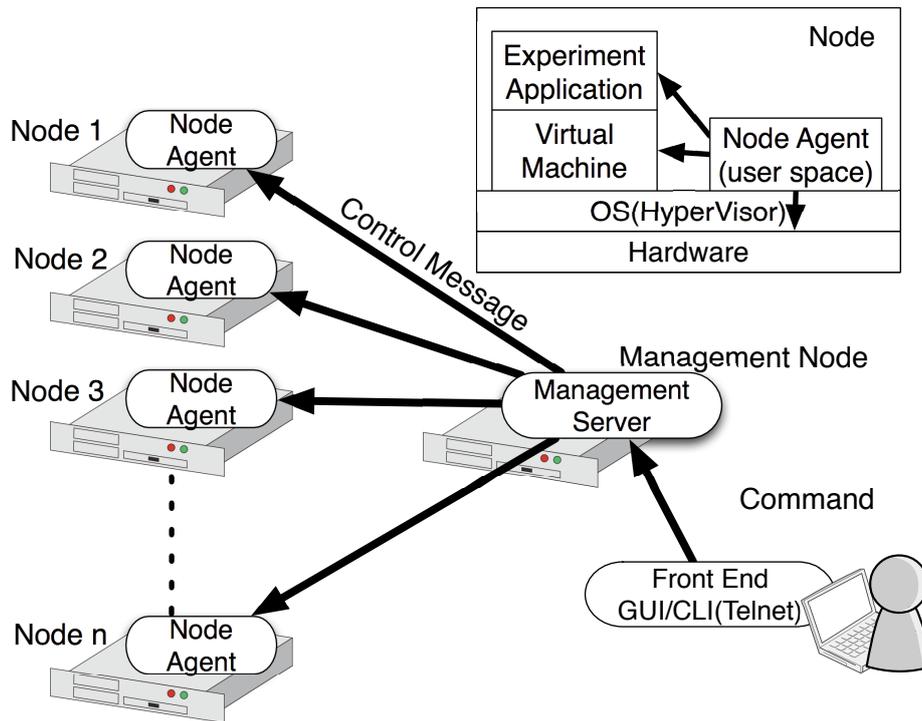


図 6.2: Node Agent を用いた実験制御フレームワークの概要

ユーザは制御メッセージをマネージャに通知する。マネージャは制御内容を解析し、HyperVisor ノード毎の制御メッセージに分割し、各 Node Agent に送信する。各 HyperVisor ノードに Node Agent を常駐させる事で、仮想化によって規模が拡大した実験環境内に階層的な制御フローを構築すると共に、同一物理ノード上の仮想ノードに統一的な制御を可能とする。

ケーションの制御を行わせる方が良いと考えられる。そして、Node Agent を管理制御する Manager Server が実験者からのコマンドを解釈し、各 Node Agent を制御する。表 6.1 に Node Agent および Manager Server が持つべき機能をまとめた。

6.4.2 実ノードの起動方式

実ノードのテンプレートディスクイメージを作成し、各実ノードのローカルディスクに OS を直接導入する手法は SpringOS で実現されている。この方法ではディスク単位、パーティション単位でのディスクイメージの配布となり、OS の導入に時間を要する [34]。ネットワークブートでの起動の場合、OS の構成ファイルを転送するだけで良いため、転送量が少なく済む。また、起動と OS の導入が同時に行われるため、環境構築作業全体の時

表 6.1: Node Agent と Manager Server の機能

	機 能
Node Agent	<ul style="list-style-type: none"> ● 実ノードのディスクの管理 パーティション作成/フォーマット、マウント...etc ● 各種ファイルの取得・転送 NFS サーバからのファイル取得、ファイルの転送 ● 実ノードのネットワーク設定 物理インターフェース・ブリッジの設定...etc ● 自ノード上の実験対象アプリケーションの制御 設定、実行、終了...etc ● 仮想ノードの管理 仮想ノードの複製、設定、起動、終了...etc
Manager Server	<ul style="list-style-type: none"> ● 実験者からのコマンド受付 コマンドの解釈、Node Agent コマンドへ変換...etc ● Node Agent の制御 コマンド送信、Node Agent 電源制御/状態確認...etc

間短縮になる。仮想ノードを用いた実証検証環境の構築では、実ノードに導入される OS は Windows 等のアプリケーションを実際に動作させる OS に比べ小容量で済み、全てをオンメモリで動作させることも可能である。以上の理由から実ノードの起動方式としては、ネットワークブートで起動する方が望ましい。

6.4.3 電源制御

電源制御では RMI に対してメッセージを送信するが、6.3.3 章で述べたように、以下の点を考慮する必要がある。

- コネクションが異常終了した場合はリトライする
- メッセージ送信間隔を一定時間空ける
- メッセージ通りの動作が確認する

これらの条件を基に 1 つのメッセージを送る際の手順を検討した。図 6.3 は単一メッセージである電源の状態確認メッセージを送信する場合の手順である。メッセージ送信時にコネクションエラーが発生した場合、または応答が無かった場合にはメッセージの再送信を行う。ただし、RMI やネットワークの障害の場合、再起動処理が終わらなくなるため、メッセージの再送信は 3 回までとした。単一メッセージの送信を行う電源状態取得や、リセットはこの手順で行う。また、実ノードの電源制御、起動確認は Manager Server が管理する。

次に実ノードの電源の状態変化が、メッセージ通りの動作が行われたか確認する必要がある場合の手順を検討した。図 6.4 は電源 ON のメッセージ送信の手順である。まず、電源の状態を確認し、電源が OFF である事を確認して、電源 ON のメッセージを送信する。その後、電源の状態を再度確認し、電源の状態が OFF のままであった場合には、再度電源 ON のメッセージを送信する。やはり、RMI やネットワークの障害の可能性があるので、メッセージの送信は 3 回までとした。電源の状態変化が起こる電源 ON、電源 OFF はこの手順で行う。

6.4.4 電源制御の並列処理

前節で個々のノードの電源制御フレームワークを検討した。1 台の電源投入に確認処理を含めて最短でも 30[s] 程度要するため、多数の実ノードの起動を逐次処理で行うには時

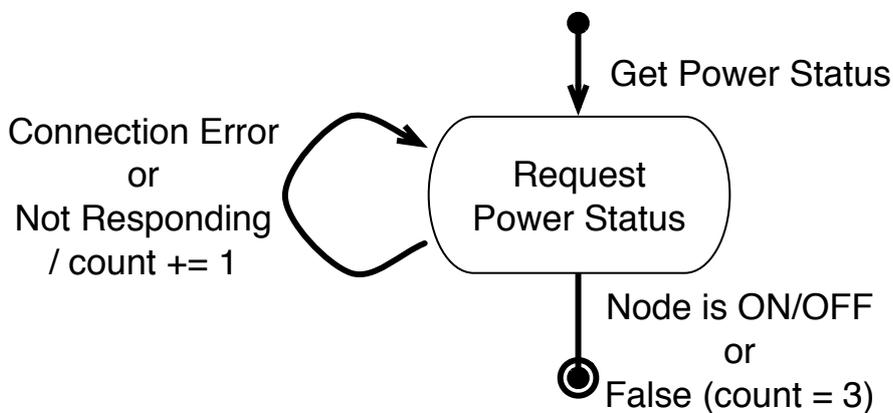


図 6.3: Power Status 取得時の状態遷移図

電源の状態の取得時に、パケットロスや制御ボードの負荷等何らかの理由で電源制御ボードとの通信が確立出来なかった場合、一定時間をおいて3回までリトライを行う。

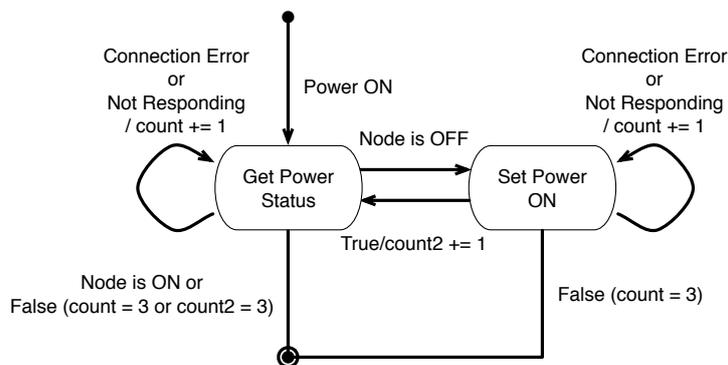


図 6.4: Power ON 時の状態遷移図

電源状態遷移の制御後、制御が正常に行われた事を確認する為に、一定時間をおいて状態確認を行う。制御命令を発行した後、どの程度の時間で状態が遷移し、且つ電源の状態問合せの結果に反映されるかは、制御ボードの実装に依存する。

間をノード台数に比例した時間を要する。したがって、実際の環境構築時には複数の電源 ON 処理を並列で処理する。しかし、多数の実ノードを同時に起動すると、障害が発生する可能性がある。PXEBoot で実ノードを起動する場合、BIOS 起動の後に DHCP サーバへの問い合わせる。多数の実ノードを同時に起動すると DHCP サーバの処理能力をリクエストが上回る可能性がある。また、一般にコンピュータは起動時に一時的に電源ユニットの最大値に近い消費電力を消費する。そのため多数の実ノードを同時に起動すると、UPS などの電源システムへの負荷により、障害が発生する可能性がある。したがって、電源 ON 時のメッセージは各実ノードを一定時間遅らせて送信する必要がある。DHCP サーバはシングルスレッド動作しているソフトウェアも多く、数リクエスト/sec 程度の処理能力である事が多い。また電源 ON メッセージの送信から DHCP クライアントの最初の packet である DHCP DISCOVER のメッセージが送信される迄の時間には、実ノード毎に揺らぎがある。そのため、実ノードの起動は 1 台/sec 程度とする。

6.4.5 PXEBoot で使用するサーバの負荷分散

実ノードが起動するまでには DHCP サーバ、TFTP サーバ、NFS サーバの 3 種類のサーバを利用する。実証検証環境構築時にはこれらのサーバの高負荷によって障害が発生しないように考慮する必要がある。

DHCP サーバと TFTP サーバ

PXEBoot では対応 PXEBoot ファームウェアの起動後、DHCP にて IP アドレスを取得し、TFTP サーバから OS 起動用のカーネルを取得する。これらの処理時間は短時間で済むため、電源制御の並列化の段階で実ノードが 1[s] 程度遅れて起動してくる場合考慮しなくて良い。

NFS サーバ

PXEBoot ではファームウェアがカーネルを起動後、NFS サーバからルートディレクトリの内容を取得する。この時、オンメモリでの OS 起動の場合にはルートディレクトリの内容を全てメモリ上にコピーする必要がある。

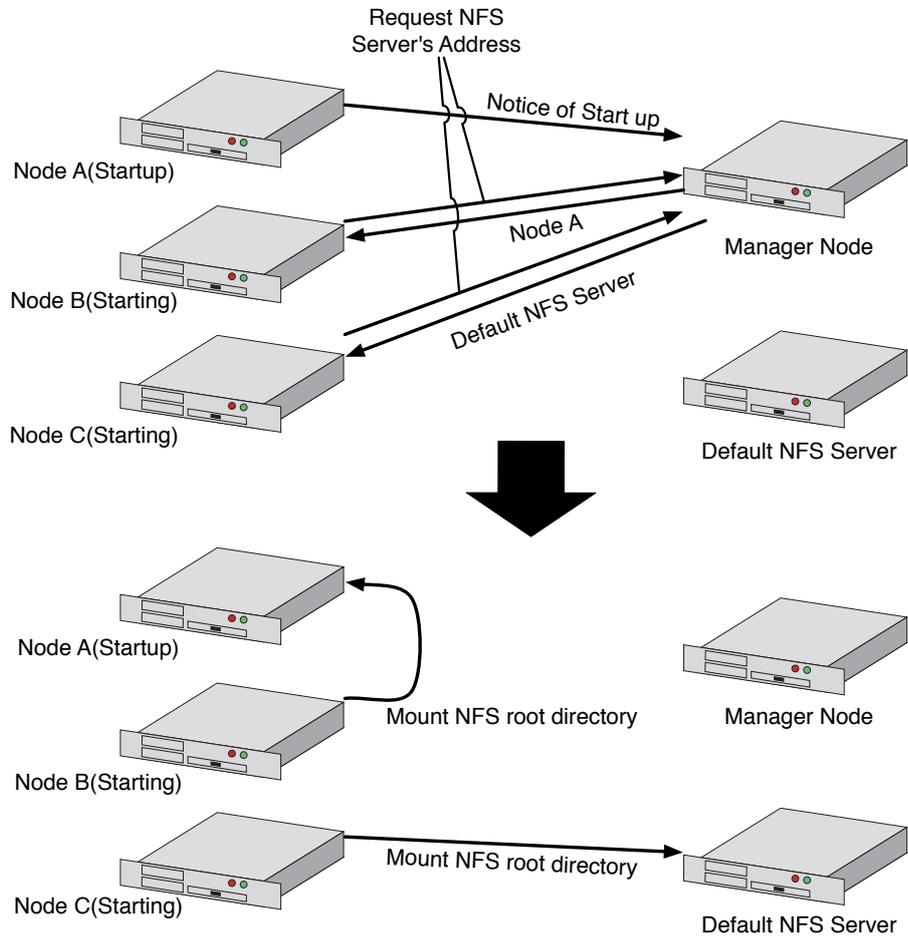


図 6.5: NFS サーバ IP アドレスの問い合わせ

HyperVisor となる各ノードを DHCP サーバの処理負荷に応じて時間差を置いて起動する事で、起動イメージを取得するタイミングにも時間差が生じる。これを利用して、起動に必要なデータを先に起動したノードから取得する様にする事で NFS Server の負荷分散が可能となる。先に NFS サーバに Root Directory のデータを取得したノードは Manager Node に通知する事で、NFS Server として登録される。Manager Node は NFS Server として利用可能なノード情報を管理し、後から起動してきたノードからの NFS Server のアドレス問合せに対し、保持している NFS Server リストからランダムにノードを選択し返答する。

仮想化ソフトウェア、Node Agent、および実験対象のアプリケーションを制御するソフトウェアなど、実ノードの起動 OS のルートディレクトリは少なくとも数百 MB 程度のサイズを持つ事になる。そのため、DHCP サーバや TFTP サーバと異なり、電源制御によって発生しているリクエストの時間差である約 1[s] では転送が終わらない。その結果、実ノードの起動開始後、時間と共に NFS サーバの負荷が増大する。したがって、PXEBoot を用いた実証検証環境の構築時には NFS サーバの負荷を分散させる機構が必要となる。

そこで、図 6.5 の様に起動済み実ノードが順次代替 NFS サーバとなることで負荷を分散する。Manager Server は NFS サーバのリストと、初期値としてデフォルト NFS サーバの情報を持っている。実ノードは起動時に Manager Server に NFS サーバのアドレスとルートディレクトリを問い合わせる。Manager Server は NFS サーバのリストからランダムに NFS サーバを選び応答する。起動した実ノードは Manager Server に起動済みであることを知らせるメッセージを送信する。Manager Server は起動済みメッセージを受け取り、その実ノードを NFS サーバリストに追加する。

DHCP や TFTP サーバはその仕様上パケットロスが発生するような輻輳に弱く、バースト的なリクエストへの応答性を確保する事は難しい。本提案では電源制御の段階から生じるノード起動の時間差を利用し、NFS Server の負荷を分散することで規模が全体での起動時間の短縮を可能とする。

起動不良ノードに対するリトライ

実ノード起動の際 NFS の負荷分散を行っても、5%程度起動しないノードが見られた。起動しないノードの一部はハードウェア故障ではなく、TFTP サーバとの通信障害、DHCP サーバからの IP アドレスの取得失敗が原因であった。そこで、起動しないノードを自動的に再起動する仕組みを検討した。Manager Server は電源投入開始から最初のノードが起動するまでの所要時間 T を記録する。その時間の最後に起動済みメッセージを受け取った時間から、 $T \times 2$ の時間待っても起動済みメッセージが来なかった場合、まだ起動済みメッセージを受け取っていないノードの RMI に対して電源リセットのメッセージを送信する。再起動が必要なノードが多い場合、故障以外の理由で再び起動不良が起こる場合があるため、この再起動作業は 3 回まで行う。

仮想ノードのディスクイメージファイルの配布

仮想ノードのディスクイメージファイルは、エンドユーザが多く利用している Windows 系 OS の場合、最低でも数ギガバイト程度の容量になる。この量のファイルを全ての実ノードが取得しようとした場合ネットワークの輻輳が起こり、またファイルサーバが高負荷状態になり処理が遅延する。そのため、仮想ノードのディスクイメージファイルはマルチキャストによって転送する。そして、実ノード上で複数の仮想ノードを起動させる際に、同一の仮想ノードを用いる場合には、各実ノードで仮想ノードのディスクイメージ及び設定ファイルの複製と書き換えを行う。NFS サーバには実験制御のプログラムを設置しないため、図 6.6 の様に、まず 1 台の実ノードが仮想ノードのディスクイメージを取得し、それを実ノード同士でマルチキャスト転送する。マルチキャストは、通常 UDP を用いた転送方式で、再送機構を持たない。その為、ディスクイメージなどのデータの完全性が必要な転送には、再送機構を持った信頼性のあるマルチキャスト転送を用いる必要がある。

6.5 プロトタイプによる評価

前章で設計したフレームワークの評価を行うために、仮想ノードによる実証検証環境を自動構築するツールのプロトタイプを作成した。本章では作成したツールの概要、実験とその評価を説明する。実ノードで動作する OS には Debian Linux を用い、Manager Server と、Node Agent は Ruby で実装した。また仮想ノードの OS として Windows XP をインストールしたディスクイメージファイルを用いた。

6.5.1 実ノードの OS 作成

仮想化技術を利用可能な OS には、ESXi、VMWare Server、Xen や KVM が動作する Linux 等がある。今回のプロトタイプ作成では、改造の容易さ、動作の軽量さから Debian Linux 5.0.6 (lenny) の Net-Boot 版をベースに作成した。仮想化の方式としては Linux で公式サポートされている KVM を用いた。一般に配布されている Debian Linux 5.0.6(lenny) Net-Boot 版からの変更内容は以下の通りである。

- KVM、Ruby 実行環境、nfs-kernel-server の導入実ノードが起動後代替 NFS サーバとしてサービスを行うため、NFS サーバプログラムを導入した。

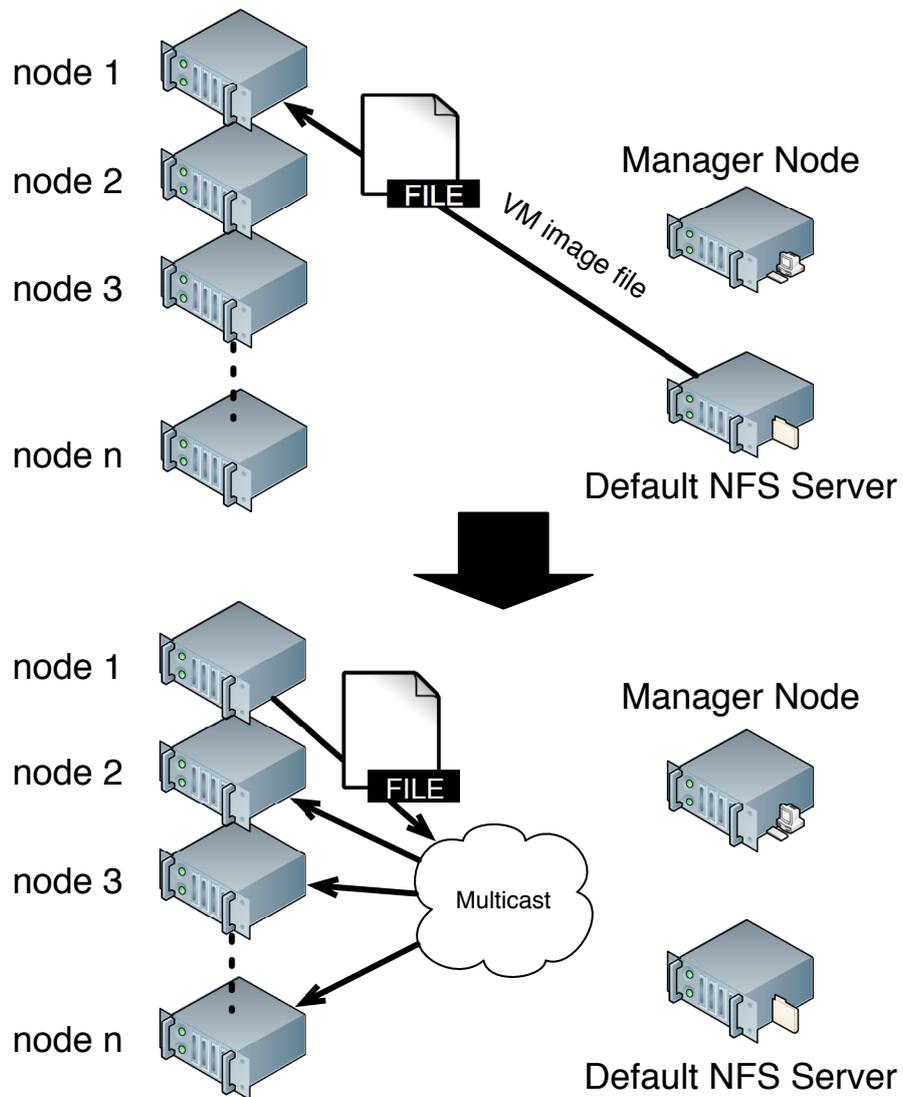


図 6.6: VM image file の配布

- initrd の改造実験用実ノードの起動時に NFS サーバの IP アドレスを Manager Server から取得するために、カーネルロード後に Manager Server に Telnet を用いて、問い合わせる処理を追加した。また、実ノードが複数のネットワークインタフェースを持っており、Linux のデバイス認識順序によっては、カーネルロード後 PXEBoot に利用しているインタフェースを正常に選択できない場合があったため、起動に用いるネットワークインタフェースの MAC アドレスを initrd に引数で渡せる様に起動オプションを追加した。同様に Manager Server の IP アドレスも initrd の引数として渡すように起動オプションに追加した。これによって柔軟に Manager Server を指定可能になった。

6.5.2 管理系サーバ

実証検証環境構築を行う上で必要な管理系サーバである、DHCP サーバ、TFTP サーバ、NFS サーバは StarBED で提供されているサーバを利用した。

6.5.3 Node Agent および Manager Server の実装

- 実験者、Manager Server、Node Agent 間の通信方式

Node Agent および Manager Server は、実ノード上で常駐するサーバデーモンとして実装を行った。通信の信頼性が必要であるため、実験者と Manager Server の通信、Manager Server と Node Agent の通信は全て TCP を利用して行うよう実装した。また各実ノードの RMI への通信、Node Agent への通信は並列で処理を行うが、並列処理を行うスレッドは最大 32 スレッドまでとし、スレッドの生成は 1[s] ずつ間隔を入れて生成する。これは 6.4.4 節の電源管理の並列化でも述べたが、あるイベントが短時間で集中して発生した場合の障害を回避するための処置である。

- マルチキャストによるファイル配布の流れ

仮想ノードのディスクイメージは、実ノードにマルチキャストを用いてファイル配布する。ファイルの転送には信頼性を確保するため、パケットレベルでの再送制御を行っている。Manager Server は実験者に指定されたファイルを NFS サーバから取得するようコマンドを送信する。その実ノードがコマンドを取得完了した

後、Manager Server はそのファイルをマルチキャストで配布するようコマンドを送信する。

6.5.4 計測

作成したツールを用いて、提案したフレームワークの有効性を評価するために、StarBED のノードを利用して実証検証環境構築試験を行った。実験に利用したノードは実ノードとして StarBED の GroupH 230 台、Manager Server を動作させるノードとして同 GroupC 1 台である。

まず、実ノードの起動を開始してから全実ノードの起動が完了するまでの時間を計測した。図 6.7 は時間計測を、実ノード数 1、50、100、150、200、230 台の 6 セット、各セット 10 回ずつ試行した結果である。点は全計測結果であり、線は各セットの平均である。グラフから、230 台規模の実ノードを起動するのに 1400[s] 程度で行える事が読み取れる。実験中に実ノード 4 台が起動せず、2 台の資源情報の不備を発見した。本ツールでは再起動後も起動不良であったノードを故障ノードとして提示する。実験は故障ノードを除外し、予備のノードを利用して継続した。処理時間の増加は線形に推移しており、高負荷による処理遅延などの障害は無い事が推測できる。

次に、起動後の実ノードの初期化と、仮想ノードのディスクイメージの配布にかかる時間を計測した。実ノードの初期化は、仮想ノードを起動するために必要なブリッジインターフェースの設定、仮想ノードのディスクイメージを保存するために、ローカルディスクのパーティションの設定とフォーマットする処理である。配布に用いたディスクイメージは KVM で起動可能な Windows XP で、10GB のディスク領域を確保して作成した。そしてイメージを tgz 形式に圧縮した。ファイルサイズは約 2.7GB である。実験に利用した実ノードは起動実験と同じである。各セットでの実験を 3 回行った。図 6.8 にその結果を示す。結果は 230 台の初期化とファイル配布に 1800[s] 程度要する事が示された。実ノードの初期化は、実ノードの台数の増加に伴いほぼ線形に推移しているのに対し、ディスクイメージファイルの配布は台数の伸びに伴い時間の伸びが大きくなっており、台数が増えるとパケットロスなどによる再送制御が多くなってきていると推測される。

これらの実験結果から、本フレームワークを用いて環境を構築することで、230 台規模の実証検証環境構築が自動で行える事が示唆された。各ノードでは 6 台程度の仮想ノードが起動可能であり、実ノード内でのディスクイメージファイルの複製と起動を行えば 1200

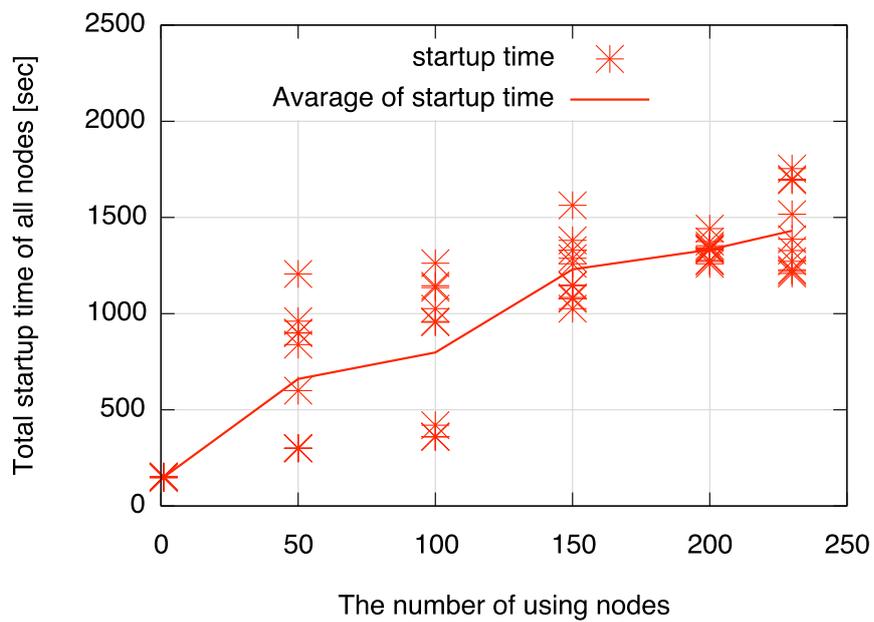


図 6.7: 実ノード起動時間

50台では起動ご NFS Server として追加されるノードのが少なく、NFS Server の負荷分散が十分に機能していないため、起動時間にばらつきが生じている。しかし、実ノードの台数が増えるにつれ、負荷分散の効果上がる為、徐々に起動時間の分散が減少している傾向が見て取れる。

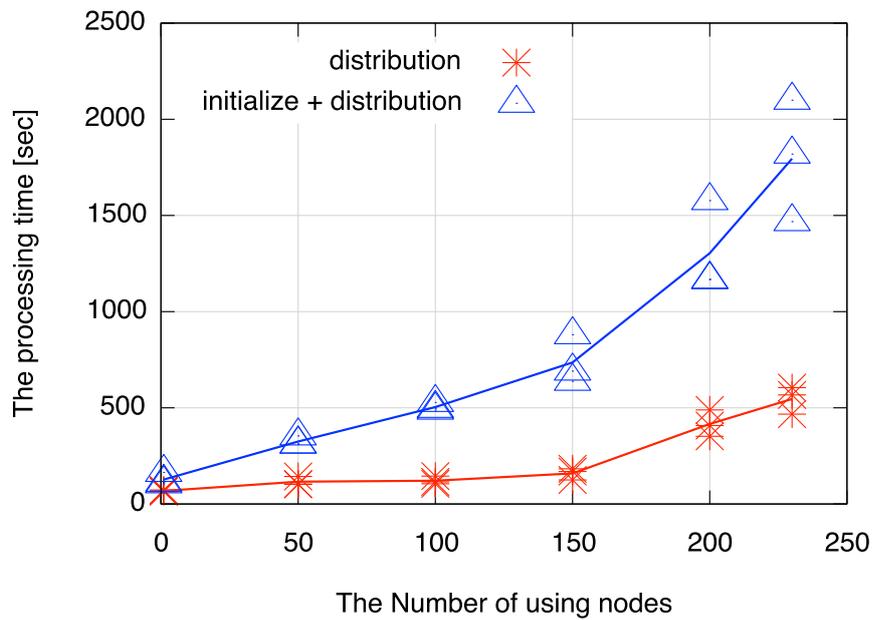


図 6.8: 実ノード初期化と仮想ノードのディスクイメージ配布時間

150 台以下のノードでは Multicast によるディスク配布が有効に機能している為、全体の環境構築時間はノードの起動時間と同様な傾向となる。しかし、200 以上では Multicast の再送要求の増加が確認され、ディスクイメージの配布にかかる時間が増加している。

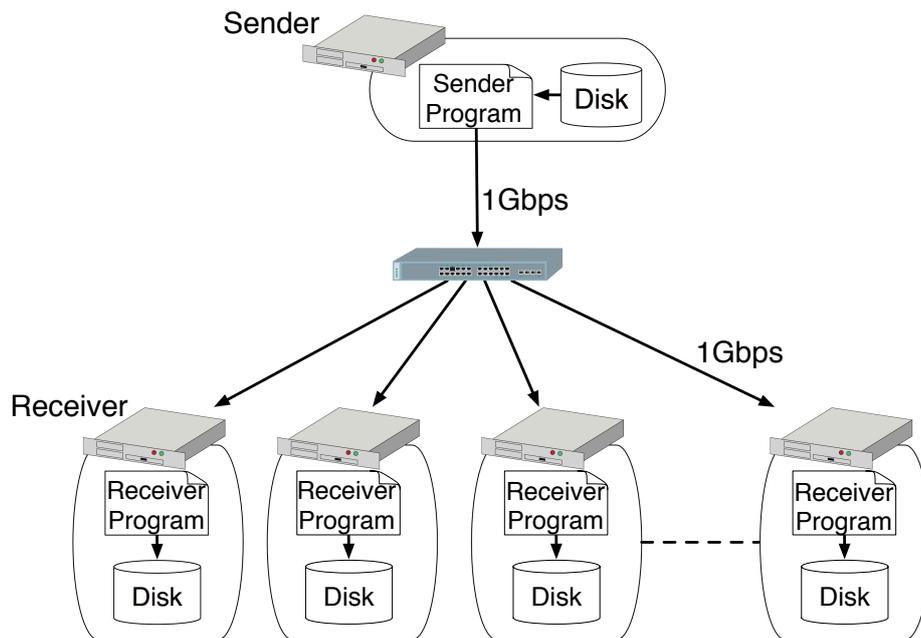


図 6.9: マルチキャストにおけるデータ転送の流れ

台以上の仮想ノードによる実証検証環境が自動構築可能である。

6.5.5 ディスクイメージ配布のボトルネックに関する検討

環境構築のフレームワークを評価した際に、マルチキャストを用いて仮想ノードのディスクイメージを転送した際に、再送制御の負荷が大きくなり、配布台数が増加に伴い転送時間が大きく伸びる現象が発生した。そこで、信頼性マルチキャストを導入する際に発生する問題とその原因を議論し、そして StarBED のような大規模なネットワーク実験環境において信頼性マルチキャストが有すべき機能を検討する。

再送要求の原因

再送要求増加の原因となるパケットロスがどこで発生するのか、図 6.9 のような実験環境を構築し、マルチキャストによるデータ転送時のパケットロスに関する調査する為の計測を行った。計測実験は送信ノードと受信ノードはそれぞれ 1Gbps のイーサネットを実験

表 6.2: 実験条件

Node Group	Group H
No. of Sender Nodes	1
No. of Receiver Nodes	90
Distribute Data Size	100 MBytes
Experiment Node's OS	Debian Linux
Makuosan Version	1.3.1

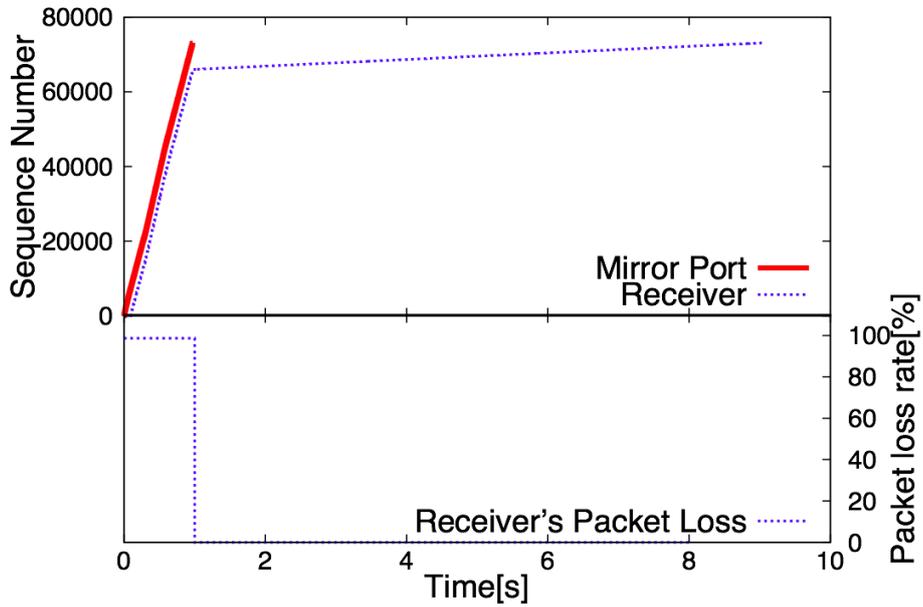


図 6.10: パケットのシーケンス番号とパケット損失率

送信側では約 1[s] でデータの送信が完了している事が解る。しかし、その間は 90% のパケットロス率になる。そして、送信側が送信を終えた後、受信側では約 8[s] 間パケットロス無く残りの 10% 程度のデータが受信される。

スイッチに接続されたスター型のトポロジで行った。送信データは100MBの模擬データとし、受信ノードは90台、送信ノードは1台である。その他の実験条件は表 6.2 に示す。送受信のプログラムは Ruby で実装し、送信するパケットのペイロードサイズは1400 bytes とした。ペイロードにはパケット毎にシーケンス番号を埋め込み、ドロップしたパケットを識別可能にした。実装したプログラムは事前に動作検証し、UDP によるユニキャスト転送で800Mbps以上の送信性能を有することを確認し、またスイッチでパケットドロップが発生しないことも確認した。そして、スイッチに送信ノードに繋がるインターフェースのミラーポートを設定し、トラフィックを計測するノードを接続した。

図 6.10 にその結果を示す。図の Y1 軸は受信したパケットのシーケンス番号を示し、Y2 軸はパケットのロスレートを示す。X 軸は送信開始からの経過時間である。ミラーポートの結果はパケットロスが無かったため記載していない。この結果は、この実験環境の基本的な性能・特性である。送信ノードは全てのパケットを約1秒で送信している。各受信ノードでは90%以上のパケットロスレートと、0.1秒の遅延が観測された。そして、消失したパケットのシーケンス番号は受信ノード90台全て同じであった。この結果は、IP multicast のパケットロスの主要な原因と言え、スイッチの性能・設定が強い影響を及ぼすことを示している。

HDD Writing Speed

スイッチが IP multicast をワイヤーレートで転送可能であったとしても、転送データを Read/Write する HDD の性能が次のボトルネックとなり得る。図 6.9 の様に、全ての送受信ノードは HDD を有している。そして、送信ノードはデータをローカルの HDD から読み込み、受信ノードは受信したデータをローカルの HDD に書き込む。一般に、HDD の書き込み速度は HDD の読み込み速度を下回る。その為、HDD の書き込み速度を超えて、データを受信し続ける場合、受信プログラムはデータを破棄せざるを得なくなる。また、HDD の書き込み速度は HDD の書き込み位置によっても差がある。その為、一般的なコンピュータは HDD の書き込み速度性能を擬似的に高速化/均一化するために一定のバッファを実装している。この書き込み位置、及びバッファの影響を計測した結果が図 6.11 6.12 である。図 6.11 は HDD を 10 のパーティションに分割する事で、ディスク内の書き込み位置を制限し、書き込み性能を計測したグラフである。第1パーティションに対し、第10パーティションに書き込む場合の方が書き込み速度が遅い事が解る。また、書き込み開始後 10[s] 程度書き込み速度が高速に維持されるのが、書き込みバッファの効果である。この結果は、外

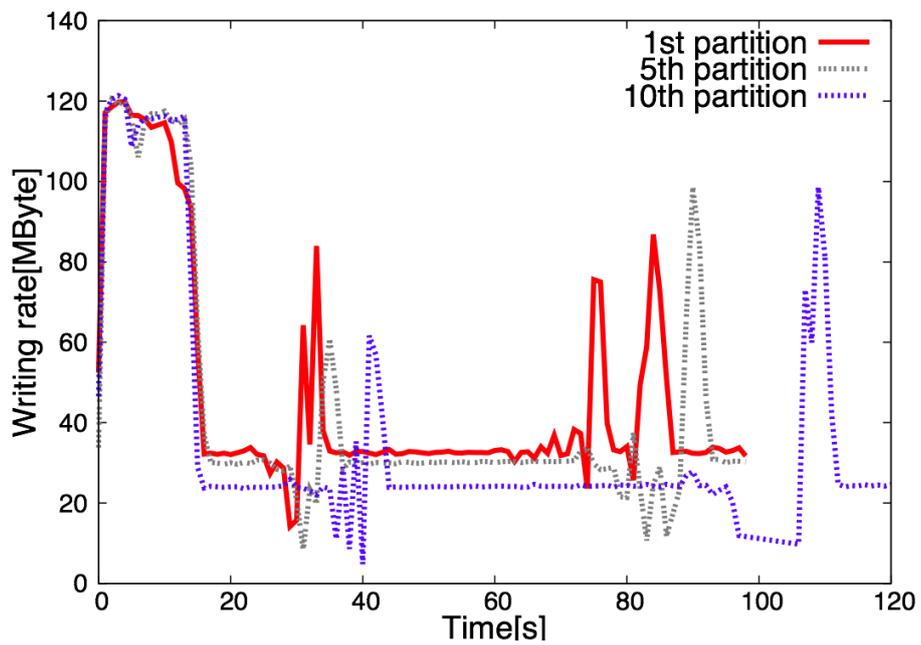


図 6.11: HDD の書き込み位置毎の書き込み性能の時間遷移

同じ容量のデータをそれぞれのパーティションに書き込み、単位時間当たりの書き込み速度の変化。Ruby のプログラムからデータを書き込む (File.write()) 毎に File.flush() 命令をしている。

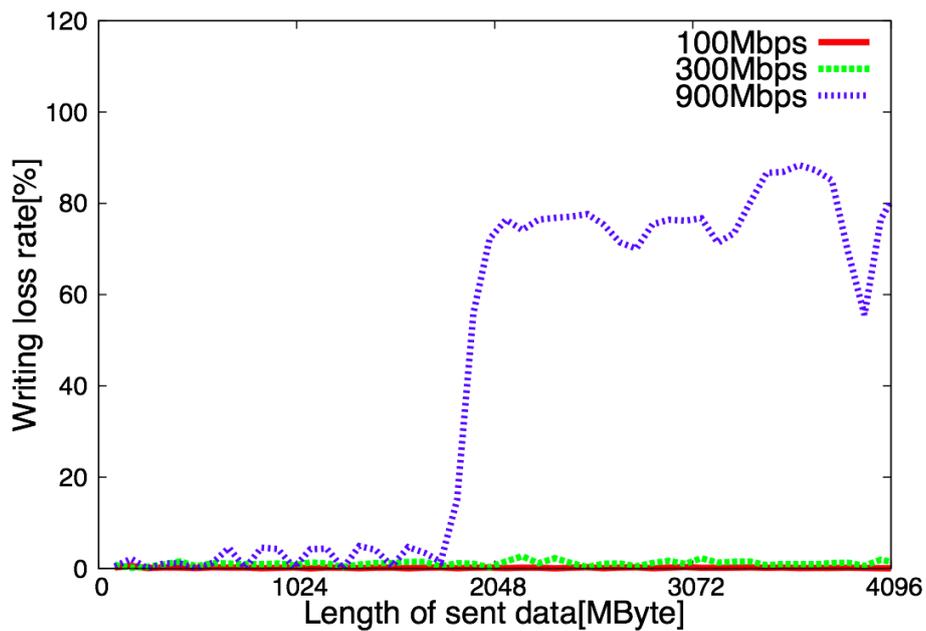


図 6.12: 各転送速度毎の積算書き込み量とデータ損失率

Sender から Reciver までそれぞれの転送速度で 1 パケット 1000Byte のデータを送り Reciver は受け取ったデータを File.write() メソッドで書き込み、直後で File.flush() メソッドを呼んだ場合のデータ損失率 (パケット単位でのドロップ率)。横軸は積算転送量で、各転送速度で割るとに時間軸になる。900Mbps で転送した場合、約 1.8GByte 程度転送した箇所から損失率が急激に悪化している。

部観測的には書き込み速度は時間変化する可能性がある事を示唆する。そこで、Multicast で利用される UDP を用いて送られて来たデータをディスクに書き込む際に、バッファオーバーフローが発生する際の時間変化を計測した結果が 6.12 である。100/300/900Mbps で送信されるデータを逐次 HDD に書くプログラムを用いて、HDD に書き込めなかったデータ損失率の時間変化を計測した。900Mbps で送信しても、1.8GByte 程度まではデータ損失率は高くないが、それ以降急激にデータ損失率が向上する。これは、データの受信側の見かけ受信性能が時間変化する事を示唆する。

6.5.6 実証検証環境構築におけるデータ配布手法に関する議論

前項までに、OS ディスクイメージの様な大容量のデータを効率的に配布する場合のボトルネックを述べた。本項では、この結果から大規模実証検証環境で必要とされる Reliable Multicast のアーキテクチャの議論をする。

ディスク書き込み速度を考慮したデータ配布

StarBED の様な物理的に 1 カ所に集中して設置され、均一なネットワークを持つ実証検証環境と、PlanetLab の様な地理的に分散して設置された、不均一なネットワーク環境で構成される実証検証環境とでは、データ配布の際のデータ欠損の原因が異なる。

近年の Ethernet の主流である 1Gbps の Ethernet はデータの転送速度が一般的な HDD の書き込み容量を上回る。10Gbps Ethernet の普及状況を見ても、今後も HDD の転送速度と Ethernet の速度の不均衡は解消されないと考えられる。この為、本質的に Multicast の様な UDP を用いるデータ転送にはパケットロスによるバースト的なデータ欠損を防ぐための帯域制御機構が必要となる。StarBED ではネットワーク機器の特性により、Multicast を利用した場合に Ethernet のリンク速度以下であってもバースト的なデータの欠損が発生した。StarBED の様な単一または少数のネットワークスイッチにより構成されるネットワークでは、この様なバースト的なデータ欠損において、クライアント側で同一のデータを欠損する可能性が高いことが解った。このような環境では SRM [35] の様な、クライアント同士で欠損データを補完し合う技術は有効に機能しない。その為、ネットワークスイッチの特性と HDD の書き込み速度の両方をトリガとした帯域制御技術が必要となる。また、HDD の書き込み速度は書き込みバッファの存在から時間変化が有る。そして、HDD の書き込み速度は各ノードがデータを HDD のどの領域に書き込むかによっても速度が変わる

ため、帯域制御は事前計測で得る固定的な値でのシェーピングではなく、動的な制御を必要とする。

第 7 章

汎用的な構造指示ファイルによる視覚的な 実験環境構築

大規模な実証検証環境を最初から構築する事は、その環境に不慣れな利用者にとっては実証実験自体が障壁となってしまい、開発の停滞を招く。ソフトウェアの検証に際しては、アプリケーション・システムの設定や規模を変更して複数回の実験を行うこととなるため、特に初期の実験をスムーズにし、実験者の知見蓄積を助ける必要有る。そのため、実証検証環境利用に容易に取り組むためのツール群が必要となる。本章では、実ノードを用いた大規模な実証実験を行うための施設である StarBED において初期の大規模実証検証環境構築向け支援ソフトウェアを検討する。

7.1 大規模実証検証環境の初期利用時の課題

StarBED Project では、StarBED に存在する実験資源を操作し実証実験を容易に実行するための支援ソフトウェアである SpringOS を開発してきた。利用者は実験を行うために、実験ノードに対象のソフトウェアおよびそれを動作させる環境を導入し、ノードが接続されているネットワークスイッチを設定することで実験環境を構築する必要がある。StarBED では物理配線を変更せずに、VLAN を用いて仮想的に実験トポロジを構築する。現状の SpringOS は図 7.1 の様な CLI ベースのユーザインタフェースツールと実験シナリオ記述言語である K 言語による環境構築と実験シナリオの指示書から構成されている。CLI ツール群の操作方法の習得と K 言語の習得を行う為に、利用者は長時間拘束される。そして、実験環境全体を思い通りに制御出来たのか、想定したトポロジと実験の指示書は一致

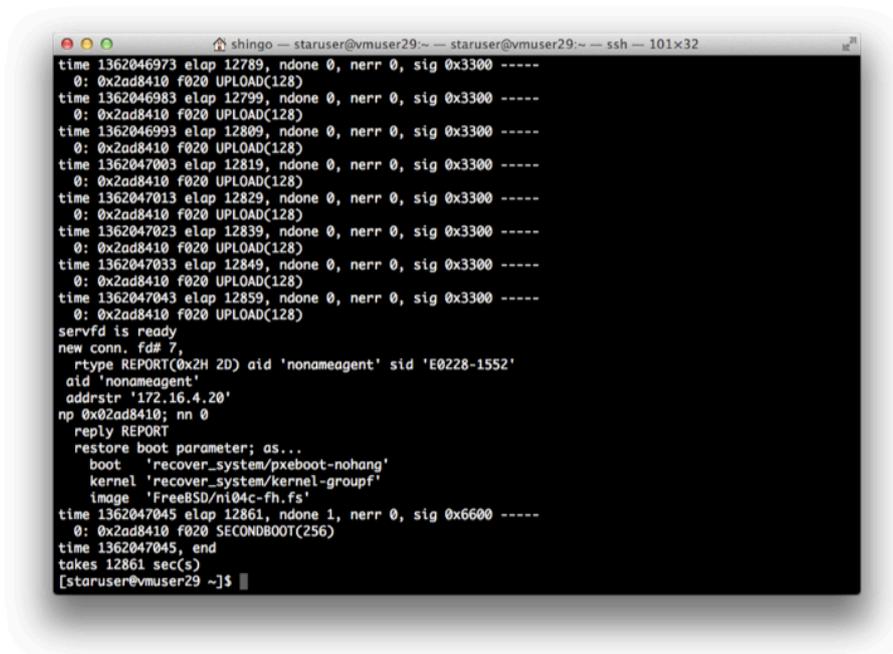


図 7.1: SpringOS で Pickup(ノードイメージの複製) 作業中のスクリーンショット

SpringOS が ni(ディスクイメージ操作用の OS) を起動し、対象ノードのローカルディスクのデータの読み込み・ファイルサーバへのアップロードが完了した様子。

しているのか判断するには時間と経験を要する。

また、現状では各利用者が実験に用いるノードのイメージやトポロジは各利用者が各自から作成する事になっている。通常利用者はアプリケーションの検証や性能の検査が目的であり、その周辺環境の構築に時間をかけることは本質ではない。そのため、利用者が意図する実験環境を容易に構築する為には、基本的な OS イメージやネットワークトポロジを他の利用者と共有化して再利用したり、少ない改変で自身の実験に利用出来るようにする事で環境構築の利便性を向上が可能となる。

7.2 統合実験支援ツール StarBuilder

CLI は原始的である代わりに強力な UI で有るが、操作結果の視認には経験が必要である。近代的なコンピュータでは CLI に代わり GUI による操作を行える様になっているが、大規模な実験環境を GUI で設計、制御するには様々な障害がある。しかし、比較的小規模

Code 7.1: NEED の ROOT エLEMENT 例

```
-<NEED xsi:noNamespaceSchemaLocation="NEED.xsd" ID="StarBuilder-140331" version="1.0">
  +<node ID="root" role="root"></node>
  +<node ID="1" role="emulant" parentID="root"></node>
  +<node ID="2" role="actant" parentID="1"></node>
</NEED>
```

実験に用いるノードを記述するエレメント群からなる。role="emulant"はHyperVisor ノードを指し、role="actant"は、Parent で指すノード ID のHyperVisor 上で動作する VM ノードである。

な実験環境であれば GUI によるトポロジノード構成の記述が行え、かつその様な規模で実験することが多いであろう大規模実験環境利用初心者の利便性が向上する。そこで、GUI により実験環境が構築可能なツールとして StarBuilder を設計、実装した。

7.2.1 実験ノード・ネットワークトポロジの構成記述方式

K 言語は実験環境構築の為の設定群を記述することが可能であるが、各ノードのイメージの配布、ネットワークインターフェースの設定は、設定記述の羅列となり、全体としてのトポロジを把握する事は困難である。また、仮想ノードを実験ノードとして用いて実証実験環境を構築するための拡張は研究段階となっている。そこで、初心者でも可読性が高く、GUI を持ったツールからの変換が容易な記述形式として、StarBED で運用されているマルウェア解析サンドボックスツールである MIMI/MAT [36] の環境記述フォーマットである NEED 形式を利用した。NEED 形式は、XML を用いた環境記述フォーマットで、本来仮想マシンのみを対象とした環境記述フォーマットであったため、今回物理ノードの利用および混在での記述も可能とする拡張を行った。その結果、物理ノード、仮想ノード両方に対応し、ネットワークの記述や、内部へ埋め込む設定ファイルの指定など、柔軟な設定が行う事が可能となった。物理ノード、VM ノードとハイパーバイザの対を記述することが可能である (Code 7.1 7.2 7.3 7.4 7.5)。

Code 7.2: NEED の NODE エLEMENT 例

```
-<node ID="1" role="emulant">
  +<functions></functions>
  +<parameters></parameters>
  +<environments></environments>
</node>
```

ノードが持つ function、parameters、environments のELEMENTからなる。

Code 7.3: NEED の Function ELEMENT 例

```
-<functions>
  <function/>
  <!-- @funcname is a primitive function name -->
  -<!--
    Recommended primitives:
      install_host, _OS, _network, _port, _content;
      build_host, _OS, _network, _port, _content;
      renew_host, _OS, _network, _port, _content;
      barusu_host, _OS, _network, _port, _content;
      cleanup_host, _OS, _network, _port, _content;
      change_host, _OS, _network, _port, _content;
      startup_host, _OS, _network, _port, _content;
      terminate_host, _OS, _network, _port, _content;
      restart_host, _OS, _network, _port, _content;
      check_startup_host, _OS, _network, _port, _content;
      check_terminate_host, _OS, _network, _port, _content;
      collect_data, command;
  -->
</functions>
```

起動/終了などのノードに指示可能な動作コマンド群からなる。外部から指示可能なコマンド一覧を記入する。現在は拡張オプションとしての使い方が記述されている。

Code 7.4: NEED の Parameters エlement 例

```
-<parameters>
  <!--
    parameters elements of the root node shows global
    parameters
  -->
-<network>
  <!--
    network elements shows networks = vlans info.
    although this template shows a fixed network
    structure provided on Builder
  -->
  <manage_vlan>__MANAGE_VLAN__</manage_vlan>
  <experiment1_1_vlan>__EXPERI1_1_VLAN__</experiment1_1_vlan
>
  <experiment1_2_vlan>__EXPERI1_2_VLAN__</experiment1_2_vlan
>
  <experiment2_vlan>__EXPERI2_VLAN__</experiment2_vlan>
  <ipmi_vlan>__IPMI_VLAN__</ipmi_vlan>
  <isolate_vlan>__IPMI_VLAN__</isolate_vlan>
</network>
<parameters>
```

ネットワークインターフェース情報や、所属するべき VLAN ID 等からなる。

Code 7.5: NEED の Environments エlement例

```
-<environments>
  -<!--
    environments elements of the root node shows global
    environments
  -->
  -<!--
    example:
      <ipmicmd>/usr/bin/ipmitool</ipmicmd>
      <ipmiuser>ADMIN</ipmiuser>
      <ipmipass>ADMINPASS</ipmipass>
      <log_dir>~/dataset</log_dir>
  -->
</environments>
```

該当ノードを制御するために必要な環境パラメータ IPMI の PATH、ユーザ名、パスワード等からなる。StarBuilder が動作する環境が複数想定されているため、用意されている Element。StarBED ではユーザがノードの電源制御を行う場合、PWMG 経由でノード貸与時に設定したユーザ名、パスワード、プロジェクト名で電源制御が行われる為、ここではコメントのみで使われていない。

7.2.2 GUIによる環境記述方式

NEED ファイルが用いられていた MIMI/MAT は固定的な環境であったため、これまで NEED ファイルを実証実験毎に動的に生成するツールは提供されていなかった。この NEED 形式を初心者でも使いやすいように視覚的に生成可能なツールの出力から変換プログラムを作成する事で、GUIによる環境記述が可能となる。

そこで、同じ XML をベースとした記述ファイルである vdx 形式を利用する事にした。vdx 形式は Microsoft Visio[®] で用いられているファイル形式で、Visio の他に The Omni GroupTM の Omnigraffle[®] でも利用可能である。Visio および Omnigraffle で利用可能な vdx 形式を利用する事で一般的な PC 用 OS である Windows[®]、Mac OSX[®] のどちらでも GUI で環境記述が可能となる。

7.2.3 SpringOS との連携

今回実装として注力した部分は、GUI で作成された vdx ファイルから NEED 形式へのコンバータ、作成した各エレメントの再利用基盤、および MIMI/MAT ツール群の StarBED 環境への適合である。StarBED でのノードの複製など、基本的な実験支援機能は SpringOS の機能を利用した。具体的には、StarBuilder 全体を Ruby で実装した為、SpringOS との連携は SpringOS の単位機能を各種プログラムから呼び出すためのライブラリである Eggroll の Ruby swig を作成し利用した。

7.2.4 全体設計と外観、コンポーネントの共有

図 7.2 に StarBuilder の全体のフレームワークを示す。利用者は各自の端末で図 7.3 の様に実験環境を記述し、vdx ファイルを生成する。この vdx ファイルを基に StarBuilder は vdx ファイルを NEED 形式へ変換する。その NEED 形式ファイルを基に Eggroll を経由して SpringOS を操作し、環境を構築する。

この時、各実験記述で利用されるノードの設定ファイル、ノードのディスクイメージ、Visio 用のアイコンなど一式をコンポーネントと呼び、StarBuilder の図 7.4 システム上でサンプルコンポーネントとして共有する。各利用者は共有されているコンポーネントを新しく作成する他、既存のコンポーネントをダウンロードして適宜変更を行い、自身の実験に利用することが可能である。

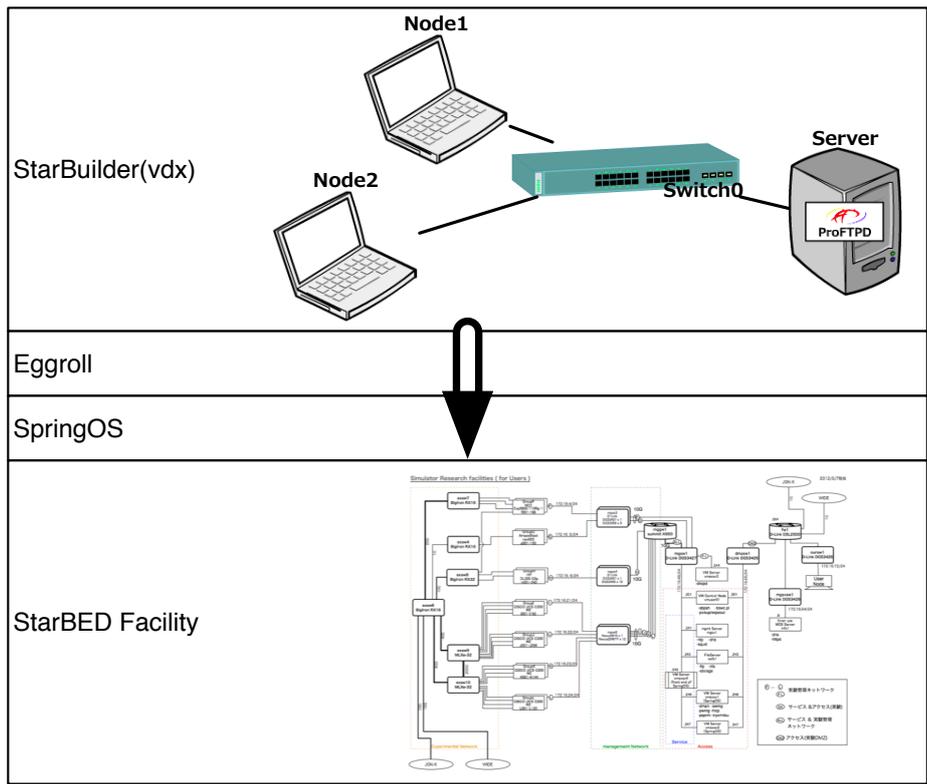


図 7.2: StarBuilder のアーキテクチャ概要

StarBuilder は vdx ファイルを解析し、作成すべきネットワークトポロジ、及びノードの情報を取得する。そして、SpringOS をプログラムから呼び出すための API である、Eggroll を用いて SpringOS と連携し、StarBED の物理環境に対して環境構築の為の変更を加える。

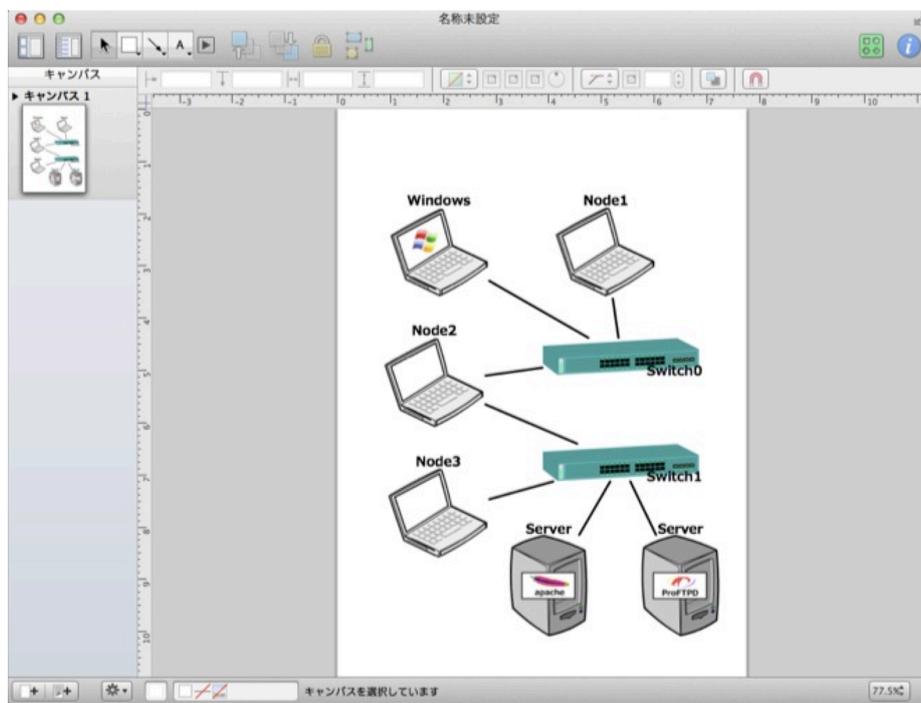


図 7.3: Omnicraft® で実験環境を記述している画面

Switch0 および、Switch1 に繋がる L2 ネットワークが 2 つ、クライアントノードが 4 つ、WEB サーバ、FTP サーバがの計 6 台からなる実験環境を記述した図。Switch コンポーネントは L2 ネットワークを記述する特殊コンポーネントで、ノードでは無く VLAN ID に基づくネットワークを表す。

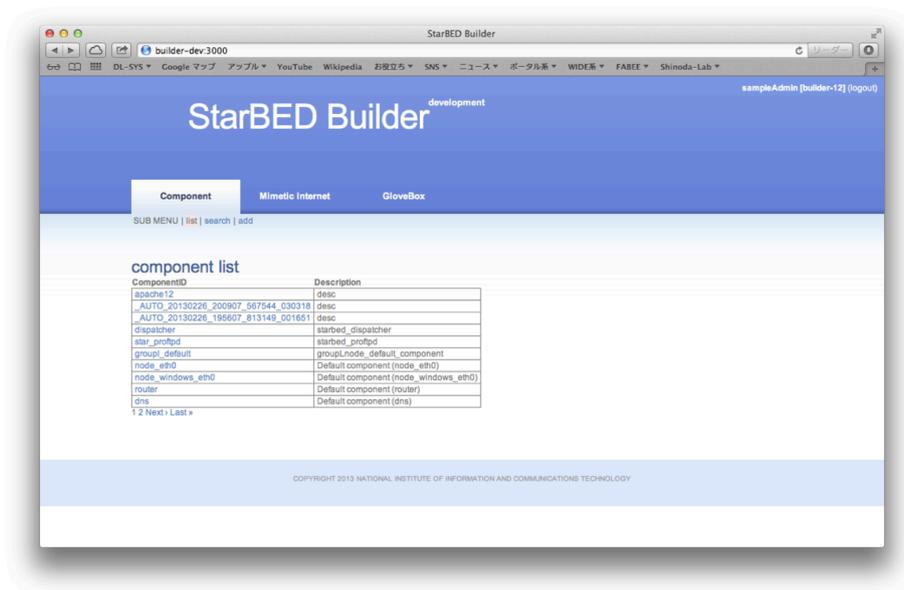


図 7.4: 保存されているコンポーネント一覧画面

StarBuilder のコンポーネントライブラリの表示画面。ユーザは作成したコンポーネントを登録する事で、そのコンポーネントを利用した環境構築が可能となる。同時に他のユーザが作成し他コンポーネントを利用する事も可能となる。ただし、登録したコンポーネントにはアクセス権が設定可能であり、作成したコンポーネントを他のユーザに利用させない (リストに表示されない) 事も可能である。

第IV部

アプリケーション特化センサーネットワーク の設計検証

第 8 章

アプリケーション特化型センサーネットワークの設計検証フレームワークの提案

3章で述べた、野生動物の生態・環境観測ネットワークの設計では、論理的検証段階でデバイスの性能や地形による電波伝播への影響、動物の移動を考慮せずに、最適な設計が行えない事を述べた。この例で影響を受ける設計とは、デバイス間の時間同期手法に関するデバイスノードの役割に及ぶ問題や、情報伝播速度やデータ量、ルーティングの選択に及ぶ。そして設計への影響はパラメータチューニングでは解決できない実装手法、アルゴリズムそのものに及ぶ。これらの技術の取舍選択が最適で無い場合、実証検証段階で解決できない問題が発生しやすく、問題が発生した場合、実証検証段階から論理検証段階への手戻りが発生する。また、車車間通信の検討例では、ノードの移動モデル、密度がネットワークの有効性に大きく影響を受ける事を述べた。これらの想定を、論理検証段階で行い技術の定期用範囲を明らかにしておかなければ、実装後に提案技術が想定したセンサーネットワークアプリケーションに不適合である場合が出てくる。

また、6章、7章では、検証環境、および大規模実証環境で効率的、反復的に検証を行う手法を述べた。小規模な検証環境では、ソフトウェアの機能検証に特化し、大規模実証検証環境ではデバイスの多様性による内乱、論理的検証ステップで考慮した環境特性を再現した外乱を考慮した実証実験により、ソフトウェアの潜在的なバグや想定不備を検証することが望ましい。その為、小規模な検証環境と、大規模実証検証環境では検証すべき項目を明確に分離する必要がある。

ワイヤレステレメータ等のセンサーデバイスの展開と共に規模が拡張するセンサーアプリケーションに対し、IoT時代のセンサーネットワークアプリケーションでは、既に展開し

ているモバイルデバイスなどにアプリケーションを導入する事で計測や情報収集を行う事例もある [37-39]。この様な既に普及し、コモディティ化しているモバイルデバイスを利用したセンサーネットワークアプリケーションは、システムの展開が高速に行える反面、センサーデバイスの多様性を受容し得られるデータの品質を担保する為の、技術検証やデータ解析が必要となる。また、様々なモバイルデバイスが普及し、センサーネットワークアプリケーションのインフラとしての有効性が示されると、これらのインフラを利活用するアプリケーションソフトウェアの開発競争が加速する。その為センサーネットワークアプリケーションの提案から実現までの開発期間の短縮重要となってくる。

大規模なネットワークシステム開発では、対象となる実環境と検証環境との差異を減らし、より現実的な検証を行う事でシステムの妥当性を高めるために、大規模実証検証環境を用いたシステム検証が行われている。大規模実証検証環境は様々なネットワーク環境特性を再現する機能、大規模な実証実験の支援をするための機能を持つ。しかし、これらの機能は汎用的にネットワーク環境特性を再現する機能で有り、特定のアプリケーションに基づくネットワークシステムの開発において重要となる、そのアプリケーションに特化した環境再現手法、つまり環境特性に基づくネットワーク特性を再現するパラメータの決定手法は実験者に任せられ議論されてこなかった。その為、大規模実証検証環境を用いた検証までの各開発のステップで考慮すべき環境特性や、ステップ間で引き継がれるべき特性、注視すべき項目が明確になっておらず、大規模実証検証環境に期待される機能と検証効果が肥大化し、結果として実証実験からの手戻りが発生しやすくなっている。本章ではそれらの課題と提案を整理し、アプリケーションに特化したセンサーネットワークの設計検証フレームワークを提案する。

8.1 ネットワークシステム開発の設計検証におけるフェージング

新しいセンサーネットワークアプリケーションを考え、論理的な設計を行う場合、アプリケーションが対象とする環境、利用するデバイス、適用するネットワーク技術など様々な特性を考慮しない場合、図 8.1 の様にアプリケーションで採用する既存技術の選択、既存技術の改善要件や各種パラメータを適切に選択する事が困難となる。既存システムの改善や新しいアプリケーションを提案し、実際の設計・開発を進めるためには、予算や人員

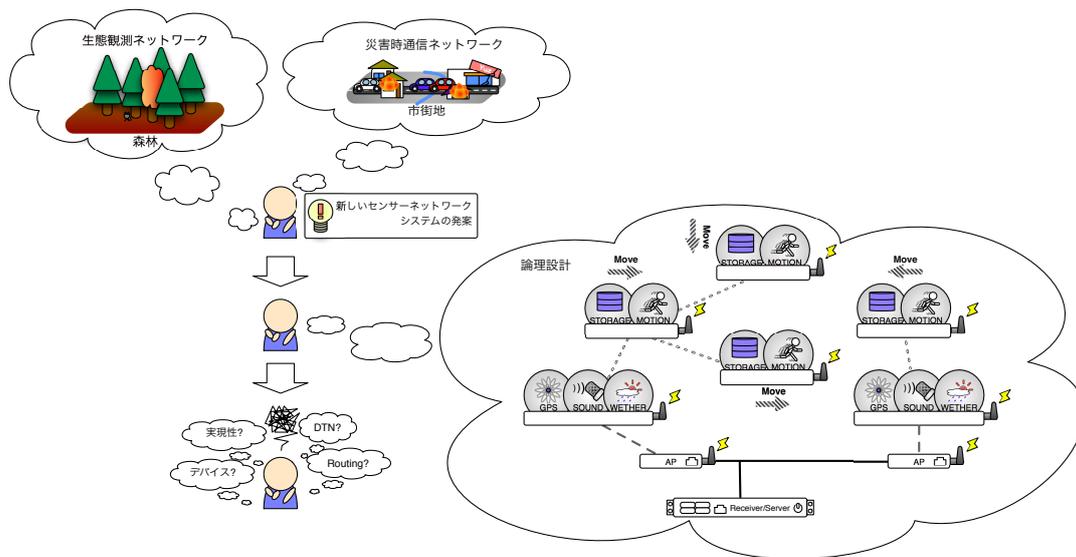


図 8.1: 新しいセンサーアプリケーションの発案

アイデアから、論理的な設計の検討を行う際、システムの外部要因である、環境特性が不明確では最適な設計が困難となる。

などのコストを獲得するためにも、早い段階でそのアプリケーションが実現可能である事、その為に必要となる技術やシステム設計の妥当性を示す事が重要となる。

これまでアプリケーションに特化したワイヤレスセンサーネットワークシステムを設計する為には、大規模実証検証環境を利用した図 8.2 の様な設計段階に基づきネットワークシステムの設計と検証を行っていた。しかし、各段階での確かな設計、検証が行われない場合、実証検証段階で解決できない問題が発生しやすく、問題が発生した場合、図中の赤字で示した設計・開発の段階を跨いだ手戻りが発生する。

設計・検証の段階を跨ぐ手戻りを抑制し、各段階で必要な設計・検証と考慮すべき特性を明らかにするために、本論文では、図 8.3 の様に設計検証の段階を 4 つのフェーズに分類する事を提案する。この 4 フェーズはアイデアの発案の後の最初のフェーズとなる論理検証フェーズから、実装とコンポーネントテストを行う機構検証フェーズ、大規模実証検証環境で実践的な実証実験を行う実証検証フェーズ、実環境へ投入したセンサーネットワークから得られたデータを基に解析を行い、次のアイデアへとつなげる解析/フィードバックフェーズからなる。それぞれのフェーズでは、設計、実装、および検証を行うがそれぞれのフェーズで、手戻りを発生させない為に反復的に検証を行い、各フェーズでの問題の修

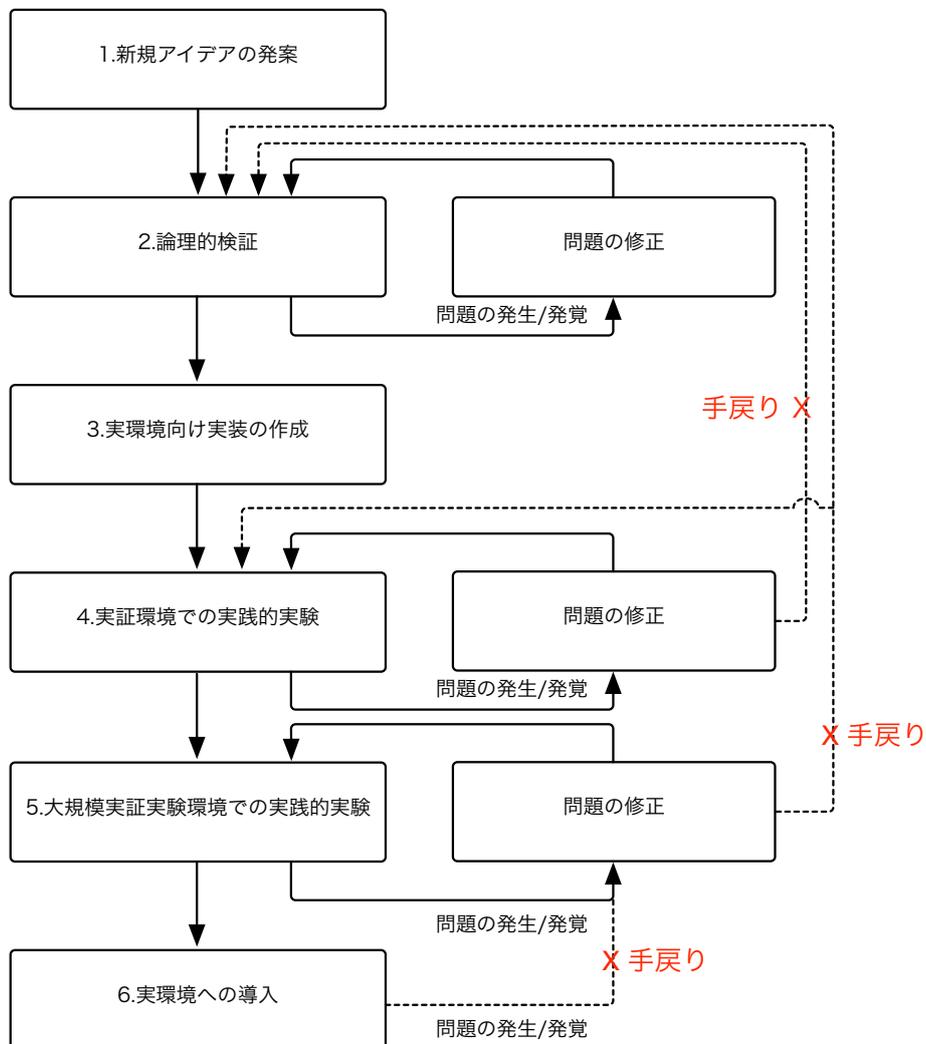


図 8.2: 既存の設計検証手法における課題

各ステップで考慮すべき環境特性や、ステップ間で引き継がれるべき特性、注視すべき項目が明確になっておらず、手戻りが発生しやすい。IoT時代のワイヤレスアプリケーションは実環境への導入後システムを改修することは困難である場合もあり、大規模実証検証環境での実践的実験の妥当性を向上させる必要がある。

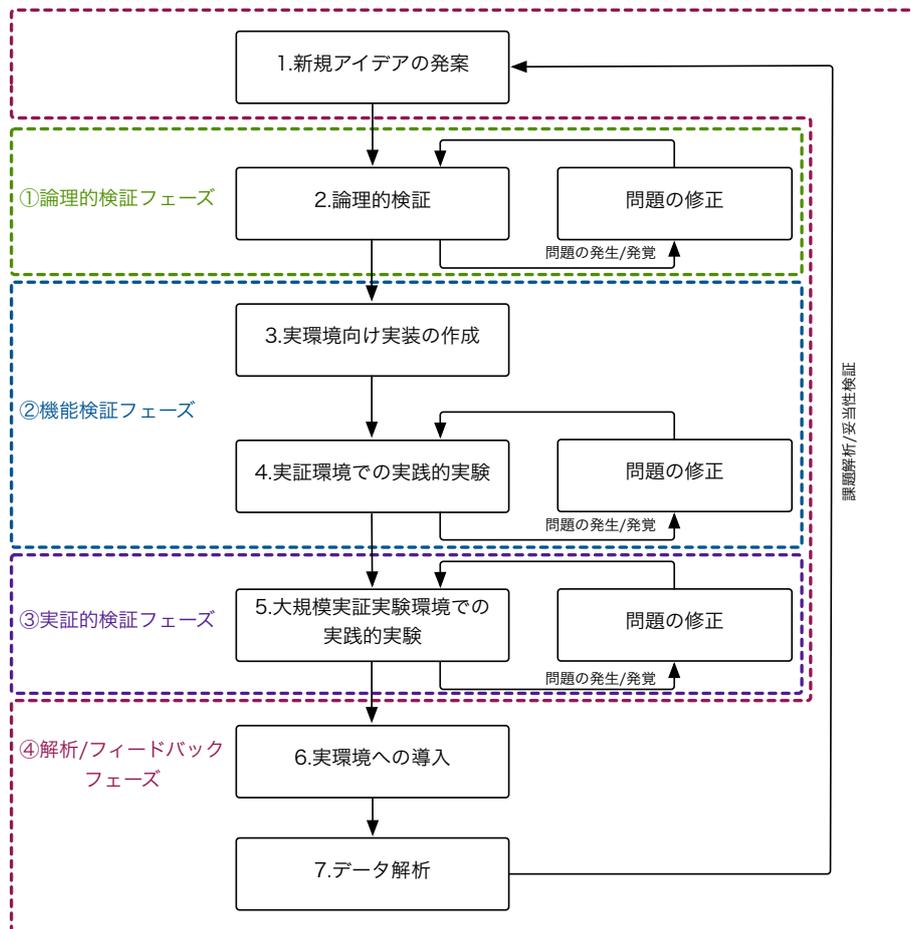


図 8.3: アプリケーション特化型センサーネットワークの設計・検証フェーズ

1章で示した大規模実証検証環境を用いたネットワークシステムの設計検証工程を4つの工程に分類し、各工程で考慮すべき項目、解決すべき課題を定義する。その結果、手戻りとなる工程間を跨ぐ問題の発生と修正を減少させる。また、実運用から得られたデータの解析や課題の検討を通して、新たな新規アイデアの発案や論理検証へとつながるサイクルを構成する。

正を行う必要が有る。その為には、各フェーズで期待する結果も定義する必要がある。

8.2 論理的検証フェーズ

新しいセンサーネットワークアプリケーションを提案した場合、まず、全体設計と技術的な妥当性を検討する為に論理的検証を行う事となる。論理的検証フェーズで行べき開発作業とは、実装物を伴わないシミュレーションベースでの設計の詳細化と、設計の妥当性の検証となる。そして、対象となる環境、デバイスの特性を設定し、それにより必要となる技術の検討を行う。

技術の検討を行った結果、問題が発生/発覚した場合は、その修正を行う事となるが、その修正対象は提案したシステムの構成、デバイスの役割と多岐にわたる。これらの変更は、実装にも大きな影響を与えるため、デバイスの役割、システムの全体構成は論理的検証フェーズで十分な検証の上で決定されている必要がある。

8.2.1 論理的検証フェーズで考慮すべき環境特性

論理的検証段階において、考慮すべき重要なシステム・環境の特性は3章の検討事例により、地形と地形による通信への影響、およびノードの移動特性である事を述べた。図8.4は、論理的検証段階においてこれらの物理特性により影響を受ける特性の関係性を示している。想定環境でのノードの移動特性は、ノードの役割や電波の伝播特性、すなわち通信環境の特性に影響を与える。同様に地形やセンサーデバイスの機能も通信環境へ影響を与える。これらが、最適に考慮されないまま実装を行った場合、選択したルーティング、デバイスの役割など論理的検証で検討・選択した要素に不備が生じる可能性がある。その結果は、実装、またはそれ以降の段階で解決できない課題として表面化し、論理検証段階への手戻りの原因となる。

しかし、これらの特性を論理的検証段階で考慮する為には、電波工学や、動物や人間の生態学、行動のモデリングなど複数の分野にまたがる横断的な技術を活用した検討が必要となる。

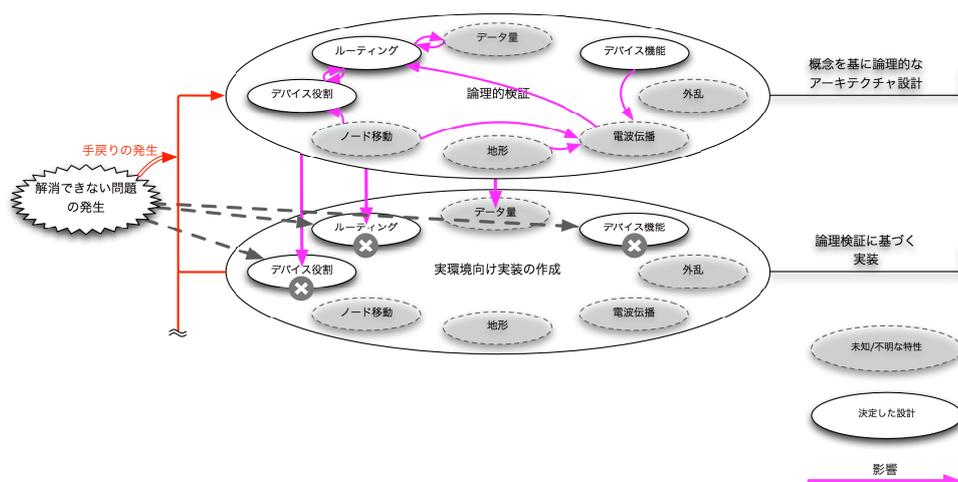


図 8.4: 論理的検証段階における対象環境の特性とその影響

論理設計段階ではルーティングやデバイスの役割、デバイスに求める機能などを設計し、次の実装段階へと進むが、論理的検証段階で未知/不明であったり、考慮しない特性が多い場合、最適な設計が行われず、以降の開発段階で解消できない問題が発生する可能性が高くなり、結果的に開発コストが増加する。

8.2.2 ノードの移動特性の推定

ネットワーク技術の一般的な開発で用いられている移動モデルは、例えばオープンソースのネットワークシミュレータである NS-2 [40] では、Random walk と Random Waypoint を、商用シミュレータである QualNet [41] ではこれらに加え、group mobility や trace mobility に基づく動きが選択出来、利用されている。

ノードの移動データは、昨今のビッグデータに代表されるユビキタス環境の普及に伴い、様々なデータが収集可能となってきた。システムの想定環境としてノードの実移動データがある、または収集可能であれば、それらを論理検証段階から積極的に利用すべきである。

実データの入手が困難な場合であっても、アプリケーションの詳細な仕様、ノードの相互作用がある程度推定可能な場合は、MobiREAL [42, 43] などにより、詳細なノード移動をシミュレート出来るツールを利用する事も可能である。MobiREAL では各ノードがアプリケーションやノードの周辺環境から得られる情報を基にそのノードの行動を変化させる設定を記述可能としている。

しかし、論理的検証段階では、ノードやソフトウェアの挙動として、相互作用まで要件

定義が詳細に検討されていない場合も多い。その場合は、移動モデルを利用したノードの移動を採用する事になる。ノードの移動モデルは、人間行動学や、動物行動生態学、統計学などの研究で様々なモデルがある。それらの中から可能な限り想定するノードの移動に近い移動モデルを採用すべきである。

8.2.3 長期生態観測システムにおける野生動物の行動推定

動物をノードとした場合の移動モデルは、種別による違いも大きく、単一の最適なモデルは無い。ノードの移動をシミュレーションする場合、一般的には単純な線形移動や Random Walk 等が用いられる。しかし、生物の移動においては Random Walk とは乖離が有り、実際の行動データの解析からいくつかの動物は Levy Walk [44] に従う傾向があると報告されている。人間においては、帰巣行動（帰宅）があるため、帰巣行動を Levy Walk に組み込んだ Homesick Levy Walk [45] に従う事が報告されている。そこで、本稿ではセンサーネットワークの各種パラメータの決定を行う為に既知の推定移動モデルとして、ねぐらを持たない動物は Levy Walk, ねぐらを持つ動物は Homesick Levy Walk を用いた移動推定を検討した。

Levy Walk は1回の直線的な移動距離を L としたときに、 L の分布 $P(L)$ が、式 8.1 の様な、べき乗則に従うとするモデルである。この時の移動軌跡は図 8.5 のようになる。しかし、実際には動物の行動範囲には限りがあり、また種別によって行動範囲も異なる。そこで、本稿では1回の目的地までの移動距離 L を式 8.2 を用いて生成することとした。ここで、 M は最大値、 N は最小値、 $rand()$ は $0 \sim 1$ の一様分布乱数関数とする。

$$P(L) \propto L^{-\beta}, 1 < \beta < 3 \quad (8.1)$$

$$L = (rand() \times (M^{-\beta} - N^{-\beta}) + N^{-\beta})^{-\frac{1}{\beta}} \quad (8.2)$$

Homesick Levy Walk の場合、図 8.6 中央のように目的地の決定時に一定確率で固定の場所 (Home) が目的地となる。そして、活動時間終了後にもねぐらへ帰巣する。野生動物の場合はねぐら以外にも水辺、餌場等の固定の立ち寄り場所が存在すると考えられる。その場合、図 8.7 様な複数の固定地点を Home として設定するモデルと考えられる。設定した Home 等の固定地点以外の目的地はランダムとする。

Levy Walk

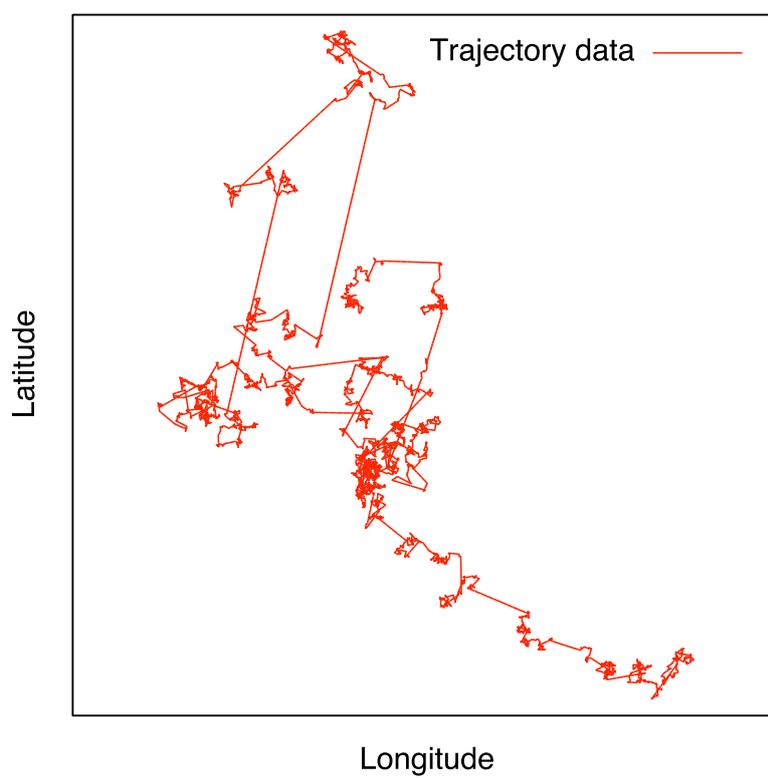


図 8.5: Levy Walk での移動軌跡

Levy Walk は Flight 距離の分布に注目した移動モデルであるため、目的地の方角をランダムに与えた場合はどこに向かうか不定。Random Wait Point に比べると移動/探索と見られる移動特性を持つ。

Homesick Levy Walk(1 home)

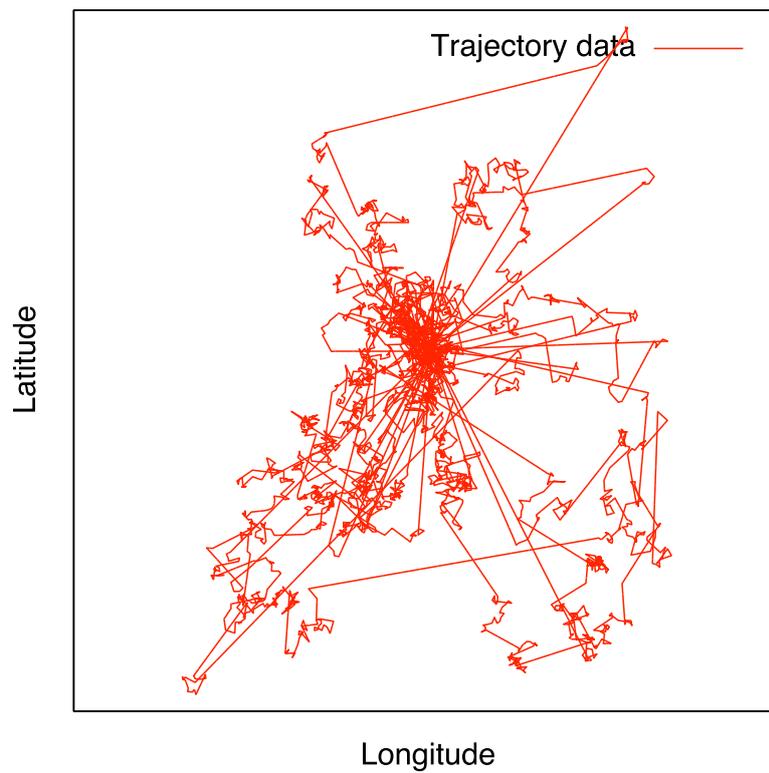


図 8.6: Homesick Levy Walk での移動軌跡

単純な Levy Walk に比べ、一定確立で Home に戻る為、Home を中心とした移動と探索を行うモデルとなる。

Homesick Levy Walk(2 home)

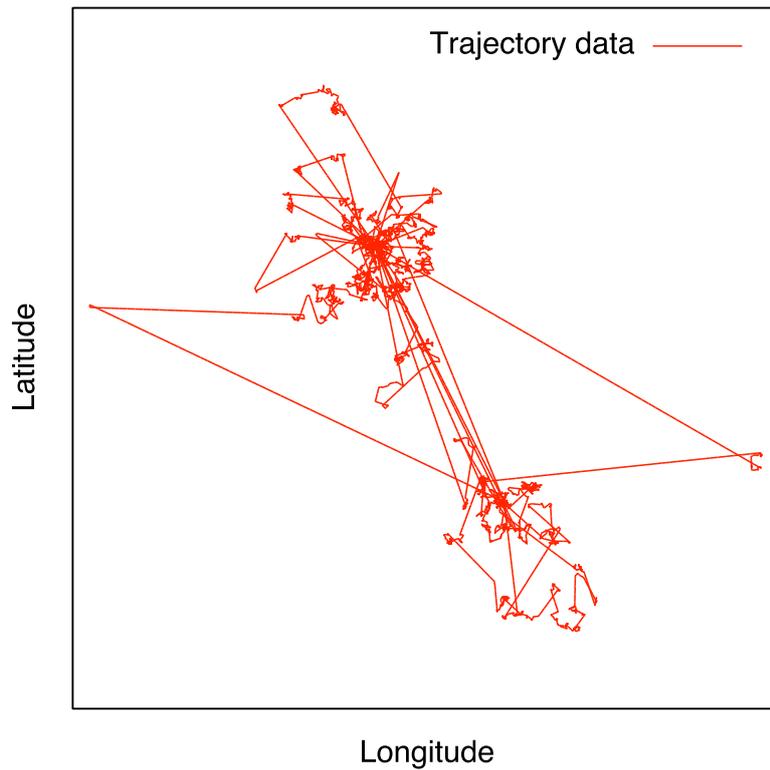


図 8.7: 2つの Home を設定した場合の Homesick Levy Walk の移動軌跡

人間においては、家と職場、動物にはねぐらと餌場や水飲み場といった、特定の立ち寄り場所を複数指定した場合の移動モデル。このグラフでは2の立ち寄り場所 (Home) を同確率で選択した場合の移動軌跡を表している。立ち寄り場所が明らかである場合、このような条件追加により実際の移動結果に近づくと考えられる。

8.2.4 データ伝播モデル

データの伝播モデルは、フラッシングでデータ転送した場合、以下のように定義される。総ノード数を N とする、ある時刻での情報未保持ノード数を S 、情報保持ノード数を I 、消失ノード数を L とする。消失ノードとは、故障や観測圏外へ離脱したノードである。このとき、式 8.3 が成り立つ。情報があるノードで生成された場合の情報伝播による、情報未保持ノード数の変化、情報保持ノード数変化、消失ノード数変化は、それぞれ式 8.4–8.6 となる。ここで、 γ は単位時間あたりの他ノードとの接触確率と接触時の情報交換成功率の積であり、接触時の情報交換成功率を α 、ノードの他ノードとの単位時間あたりの接触回数を k として式 8.7 として得られる。移動モデルから求められる接触回数と、接触時間の平均値を基に”伝播力”、すなわち通信速度や通信成功確率などのパラメータバランスが、技術要件としてもとめられる。

$$N = S + I + L \quad (8.3)$$

$$dS/dt = -\gamma \times (S - S * \theta) \times (I - I * \theta) \quad (8.4)$$

$$dI/dt = \gamma \times (S - S * \theta) \times (I - I * \theta) \quad (8.5)$$

$$dL/dt = (S + I) \times \theta \quad (8.6)$$

$$\gamma = \alpha * k / (S + I) \quad (8.7)$$

8.2.5 ノードの移動・情報伝播モデルを用いた論理的検証

センサ・通信デバイスを用いて野生動物の長期生態観測を行う為には、消費電力の最適化によるデバイスの長寿命化が必須であるが、その為にはデバイスのスリープ間隔や、首輪・ねぐらの各デバイスの役割の最適化を行う必要がある。そのためのパラメータは、対象となる動物の密度、移動モデルによって異なるため、生態が未知の動物であっても、明らかになっている特徴に基づいたシミュレーション、エミュレーションを基に決定する必要がある。本章では、WAOC Network のパラメータチューニングを行うための知見を得るため、既存の移動モデル、動物の生息密度を考慮した DTN の情報伝播特性の検証を行った。

想定フィールドのパラメータ

8.2.4 で検討した移動モデルに基づき、各ノードのすれ違い・情報の伝播特性を検証するためには、生息密度 (単位面積あたりのデバイス装着個体数) や獣道利用の有無なども重要なパラメータになる。獣道を利用する動物では、市街地の移動モデル同様に”道”が行動範囲を限定するため、すれ違いが発生しやすく情報伝播の効率が向上する。それに対し、獣道を利用しない動物の場合はより広い範囲を移動するため、すれ違いが発生する確率が低下する。本稿では、アライグマの生態観測を行っている野幌森林公園地域 [46] を参考に、シミュレーションを行う各パラメータを設定した。想定観測エリアの広さは、 $5km \times 4km$ の平地とした。本稿ではアライグマを想定としてシミュレーションを行う為、獣道はパラメータとしては用いない。生息密度は、アライグマの低密度状態と定義される密度である $5頭/km^2$ で全生息数は 100 頭と想定した。

シミュレーション

前節までに検討を行った移動モデル、各種パラメータに基づいて動物の空間移動シミュレーションを行った。シミュレーションに用いたパラメータを表 8.1 に示す。一日の活動時間は 12 時間とし、それぞれの個体の活動時間はおおむね一致していると仮定し、活動開始から 12 時間経過した次の目的地決定時には目的地をめぐらとする設定にした。アライグマのオスとメスでは、行動範囲の大きさに差があることから、オス 50 頭、メス 50 頭を想定しそれぞれ 1 回の目的地計算時の最大長をオス $2000m$ 、メス $1500m$ とした。移動速度は毎分 $1m$ の一定速度とした。また、シミュレーションは、生息密度に対するデバイス装着個体数の割合による変化も見るために、デバイス装着動物個体数を 100, 50, 25 頭の 3 種類を想定した実験を行った。各デバイス装着動物のめぐらには、めぐら基地局が設置されている事とし、全体のノード数は 200, 100, 50 ノードである。ノード間の通信が成立 (接触) する距離は 4.1 節の計測結果から、 $100m$ とした。

シミュレーション結果

シミュレーションにより得られた 100 頭の首輪ノードそれぞれのめぐらノードと併せて 200 ノードの移動履歴 1 日分と 14 日分を描画した物が図 8.8 と図 8.9 である。図中の円とその中心点は、個々の動物のめぐらとめぐらを中心とした半径 $100m$ の円を示してい

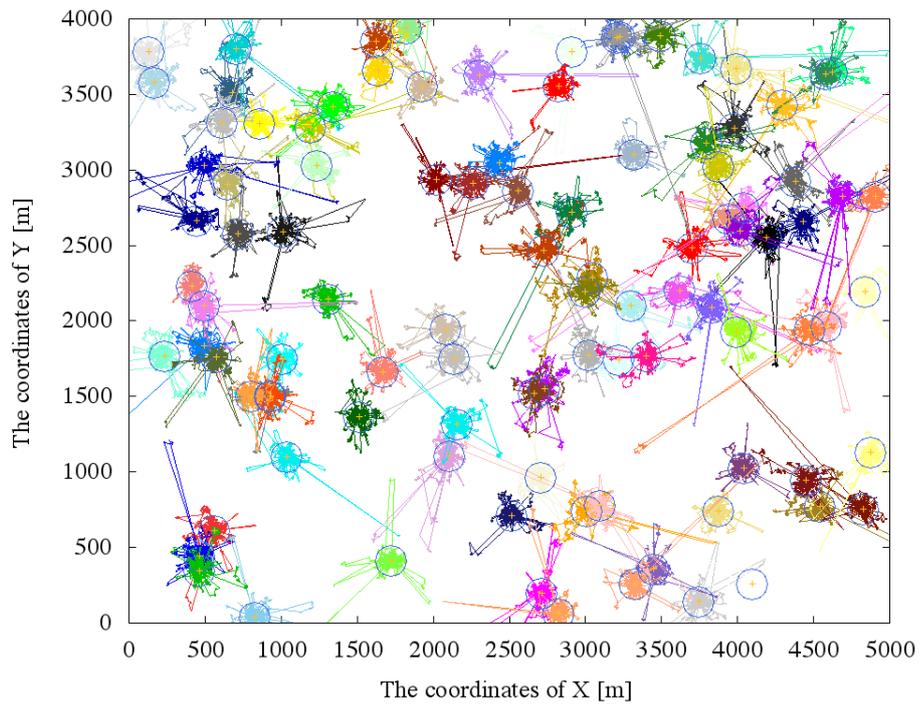


図 8.8: 200 ノード (首輪/ねぐら各 100) 時の 1 日間のノードの移動軌跡

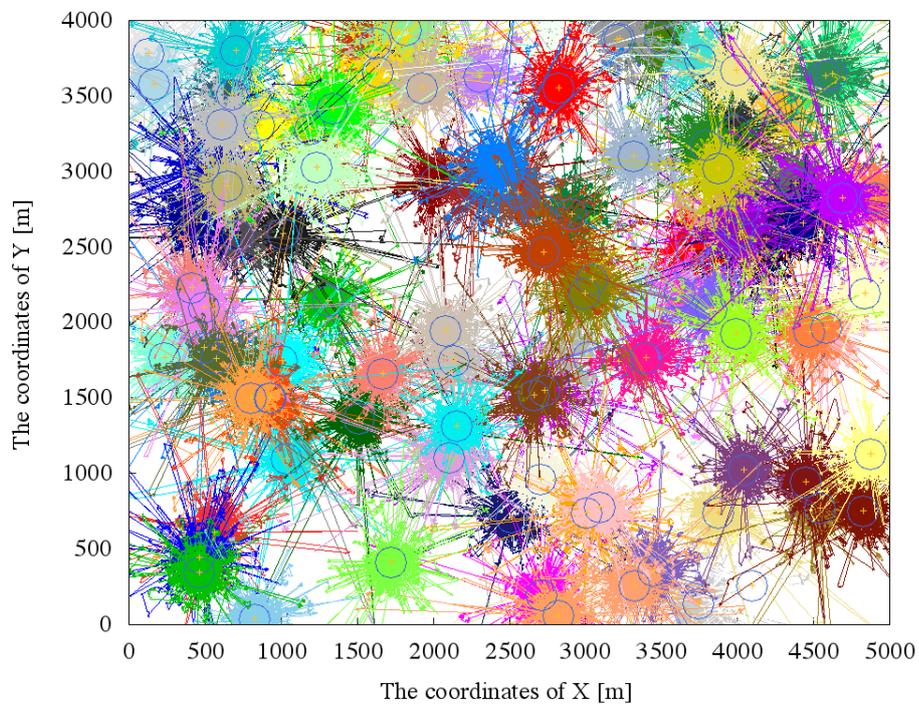


図 8.9: 200 ノード (首輪/ねぐら各 100) 時の 14 日間のノードの移動軌跡

表 8.1: シミュレーションパラメータ

Walking Model	Homesick Levy Walk $\alpha : 0.01$ $\beta : 1.5$ max(MALE):2000m max(FEMALE):1500m min : 1m speed : 1m/sec
Number of Animals	100 (MALE:50, FEMALE:50)
Cocoon positions	Random
Area size	2000 ha (5km × 4km)
Time	14 days Active Time: 12 hour / day Rest Time : 12 hour / day

る。1日でも他の動物・ねぐらとの接触がある事がわかり、14日では十分な接触が発生することが確認できる。

同様に、動物 50 頭、25 頭の 14 日分の移動履歴を図 8.10、図 8.11 に示す。そして、動物同士の接触、動物とねぐらとの接触回数を計算した物が表 8.2 である。ねぐらの平均接触時間、平均接続時間は首輪ノードを装着した動物自身のねぐらにあるねぐら基地局との接触時間は除外している。

対象地域に生息する 100 頭とそのねぐらすべてにデバイスを装着した場合、各ノードは 1 日平均で約 33 回他のノードのねぐら基地局と接触し、他の首輪ノードと約 41 回接触している。装着率 25%(25 頭)の場合であっても、1 日平均 2 回程度他のねぐら基地局と接触し、約 5 回他ノードと接触する事がわかる。各首輪ノードは帰巢時に必ず自身のねぐらに設置されたねぐら基地局と通信が可能であると考えられる。したがって、他のねぐら基地局との接触は、DTN 上での通信機会として重要で有り、今回想定した環境であれば、十分に DTN 上で情報が伝播する事が示された。

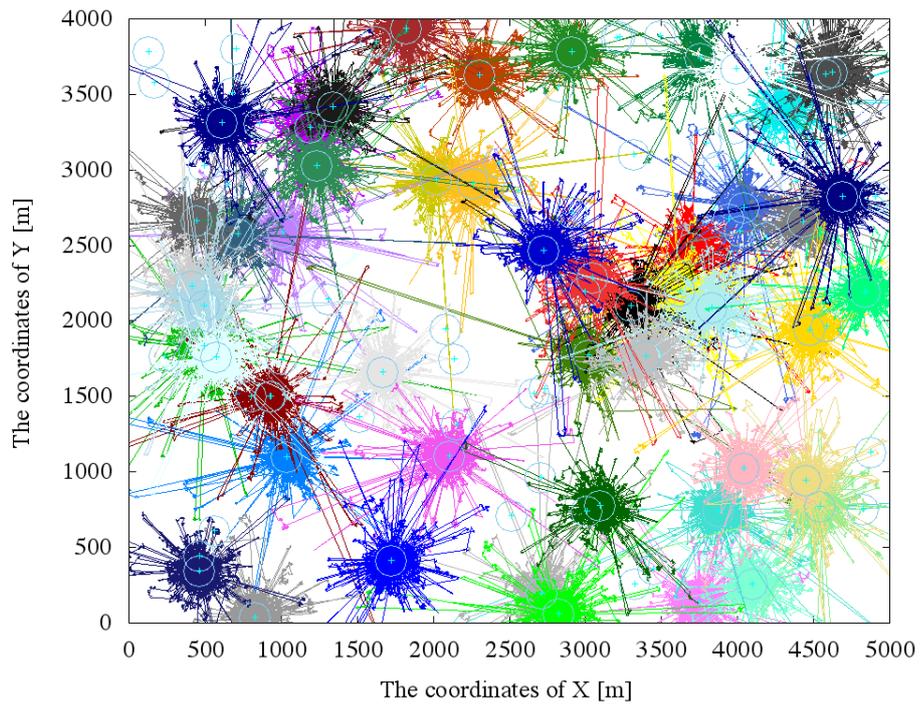


図 8.10: 100 ノード (首輪/ねぐら各 50) 時の 14 日間のノードの移動軌跡

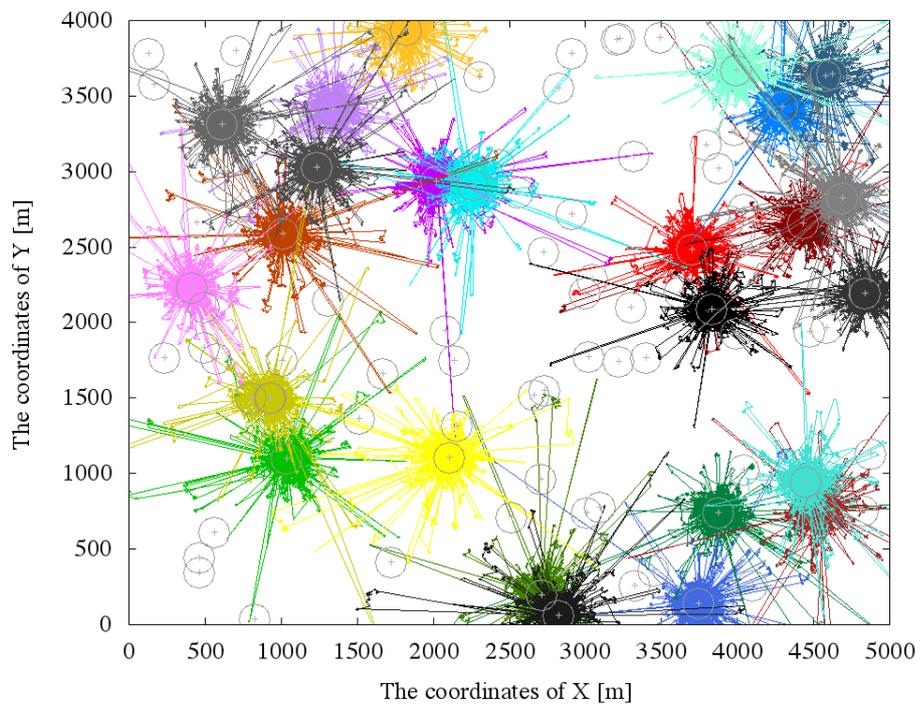


図 8.11: 50 ノード (首輪/ねぐら各 25) 時の 14 日間のノードの移動軌跡

表 8.2: ノード数とすれ違い回数、接触時間の1日平均

Nodes	200	100	50
Node to Cocoon			
Approach Count	33.05	10.15	2.08
Approach Time (sec)	5473.41	2192.5	280.68
Node to Node			
Approach Count	41.23	15.61	4.96
Approach Time (sec)	5584.19	2273.75	545.75

8.2.6 環境特性の積極的推定による設計へのフィードバック

時刻同期手法

シミュレーションの結果から各ノードは1日に数回、自ノードねぐら基地局だけではなく、他ノードのねぐら基地局とも接触することから、今想定した環境では、時間同期は首輪ノード同士では無く、ねぐら基地局からの基準信号のみで可能であると考えられる。また、25頭的首輪ノードとねぐら基地局の環境であっても、首輪ノードが他のねぐら基地局と接触することから、Homesick性が低い、自身のねぐらへの帰巢確率が低い移動モデルを取る生物であっても、時刻同期は可能であると考えられる。

情報の伝播とライフタイム

シミュレーションで得られた結果を基に、WAOCに必要な技術要件の検討を行う。情報伝播速度と、情報のライフタイムに関する考察を行う。8.2.4で述べた伝播モデルに、移動シミュレーションから得られた値を代入した結果が図 8.12である。なお、接触1回あたりの通信成功率 $\alpha = 0.5$ 、平均消失率 $\theta = 0.0003$ (故障率:年5%、域外移動月1頭)として計算している。また、ノード数はねぐらノードを含む200, 100, 50ノードである。

100頭すべてにデバイスを装着した場合2日程度、25頭であっても4日程度で全ノードにデータが伝播する。これらの事から、本稿で想定したエリアとデバイス密度であれば、4日程度のライフタイムで領域内の全ノードにデータが伝播する。これは、情報の到達距離と

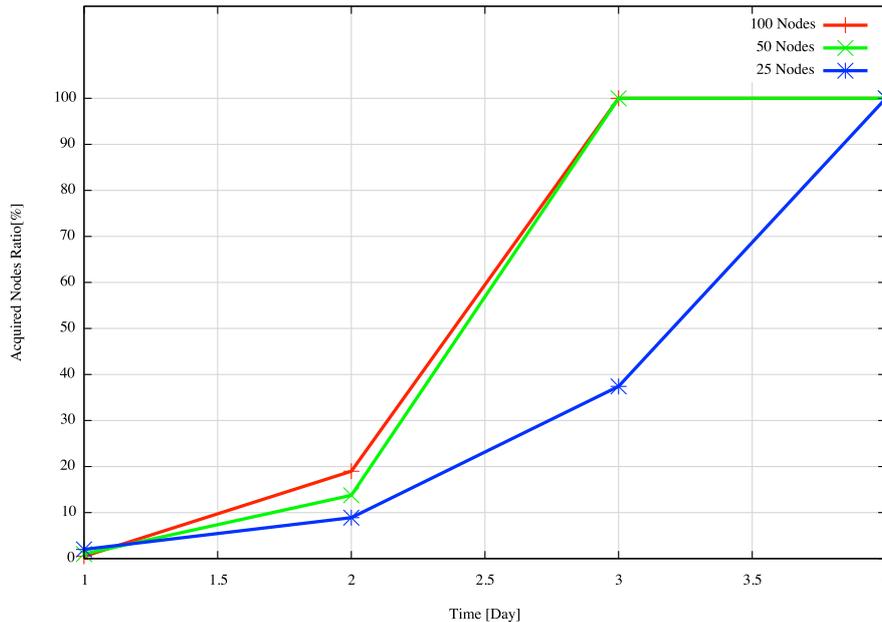


図 8.12: 全ノードに占めるデータ保持ノードの増加の数値計算結果

して見た場合、領域の直径に相当することから、実際にはそれ以下のライフタイムで、对外接続が確保されている領域外縁部のねぐらノードまで、データが到達すると考えられる。

通信速度・ルーティングによる転送量削減

ねぐらノードが生成する環境情報のデータは、気温、湿度、風向などのデータであり、1回の情報量は32bit INT型10種類として40Byte程度、META情報などを加味してもIEEE802.15.4のL2パケット、1パケットで送信可能な100Byte程度である。パケットサイズ上限である100Byteのデータをねぐら基地局が10分間隔で生成し、同様に首輪ノード同士、首輪ノードとねぐら基地局が接触時に接触情報として100Byteのデータを生成すると仮定をして、ネットワーク内で発生するデータ量を計算した物が表8.3である。1日にネットワーク内で発生する情報量は100頭の場合で3MByte程度であり、25頭にデバイスを装着した場合で400KByte程度である。

情報のライフタイムを、25頭の場合でも全体にデータが伝播される時間4日と仮定した場合、1.2MByte～12MByte程度のストア領域が各ノードにあれば、単純なフラッシングであっても、データ転送が可能であると考えられる。このデータ量は、現在のコンピュータが保持するストア領域としては十分に小さい。Arduino等の小型組み込みデバイスでは

表 8.3: 1日に発生するデータ量

Number of Animals	100	50	25
Environment Info (KB)	1440	720	360
Approach Info (KB)	1485.6	257.6	35.2
Total Size (KB)	2925.6	977.6	392.2

不揮発メモリ領域は数十～数約 KByte 程度ある事もあるが、外部記憶等を搭載する事で記憶領域を拡張することも可能である。不揮発メモリ領域は書き換え回数に制限があり、データ量の増大は書き換え回数の増加を招く為、ルーティングの最適化を行い情報量の削減を行う必要があるが、伝播情報量とストア領域の関係だけ見た場合、ルーティングによる情報量削減の必要性は低い。

一方で、今回のシミュレーションでは、各ノードが1日に他の首輪ノードやそのねぐら基地局と通信可能となる時間は100首輪ノードの場合で平均で1日約11058秒、25首輪ノードの場合で約826秒であった。IEEE802.15.4の通信速度は低周波数利用時の実効速度で10Kbps程度とされる。その場合各ノード間で通信可能なデータ量は、1日のノード間転送可能情報量は100首輪ノード時で約13.9MByte、25首輪ノード時で1MByte程度となる。この帯域で全データをフラッディングで転送するためには、各ノード間の通信路で、1日の平均実効通信可能帯域の40～60%前後の通信効率を必要とする。このため帯域から見ると全データをフラッディングで伝播するには帯域が不足しており、帯域から考えた場合、ネットワーク全体に伝播する情報量の削減が必要となる。

これらの結果から、野生動物をキャリアとしたDTNが成立し、センサーネットワークとしての有用性が示唆された。一方で、WAOC Network上で発生するデータ量と各ノード間での平均通信可能帯域を解析した結果、全データをフラッディングで伝播するには帯域が不足しており、データの圧縮、ルーティングによるネットワーク全体に伝播する情報量の削減等、ノード間の通信量の削減が必要となる事が明らかとなった。

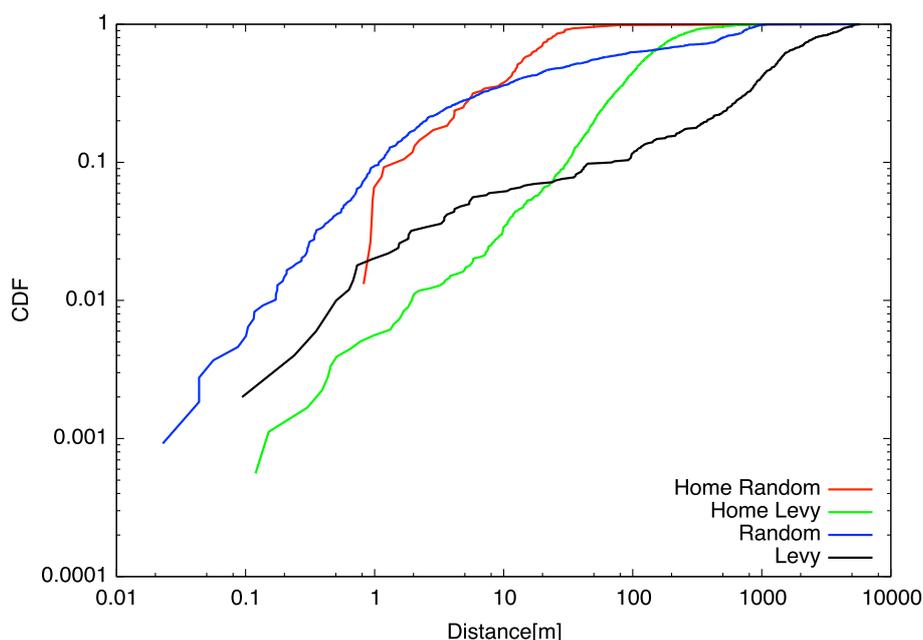


図 8.13: 200 ノード (首輪/ねぐら各 100) 時の全ノード間の推定取得接触データ

収集データからの行動推定

設計したセンサーネットワークが、最適な設計であったか、または観測対象が想定通りの移動特性を持っているかを確認する解析を行った。今回シミュレーションで論理的検証を行った野生動物の環境で、野生動物の移動モデルとして利用した、Homesick Levy Walk に加えて、Levy Walk, Homesick Random Walk, Random Walk の 4 種類ですれ違い情報を解析し、14 日間の Node0 による直線移動距離の分布を CDF で表したグラフが図 8.13 である。次回目的地の Home の選択確率は先のシミュレーションに準ずる。

直線移動距離の定義は、Node0 が他のノードおよび全ねぐらノードのいずれかと 100m 以内に接近していた時間の平均座標が、そのノードがネットワーク的に推定される座標と定義し、次に推定された座標までを直線移動した事とした。横軸が推定直線移動距離で X,Y 両軸とも Log Scale としている。Levy Walk と Random Walk との大きな違いは、flight 距離の分布であり、Random Walk は一定であるのに対し、Levy Walk はべき乗分布となる。図 8.13 8.14 8.15 は、それぞれ首輪ノード数 100,50,25 台と、同数のねぐら基地局を設置した場合に、観測が推定される首輪ノードの Flight 距離の分布をシミュレーションした結果である。図 8.13 を見ると、Homesick Random Walk と Random Walk は弧を描いている

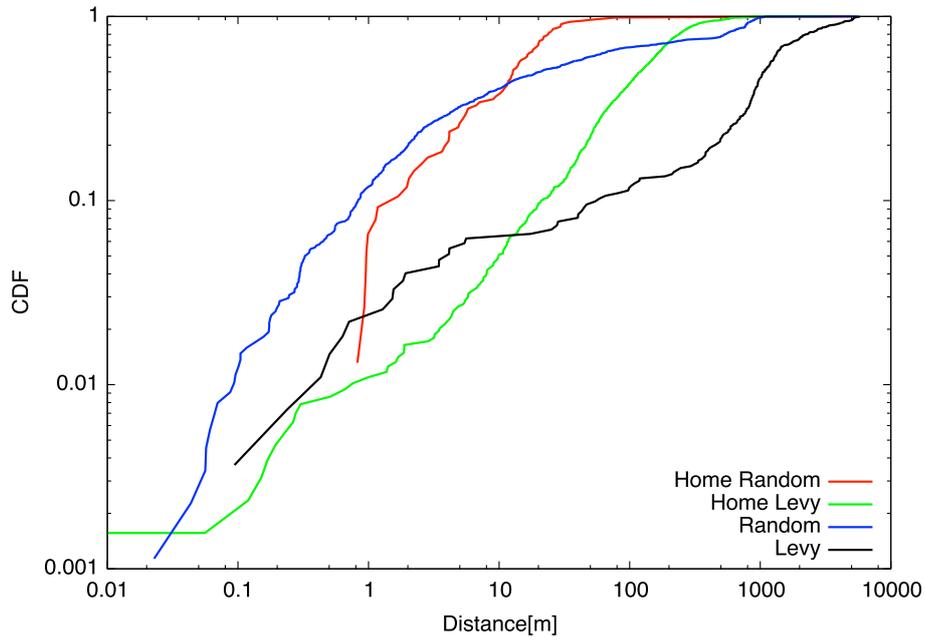


図 8.14: 100 ノード (首輪/ねぐら各 50) 時の全ノード間の推定取得接触データ

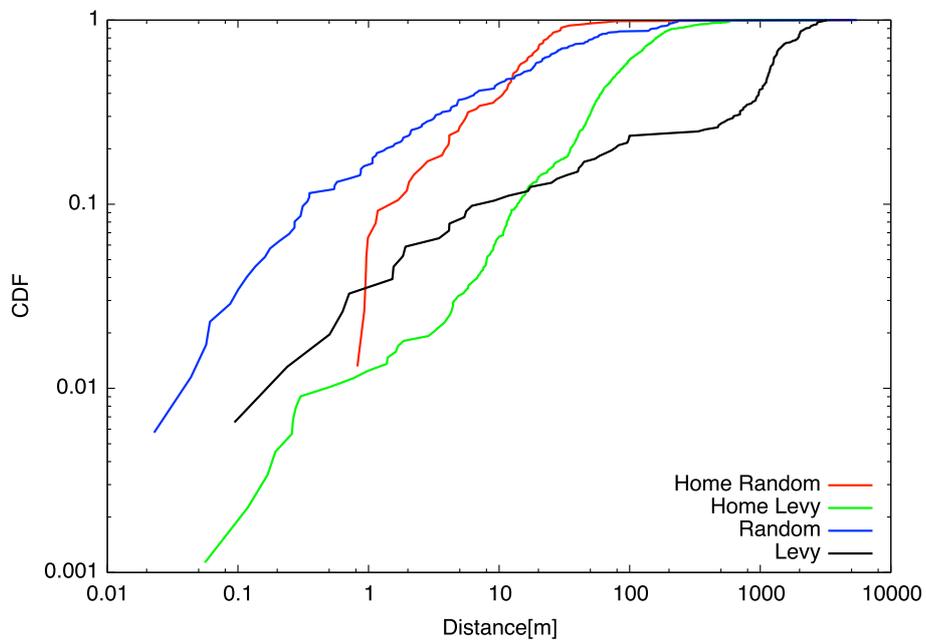


図 8.15: 50 ノード (首輪/ねぐら各 25) 時の全ノード間の推定取得接触データ

表 8.4: 各移動モデルでの Node0 の他ノードとの接触回数

Walking Model	StreetPass Count
Homesick Levy Walk	1789
Levy Walk	501
Homesick Random Walk	76
Random Walk	1087

のに対し、Homesick Levy Walk と Levy Walk は直線的な分布が見て取れる。また、それぞれの Node0 の他のノードとの接触回数を表 8.4 に示す。Levy Walk は Homesick 性を持つモデルでも他ノードとの接触回数が多いのに対し、Random Walk では他ノードとの接触回数が大幅に減少している。これは、Homesick 性を持たせたことで、Random Walk ではノードが自身のねぐら近辺から離れなかった為である。Homesick 性を持たない Random Walk, Levy Walk は帰巢条件の無い自由移動になるため、どちらも他ノードと十分に接触している。この結果は移動がもたらす、ネットワークの特性への影響の強さが示唆されると共に、計測値からも移動モデルと同様の傾向が取得可能である事を示唆している。

また、GPS 機能を保有可能なねぐらノードのみとの接触情報でも同様な、結果が得られるか、首輪ノード Node0 から、他のねぐらノードとの接触のみで同様の解析をしたグラフが図 8.16 8.17 8.18 である。図 8.13 と同様のグラフの傾向が見て取れるが、Homesick Random Walk では、他のねぐらノードと1度しか接触しなかったため、データが推定不可能であった。この結果から、Homesick Random Walk 以外では、ねぐらノードとの接触情報だけでも、今回のノード密度であれば、十分な移動傾向の推定が可能である事が解った。また、取得したデータから、ネットワークが期待通りの性能を示さなかった場合でも、ノードの移動特性を逆推定し、最適化を行う事が可能である事が示唆された。しかし、ノード数が少ない場合、すなわちノード密度が低い場合では、それぞれの推定値の傾向の差が少なくなる事が明らかになった。今回は Random Walk 2種と Levy Walk 2種との差が比較的明瞭であるため、50台の首輪ノードでも想定移動モデルとの一致が推定出来るが、より多くの Walking Model を検討した場合にどの程度のノード密度で分離が可能であるかは、今後さらなる検討と解析が必要である。

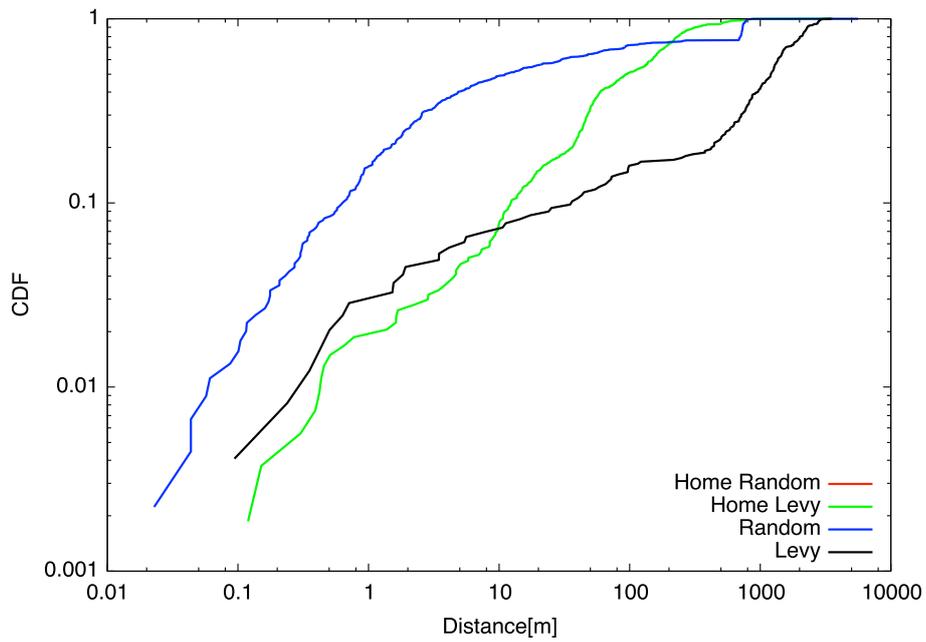


図 8.16: 200 ノード (首輪/ねぐら各 100) 時の首輪ノードねぐら基地局間の推定取得接触データ

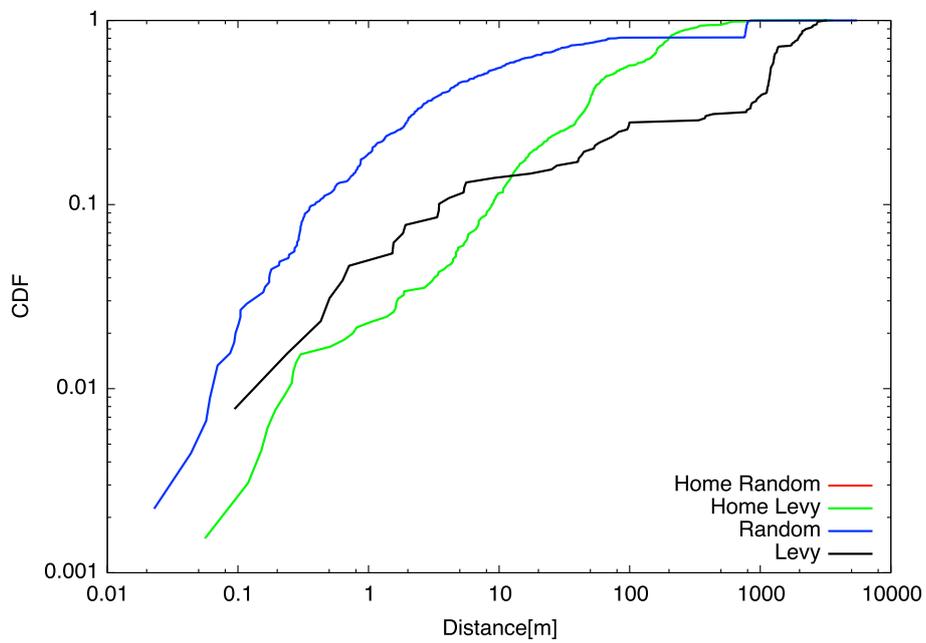


図 8.17: 100 ノード (首輪/ねぐら各 50) 時の首輪ノードねぐら基地局間の推定取得接触データ

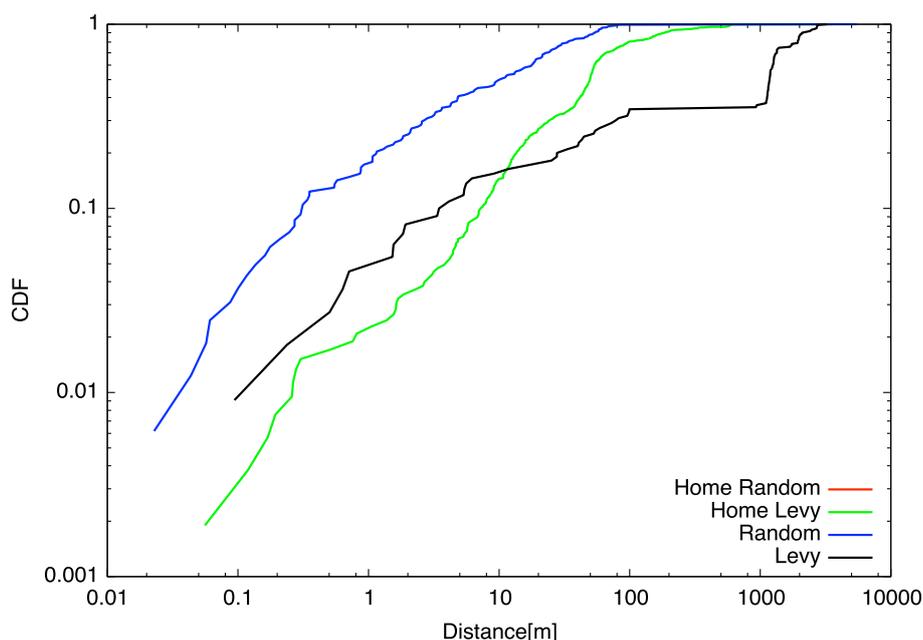


図 8.18: 50 ノード (首輪/ねぐら各 25) 時の首輪ノードねぐら基地局間の推定取得接触データ

8.2.7 環境特性の積極的推定による設計へのフィードバックの効果

論理的検証段階において、前節で述べたようにアプリケーションが対象とする環境特性を積極的に推定する事で、図 8.19 アプリケーションの提案段階では不明瞭であった環境特性を、論理的検証段階で考慮することが可能となり、システム設計の妥当性が向上する。同時に、デバイスが持つべき機能、すなわち対象となるデバイス毎に実装すべきソフトウェア機能、採用すべきルーティングプロトコルを最適に選定する事が可能となる。また、既存のルーティングやデバイスでは、有効なシステムが開発不能である事が判明した場合、研究・開発対象となるデバイスや、ルーティング/通信プロトコルの要件が明らかになる。研究・開発が必要な要素があった場合は、それを研究・開発した上で、論理的検証を再度行う事となる。

8.3 機能検証フェーズ

機能検証フェーズでは、論理的検証段階での検証によりアプリケーションに最適化された設計に基づき、実装と機能検証を行う。一般にソフトウェアやネットワークシステムの

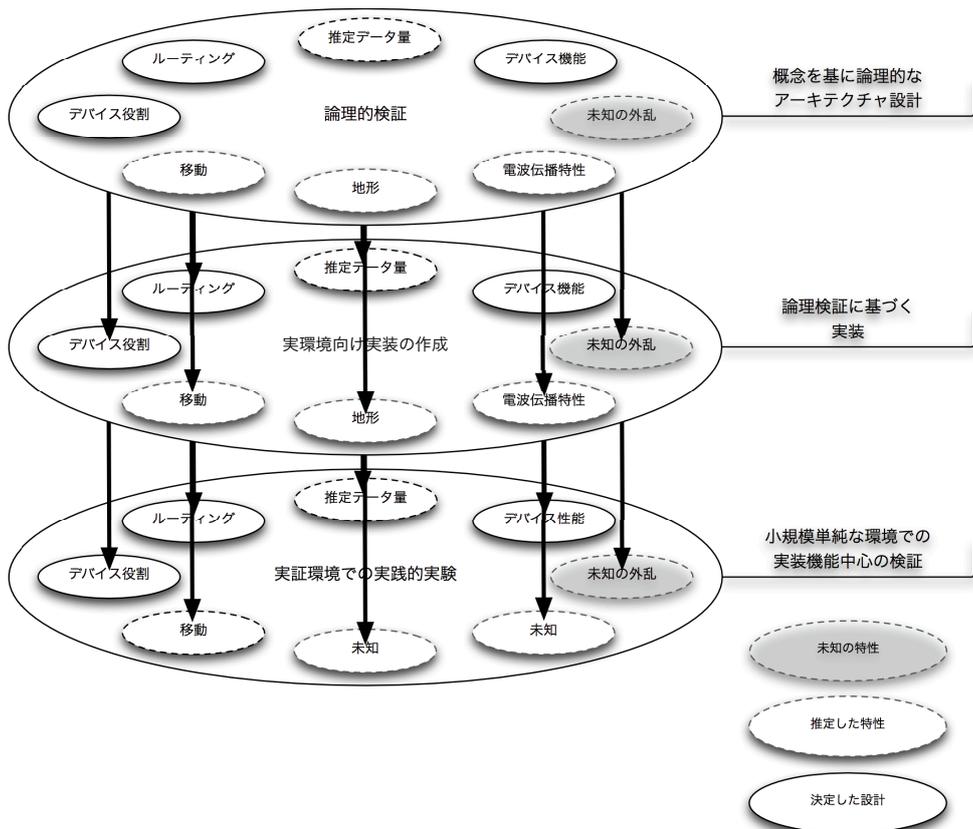


図 8.19: 環境特性の積極的推定による設計へのフィードバックの効果

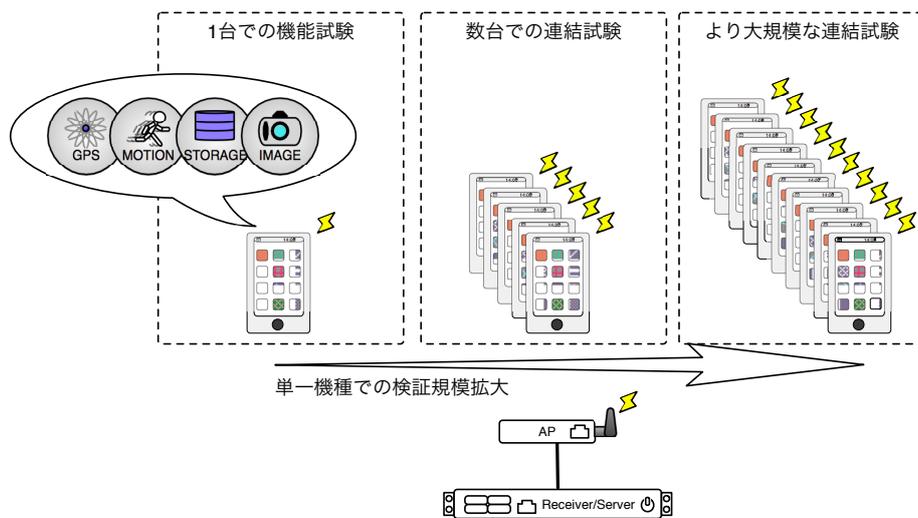


図 8.20: 単一機種の規模拡大による機能検証

単一機種の機能検証を規模を拡大して行う。ノードの複製を作成するだけでよく容易に作成可能だが、ノードの役割分担が有る場合は、単一機種だけでは検証できない機能がある。

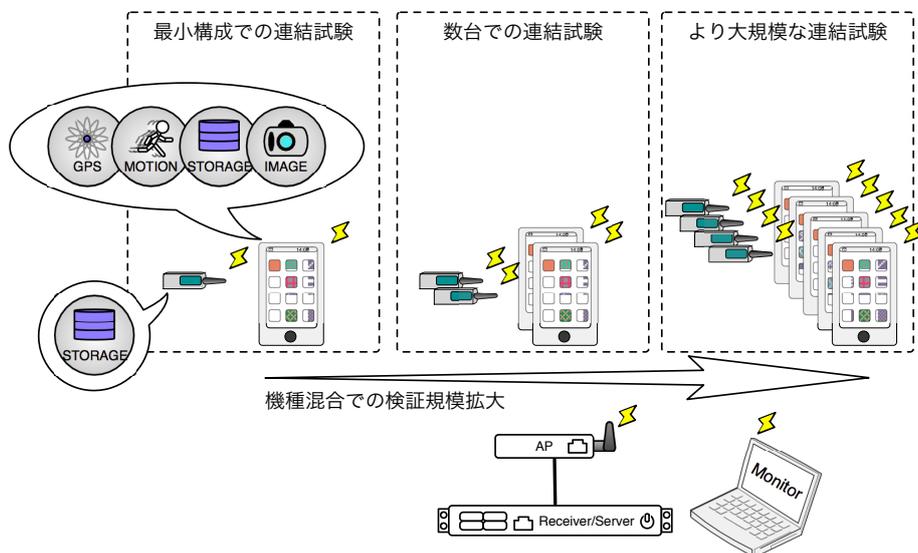


図 8.21: 機種混合での規模拡大による機能検証

機種混合での機能検証を規模を拡大して行う。同一機種のノードの増加は複製を作成するだけでよく容易に作成可能だが、機種の種類毎に作成する必要がある。ノードの役割分担が有っても全システムを構成する機種数で機能検証を行う事で、全機能の検証を行う事が可能である。

機能検証では、単体機能検証、連携機能検証などを行う。IoTによるユビキタス環境で利用するアプリケーションの場合、想定するデバイスの種類が多岐にわたり、場合によってはデバイスの種類毎に実装も異なる事から、デバイスの種類毎の機能検証が必要となる事が多い。そこで、まずこのフェーズでは実装後、図 8.20 の様に単一のデバイスで単体機能検証を行い、次にシステムに必要な最小構成の種類で検証を行い、徐々に種類とデバイスの数を増加させて図 8.21 の様な連携機能検証を行う。この機能検証フェーズでは、設計に基づく機能検証に主眼を置き、ネットワークの構成は、単純で安定した状態で行う。

8.3.1 機能検証に特化した検証環境構築支援ツール

大規模なシステムの実証実験を行う場合、まずは比較的小規模な実証検証環境を構築し、機能的な検証を行った後、様々な環境特性を再現した大規模な実証検証環境を構築し、実践的な検証を行う事が一般的である。しかし、大規模な実証検証環境の構築は作成した環境と、希望した環境の同一性の確認や環境の健全性の確認が難しい。特に初期の段階で検証を行う担当者が、実証検証環境の構築に不慣れであった場合はその影響が顕著になる。また、実証実験ではソフトウェアのチューニングも行われるため、反復的な実験環境の構築、段階的な規模の拡張が行われる。そのような環境構築を柔軟に行う為には、実験環境の構築確度の向上や、環境構築の速度向上が望まれる。本論文では、このような初期の実証実験環境を効率的に作成する為に、一般的な作図ソフトで作成した実験環境の構成図を基に実験環境を構築する StarBuilder を実装した。

StarBuilder では、実験環境構築時に利用するコンポーネントと呼ぶノードのディスクイメージとメモリ量やネットワークインターフェース数、IP アドレスなどの設定情報を共有する事が可能である。その為、実験者は自分が利用したいノードのテンプレートとなるコンポーネントが既に登録されている場合は、自分で OS のインストール作業などをせずに設定情報の変更のみで自身の実験に利用出来る。これにより、実験者は OS イメージの作成のコストを削減できる。この効果は、利用するノードが汎用的な OS で有るほど高い再利用効率を期待出来る。また、StarBuilder では複数のノードを利用する場合、コンポーネントが共通であれば、設定情報を必要台数分入力することで単一種別のノードを指定数作成する事が可能である。機能検証フェーズで行われる単一デバイスでの単体機能検証では、一つのデバイスイメージを作成すれば設定ファイルを用意する事で、規模を拡大した単体機能検証が可能となる。また、複数のデバイスで単体機能検証を行ったあと、そのイメー

ジを再利用し、設定ファイルを修正することで、連携機能検証へスムーズに移行することも可能としている。

StarBuilder のこれらの機能を利用し、機能検証環境を構築する事でより柔軟な検証環境の構築が行える。

8.4 実証検証フェーズ

実証検証フェーズでは、機能検証フェーズで行った、様々なネットワーク構成デバイスでの連携機能検証を踏まえ、デバイスの数を増やし、より現実的な規模での検証を行う。この段階では、比較的小規模で行う機能検証フェーズでは発生しなかった、実装上の問題の発見を目的とする。次に、論理的検証段階で考慮した各種環境特性を再現した、より現実的な環境での実証実験を行う。また、必要に応じて対象技術には直接は関係しない周辺トラフィックなども導入し、想定外の問題の発見や、開発した技術が他のネットワークシステムに与える影響なども調査する。

論理的検証段階で考慮した各種環境特性を再現するためには、論理的検証段階でシミュレーションベースで再現した環境特性を、大規模実証検証環境上でエミュレーションする必要がある。本論文では、IoT時代のワイヤレスセンサーネットワークの設計・検証において、論理的検証段階で特に重要となる環境特性を地形と地形による電波伝播特性、並びにノードの移動特性であると述べた。したがって、大規模実証検証環境にはこれらの論理的検証段階で考慮される環境特性を、エミュレーションする機能が要求される。

8.4.1 車車間通信の地形を考慮したエミュレーションによる実証検証

提案する実証検証環境の有効性を検証する為に、QOMET に地形を考慮した電波伝播特性の計算を行う拡張を行い、それを利用して、災害時情報収集交換ネットワークの例として車車間通信ネットワークシステムの実証実験を行った。

地形情報を考慮した DeltaQ の作成

QOMET を利用したワイヤレスネットワークエミュレーションに地形情報を導入し、電波伝播路における地面による回折ロスをエミュレーションする実装する為に、QOMET の Linux 対応実装である METEOR [47] を基に拡張を行った。QOMET のアーキテクチャは

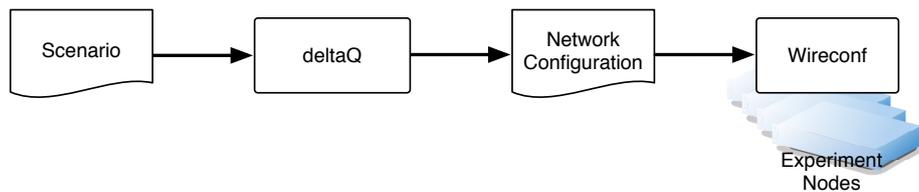


図 8.22: QOMET のアーキテクチャ

QOMET は全ノード座標の時間遷移を記述したシナリオを基に、deltaQ と呼ばれるソフトウェアにて全ノードのペアにおける通信状況の時間変化を計算し、Network Configuration ファイルを生成する。その情報を基に Wireconf が実験開始後、各ノード上でネットワークの状況をエミュレーションする。METEOR では Wireconf の実装が Linux 対応版に置き換わっている。

図 8.22 である。電波の伝播および、それによるネットワークのパラメータの時間遷移を deltaQ が計算をする為、今回 deltaQ に与えられた地形情報に基づき、地面による見通し直線の遮蔽が発生した場合、仮想単一ナイフエッジモデルに基づき回折ロスを計算するようにした。地形情報は Google Maps API [48] によって標高情報を取得する事とした。

災害時情報収集交換ネットワークシステムの実証検証実験

本システムは、自動車に搭載された通信デバイスを用いて、DTN を構成し、情報の収集と伝播を行うシステムである。基本的な、実験ノード構成は 3.3 と同様に、自動車に搭載されたデバイスとして、OpenWrt を用いたノードを仮想マシンとして作成し、DTN Protocol 実装として IBR-DTN を組み込んだ。そして、この仮想マシンを 10 台の物理マシンを用いてそれぞれ 10 多重し、100 台の仮想マシンを作成した。ノードの通信方式としては WiFi(802.11b) を想定した。また、想定地理環境として、本学（北陸先端科学技術大学院大学）近隣にランダムに 100 台の車を想定したノードを設置し、それぞれ Random Waitpoint のモデルに基づき、北陸先端科学技術大学院大学および情報通信機構北陸 StarBED 技術センターをランダムに目的地として設定する事とした。

Reliable Multicast によるディスクイメージ配布

実証検証環境作成時に、実験ノードのディスクイメージ配布を効率的に行う為に、Reliable Multicast による、実験ノードのディスクイメージ配布を行った。同時にノードを配布す

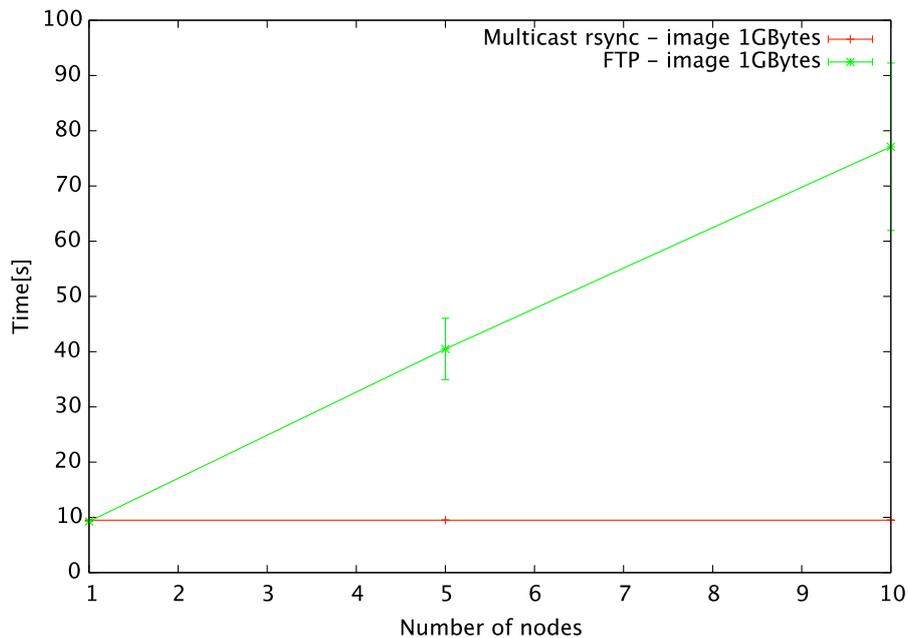


図 8.23: ノードイメージサイズが 1GB の場合の転送時間

るノード数を 1 台、5 台、10 台として、1GB の実験用仮想ノードの配布にかかる時間を SpringOS で標準的に利用されている FTP を利用した場合と比較した。Reliable Multicast によるノードイメージ配布の時間推移結果を図 8.23 に示す。今回の実験では 6.5.5 節の実験と利用するスイッチが異なったため、Multicast 転送でのスイッチによるパケットロスが発生しなかった。また、今回の実証環境のハイパーバイザはテンプレートとなるディスクイメージを memfs 上に所持するため、ストレージの書き込み速度のボトルネックも観測されなかった。その結果、FTP を用いたディスクイメージ配布では、ノード数に対して配布時間が線形に増加するのに対し、Reliable Multicast を利用した場合は、ノード数にかかわらず一定であった。

エミュレーションにおける地形情報の有無によるネットワーク特性の差

エミュレーション実験のスクリーンショットを図 8.24 に示す。

次に実験の結果、100 ノードうち、特に表 8.5 の 5 ペアの通信機会が多かった。

これら 5 ペアは北陸先端科学技術大学院大学近辺 8.25 を移動しているノード群であった。

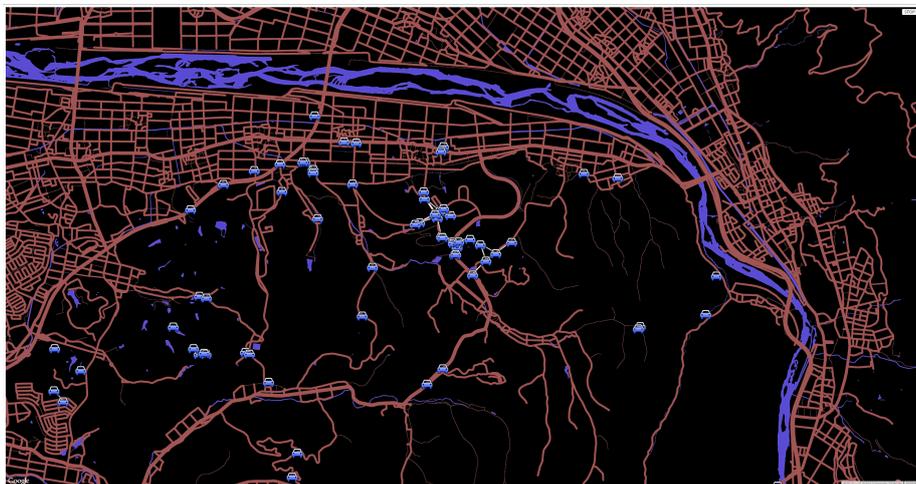


図 8.24: エミュレーションのスクリーンショット

表 8.5: 通信機会の多かった上位 5 ペア

Ranking	Node Pair
1	Node#93 & Node#96
2	Node#19 & Node#96
3	Node#85 & Node#96
4	Node#85 & Node#97

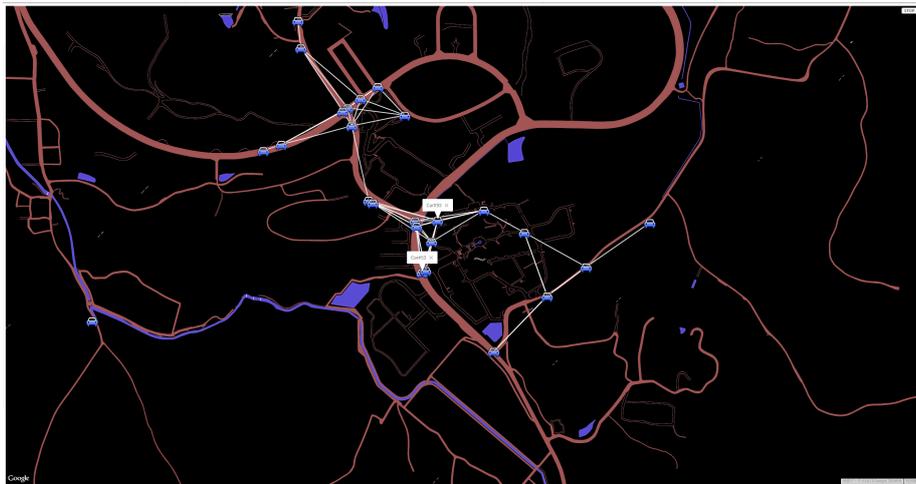


図 8.25: 通信機会が多かった北陸先端科学技術大学院大学付近の拡大図

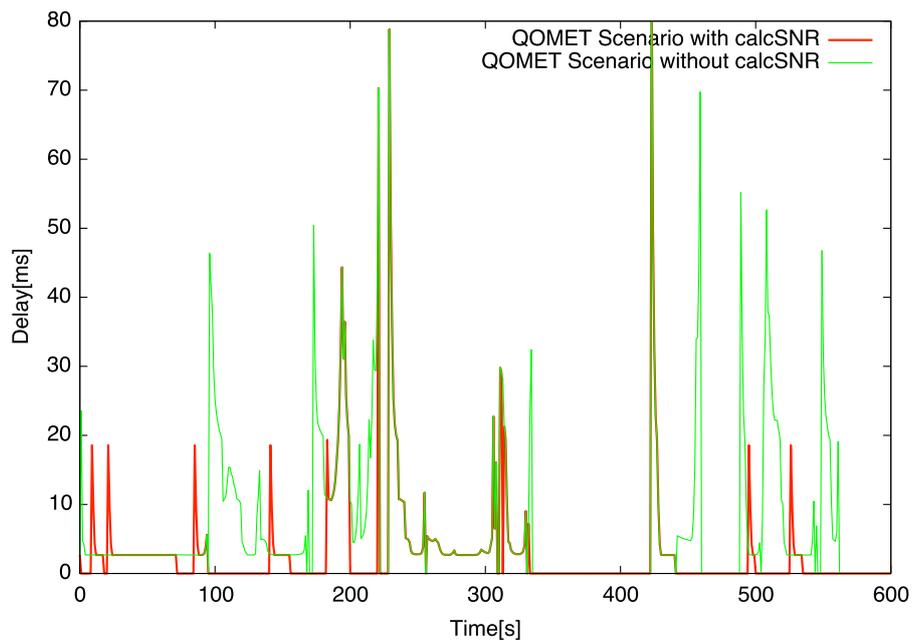


図 8.26: 地形情報の有無による DeltaQ の出力結果比較 (遅延)

これら 5 ペアの通信における DeltaQ の遅延エミュレーション結果を、地形を考慮しなかった場合と、地形を考慮した場合で比較したグラフが図 8.26 である。

地形を考慮した結果、DeltaQ の出力結果においても、部分的に遅延が増加している。この原因は、部分的にノードペア間の見通し直線上に地面の起伏があり障害物となったため

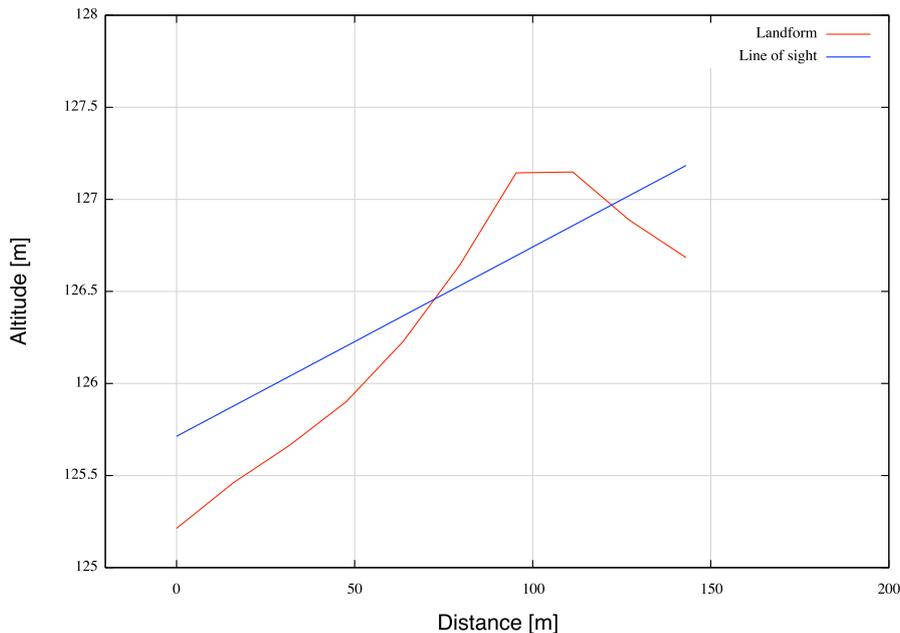


図 8.27: Node53-93 の実験中に見通し直線と地形の交差

青線の送受信アンテナ間に見通し直線に対して、赤線の地形が侵入している事から、本来は回折した電波を受信する環境である事がわかる。

である。図 8.27 は実験中 (図 8.25 のノード位置の時) 最もネットワーク環境変化の多かったペアであった Node#53 と Node#93 の間に見通し直線と、地形の形状を表したグラフである。青の見通し直線に対し、赤の地形が侵入している事がわかる。

図 8.28 は実験中に各ノード間で実際の通信の遅延時間を Ping コマンドを使って計測した結果である。DeltaQ の出力結果に従って、実際に通信を行った際の遅延時間にも影響が出ていることが示されている。

通信量の多かった上位 5 ペアの平均通信可能帯域が地形の影響を考慮しなかった場合と、考慮した場合でどのように変化したかを集計した結果が表 8.6 である。どの結果でも、地形を考慮した方が、平均通信可能帯域が減少していることがわかる。

8.4.2 アプリケーションに特化した環境特性エミュレーションによる効果

これまで、大規模実証検証環境では、汎用的なネットワーク上の遅延やパケットロスのエミュレーション機能の研究が行われきた。しかし、検証するアプリケーションに必要な

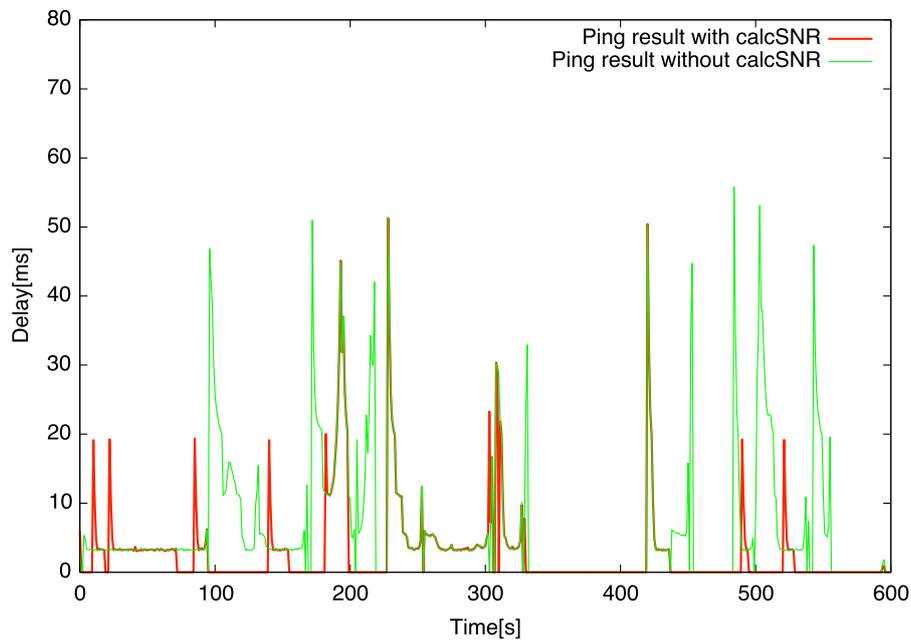


図 8.28: 地形情報の有無による Node53-93 間の Ping による遅延時間結果比較 (遅延)

RSSI が低下し、電波の受信状況が悪化するほど、無線レイヤーでの再送が発生する。その結果、地形の影響を考慮した赤線の方が遅延時間が上昇する傾向がある。逆に、赤線で遅延が $0ms$ で緑の従来手法で遅延が発生している区間は、地形を考慮した結果、通信が不能と計算された区間である。

表 8.6: 通信量上位 5 ペアの地形考慮の有無による平均通信可能帯域

Node Pair	without Elevation(current)	with Elevation(proposal)
Node#93`	5.12Mbps	3.72Mbps
Node#19`	3.80Mbps	2.76Mbps
Node#85`	3.23Mbps	2.10Mbps
Node#85a	2.62Mbps	1.86Mbps

エミュレーションパラメータの設定は、アプリケーション毎に異なり、パラメータの決定手法が重要である事を述べた。そして、論理的検証段階で考慮される環境特性から、これらのエミュレーションパラメータを設定する大規模実証実験環境支援ソフトウェアの提案を行った。

提案の有効性を検証する為に、対象エリアの地形による電波伝播特性を考慮したワイヤレスセンサーネットワークエミュレーションを行い、車車間通信ネットワークの検証を行った。地形を考慮することで、より詳細な通信帯域を再現することが可能である事が示唆された。また、移動モデルの差異が技術検証において重要なパラメータである事も示唆された。論理的検証段階で、的確なノード移動モデルや、地形の特性を考慮したシミュレーションによってネットワークシステムの設計が行われ、それらを再現した大規模実証検証環境での実践的検証を行う事で、設計・実装したネットワークシステムの妥当性の検証が可能となる。

8.5 解析/フィードバックフェーズ

新しく開発されたワイヤレスセンサーネットワークシステム、およびそのアプリケーションは、大規模実証検証環境を用いて検証した後、実環境に投入され運用される。収集したデータはアプリケーションで利用されるが、この時収集したデータの有効性を解析する必要がある。既に展開されているモバイルデバイスを利用したセンサーアプリケーションや、ネットワークにアドホック通信やDTNを利用したセンサーネットワークでは、デバイスの種類、移動特性、密度などの条件により、本来想定していた情報が取得出来ない場合がある。その為、得られたデータが開発したアプリケーションにとって有効なデータであるのか、有効で無い場合原因は何であるのか解析する必要がある。ここで注目すべきは、解析対象が遅延時間やデータ欠損率などのデータ単体での品質だけでなく、論理的検証から実践的検証に至る開発段階で想定していた環境特性と、そこから想定されるデータ品質との乖離である。ワイヤレスネットワークシステム、とりわけアドホックネットワークやDTNを用いた場合では、ノードの移動やデバイス、地形と言った環境特性と、それによって得られるネットワーク特性に最適化したネットワークシステムの設計を行う。その為、これらの環境特性が想定と異なっていた場合、ネットワークシステムの性能が想定通りで無い場合もある。

実環境と想定環境との乖離は、データの品質として観測されるが、原因の特定は非常に

困難である。これは、シミュレーション・エミュレーションと現実環境とが完全には一致しない為現れる誤差と、想定自体の齟齬との分離が困難である事による。しかし、論理的検証段階で示した野生動物を想定したノードの移動モデルと、移動モデルから推定される取得接触データは、実際のセンサーデータからも移動モデルなどが推定可能である事を示唆している。実際、人や動物などの様々な移動モデルは実観測データの統計的な特徴に基づく物で有り、移動モデルの推定に関しては実データを解析することで、想定モデルに沿っているかどうか判断することが可能であると考えられる。この場合、得られた移動モデルの推定自体もネットワークシステムで得られた重要な結果となる。

得られた解析結果は、システムの改善や次のワイヤレスセンサーアプリケーションの提案に重要な情報となる。データの解析により、より詳細な環境特性の推定を行い、システムの改善を行う事も可能となり、また新しいアイデアをもたらし、次のセンサーネットワークアプリケーションの可能性を示唆する場合もある。しかし、これらのデータは論理的検証段階から積極的に環境特性を推定し、システムを最適化した結果得られる結果・差分である事から、本研究で提案しているアプリケーションに特化したセンサーネットワークの設計と検証を開発の過程で行う必要が有る。

第 9 章

結論

本論文では、IoT時代のセンサーネットワークアプリケーション開発の現状を整理し、ワイヤレスセンサーネットワークの設計と検証において課題となる、アプリケーションが観測対象とする環境が持つ環境特性とその影響について述べた（第I部）。さらに、野生動物の長期生態観測ネットワークの設計と車車間通信ネットワークの検証を例に、環境特性がネットワークシステムの開発に与える影響と、環境特性をどのように再現し設計・検証段階で再現するかを議論した（第II部）。また、ネットワークシステムの実装検証段階において、反復的な検証を行う際の課題を整理し、効率的な検証を行う為の実験支援ソフトウェアを提案した（第III部）そして、アプリケーションに特化したワイヤレスセンサーネットワークを迅速に開発する為に、様々な環境特性を論理的検証段階から推定し設計に反映を行い、センサーネットワークシステムの設計・検証を行う、アプリケーション特化型センサーネットワークの設計検証フレームワークを提案した（第IV部）。

9.1 本研究の成果

第I部では、センサーネットワークの発展、IoT時代の到来に伴いワイヤレスセンサーネットワークシステムを利用するアプリケーションが増加すると共に、これまでの設計手法では最適な設計が困難になってきている事を述べた。IoT時代はインフラとして、各種センサー機能を持つ通信デバイスが既に展開されている、もしくは容易に展開できる環境にある。これに対して、従来のセンサーネットワークアプリケーションの設計と検証のアプローチでは最適な設計が困難であり、開発の各工程で、工程間を超える手戻りが発生しやすくなっている現状を述べた。

第Ⅱ部では、ワイヤレスセンサーネットワークシステム設計の為には積極的な環境特性の推定が必要で有る事を述べ、特に初期の設計段階からシミュレーションに基づき、考慮すべき環境特性がある事を明らかにした(3章)。そして、設計段階で考慮すべき特性の検討を、野生動物の長期生態観測ネットワークの設計を例に、実践的検証段階で考慮すべき特性を、災害時情報収集ネットワークの検証を例に議論した。その結果、IoT時代のワイヤレスセンサーネットワークアプリケーションの設計と検証においては、デバイスの特性、ノードの移動、地形による電波伝播の影響を考慮しなければ、最適なシステム設計が行えない事を明らかにした。

第Ⅲ部では、大規模実証検証環境で機能検証や大規模実証検証を行う上で、現在課題となっている環境構築時の障害について述べた(6章)。また、Multicastによるディスクイメージ配布機構を導入することで、大規模実証検証環境の高速な構築を実現した。そして、大規模実証検証環境の段階的、反復的な環境構築の効率を高め、柔軟な大規模実証事件環境の構築と利用を実現するStarBuilderについて提案と実装を行った(7章)。StarBuilderは、ノードのディスクイメージの共有や再利用向上を実現し、また汎用的なグラフ構造描画フォーマットを用いた簡易な実験トポロジの記述を可能とすることで、機能検証フェーズの段階的、反復的環境構築を直感的に行える。

第Ⅳ部では、第Ⅰ-Ⅲ部の成果を踏まえ、IoT時代のワイヤレスセンサーネットワークアプリケーションの開発を支援する為、ワイヤレスセンサーネットワークのアプリケーション特化型センサーネットワークの設計検証フレームワークの提案を行った。このフレームワークでは、ワイヤレスセンサーアプリケーションの開発を4つのフェーズに分割した。そして、それぞれのフェーズで行うべき作業、各フェーズでの設計や検証時に考慮、または段階間で継承すべき環境特性を定義した。加えて、論理的検証フェーズでノードの移動特性や、地形の影響と言ったアプリケーションの提案時には未知である環境特性をシミュレーションする手法について議論と提案を行った。また、論理的検証フェーズで考慮され、実証検証フェーズで継承されるべき、地形と地形による電波伝播特性を大規模実証検証環境でエミュレーションする技術の提案と実装を行った。そして、解析・フィードバックフェーズでは、ノードの移動モデル毎の特徴を、センサが収集するデータの推定値からも抽出可能である事を示し、ネットワーク的極限環境で機能するワイヤレスセンサーネットワークでも、データの有効性が検証可能で有る事を示した。この結果は、新たなワイヤレスセンサーネットワークの設計の指針として有益な指標となると考えられる。

本論文の最初に、情報の収集と共有は人類の根源的な欲求であると述べた。今後人類が

さらなる情報の収集と共有を行う為には、柔軟なワイヤレスセンサーネットワークの構築が不可欠である。これらの議論と提案システムにより、本論文は特定アプリケーション向けのワイヤレスセンサーネットワークを開発・検証する為の手法を明らかにし、さらなるセンサーネットワークの応用範囲の拡大、並びに人類の根源的欲求に対するセンサーネットワークの有用性を示唆した。

9.2 本研究分野の課題と展望

インターネットが急速に普及し、様々なデバイスがネットワークに繋がると共に、デバイスの小型化、低消費電力化、コモディティ化が急速に進んでいる。センサーアプリケーションもそれに伴い急速に応用範囲を広げている。その結果、それまでは全く想定していなかった様なデバイスの組み合わせ、規模、物理環境で動作するセンサーアプリケーションも今後登場してくる事が予想される。本論文で議論したアプリケーションに特化したワイヤレスセンサーネットワークシステムの開発で考慮すべき特性も、それに伴い多様化することが予想される。従って、新しいアプリケーションの提案と共に考慮が必要な環境特性も適宜検討していく必要が有る。

本研究では、その他の環境特性を探る為に1年間にわたり、本学近くの能美市岩本町山林内にバッテリー駆動の無線送受信機を設置し、受信機の電波受信強度 (RSSI : Received Signal Strength Indicator) の変化を計測した (図 9.1 9.2)。計測器は 4.1 同様 Arduino + XBee で作成し (図 9.3)、ソーラーパネルによる発電を行いながらバッテリーを充電するシステムとした。送信機と受信機は約 90m 離れた2本の立ち木に地面の影響を受けない高さで設置した。

図 9.4 は北陸では降雪が予想される1月から2月の2ヶ月間のRSSIと降雨関係を比較したグラフである。森林内で冬期の実験であった為想定よりソーラーパネルの発電量が低く、2週間程度でバッテリー切れとなった。単位時間降雨量の変化とRSSIの変化に相関は見られないが、バッテリーが切れる直前にRSSIが低下する現象が見られる。本研究の当初の目的は降雨・雪による影響の検証であった。既知の電波の知見では10GHz以下の一般的に通信に利用される周波数帯では空間上の降雨の影響は大きくないとされているが、実験の結果からもそれが明らかになった。しかし、長期にわたる実験の結果は降雨以外にバッテリーの影響が大きい可能性が明らかとなった。Arduino等の組み込み向けのシンプルな機器は入力電圧の変動がXBee等への出力電圧の変化として現れやすい。これは、位置特



図 9.1: 岩本町内山林での機器設置風景



図 9.2: 送受信機設置状況

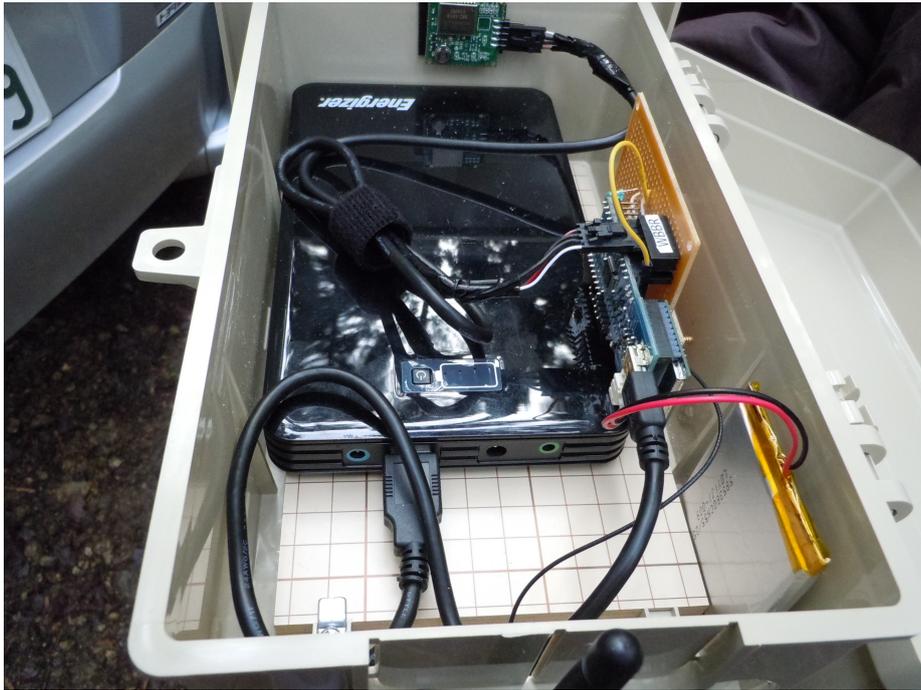


図 9.3: 製作した送受信機

定用ビーコンユニットなどでも同様であるため、バッテリーの電圧変化が電波の伝播距離に影響を及ぼす可能性が有る。また、発電量と消費電力のバランスも事前に検証が必要である。その為、バッテリーで駆動するデバイスを用いたネットワークシステムを詳細に検証する場合には、バッテリー消費をエミュレーションする必要がある事が示唆された。

本論文で議論したこれら未知の環境特性を積極的に推定した結果が、実環境とどれほど近似するのか、シミュレーションから得られた結果やエミュレーションの結果がどの程度現実の事象やシステムの挙動を模倣しているのか、これらを単純に判断することは難しい。岩本町での実験の結果の様に、予想外の機器特性、環境特性がシステムに影響を与える事もある。どれほど詳細に検討を行い特性を推定しても、実環境においては様々な想定外の外乱や障害が発生する。しかし、観測したい環境や取得したい情報に対して、センサーネットワーク並びにそれを支える実証検証環境基盤が積極的にシステムの設計と有用性を提示し、貢献していくことは我々の責務であると考える。

今後様々な環境を観測するセンサーネットワークシステムの設計と検証の支援の為に、本論文では網羅しきれなかった内乱・外乱を考慮しシステムの設計・検証を行う技術の開発を行いより高度なアプリケーション特化センサーネットワークの設計検証フレームワー

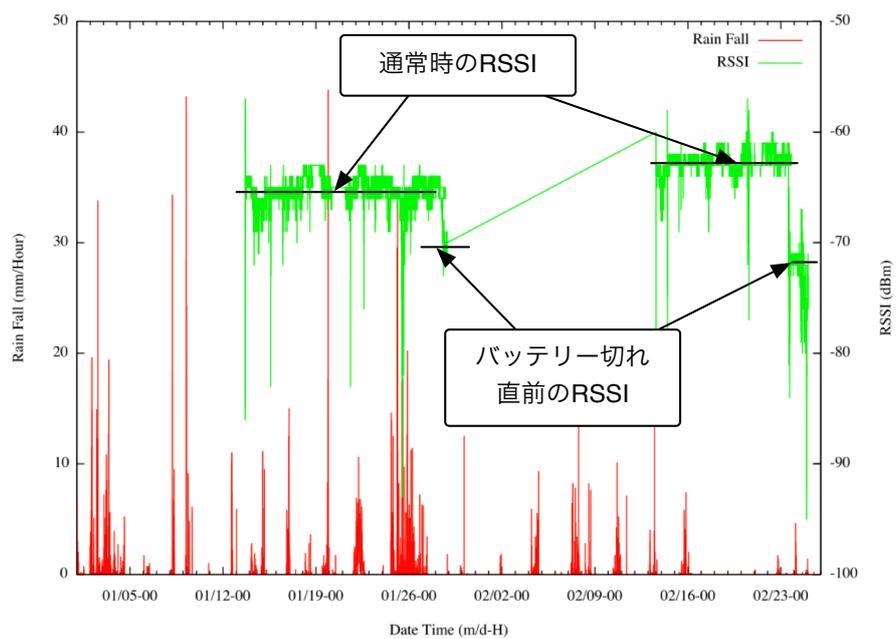


図 9.4: 2ヶ月間の RSSI と単位時間降雨量

赤の線が単位時間降雨量で、この年は降雪量が少なく計測した期間は積雪する程の降雪は無かった。RSSI の 2 月初旬に見られる右上がりの直線は、バッテリー切れでデバイスが駆動していなかった期間である。バッテリー切れ直前と、バッテリー交換直後では RSSI に $10dBm$ 程度の信号強度差があることがわかる。

ク並びに、それを支えるシミュレーション、エミュレーション手法を研究・開発していく事で、今後のセンサーネットワークの発展に貢献する事が期待できる。

謝辞

本研究を行い、論文を執筆するにあたり、終始御指導賜りました本学、篠田教授に深く感謝いたします。また、本学、丹康雄教授、知念賢一准教授、情報通信研究機構 三輪信介 北陸 StarBED 技術センター長ならびに富山県立大学 岡田敏美教授には、審査委員をお引き受けいたけき、多くの助言を頂けたことを深く感謝致します。この場を借りて、深く感謝いたします。

次に、同じ研究室の仲間として、日々の議論を行った篠田研究室の修了生、現メンバーにも感謝します。特に、同じ博士後期課程の学生であった、高野祐輝博士と井上朋哉博士には、研究と学生生活の両面で大きな支援をもらった事を感謝します。また、先輩、同期、後輩の、Latt Khin Tida 博士、Nguyen Lan Tien 博士、Nguyen Nam Hoai 博士、Saber Zrelli 博士、Muhammad Imran Tariq 氏、芳炭将氏、我如古津世史氏、出口真人氏、木下稔氏、墨岡 冲氏、高島大祐氏、内田幸治氏、磯崎直樹氏、阪上武寛氏、上野隆資氏、梅木孝志氏、野中雄太氏、松井大輔氏、千装俊幸氏、佐川喜昭氏、明石邦夫氏、川瀬拓哉氏、栗原良尚氏、立花一樹氏、中村祐輔氏、橋本将彦氏、山田悠介氏、吉岡慎一郎氏、鍛治祐希氏、大野夏希氏、田部英樹氏、向井雄一郎氏、村上正太郎氏らと共に学生生活を歩めたことを感謝したい。

次に、FABEE/WAOC Project の方々に深く感謝を致します。特に代表である東京大学 空間情報科学研究センター 小林博樹助教、並びにメンバーである本学大学院教育イニシアティブセンター 崔瞬星特任助教、福島大学理工学群共生システム理工学類 永幡幸司准教授には、本研究の骨格となるテーマを構築する上で多くの有益な助言と助けを頂きました。

WIDE Project の方々にも深く感謝いたします。本研究を進める上で、様々な助言を頂くと共に、同じネットワークを研究する仲間として大きな刺激を頂きました。また、本学、宇多仁助教、ならびに NTT コミュニケーションズ株式会社 小原泰弘氏にも、研究生活を行う上で色々と助言を頂き非常に感謝しています。

次に、現在働いている情報通信研究機構の方々にも多くの助言と、研究の支援を頂いた事を感謝します。特に、同じ研究室で多忙な時期にも業務・研究の両立を支えてくれた、宮

地利幸主任研究員、三浦良介技術員、中井浩技術員に感謝します。日々の研究のサポートを頂いた二口美穂氏、下口宗裕氏、廣澤裕子氏に感謝いたします。

次に、約1年間に及ぶ屋外でのRSSI計測実験に際し、能美市岩本町の森林地の利用についてご相談、ご協力頂いた有限会社 岩本緑化建設 代表取締役 田嶋建夫氏、並びに私有地の利用を快諾して頂いた新宅寿氏に深く感謝致します。

また、株式会社 Clwit の井沢志充博士、清原智和氏に感謝します。日々の仕事を御一緒させて頂けたことは大変刺激になりました。

以上、全ての方の温かいご支援がなければ本論文を完成させることは不可能でした。心から感謝いたします。そして、本学博士課程への進学を後押しして頂く共に、研究者の心得をご教授頂いた情報通信医学研究所 代表理事 中川晋一博士に深く感謝いたします。

最後に、研究生活を送る上で、家族そして妻の絵里には本当に迷惑をかけました、長い学徒の期間、経済的にも、精神的にも支え続けてくれた事に心から感謝を表し謝辞と致します。

本研究に関する発表論文

- 安田 真悟, 小林 博樹, 崔 舜星, 篠田 陽一, “動物指向クラウドネットワークの設計と課題”, 第13回インターネットテクノロジーワークショップ, to be appeared, 2012.
- 安田 真悟, ラズバン ベウラン, 崔 舜星, 三輪 信介, 篠田陽一, “野生動物行動によるDTNの技術要件に関する検討”, インターネットコンファレンス2013, to be appeared, 2013.
- Shingo Yasuda, “WAOC: Examination of long-term observation technique for wildlife in the forest area that devastated by earthquake”, #14 APNGCAMP, Seoul Korea, 15-18 Aug 2012.
- 安田 真悟, 篠田 陽一, 敷田 幹文, “ネットワークアプリケーション実証環境構築フレームワークの実装と評価”, 電子情報通信学会技術研究報告 110(341), 25-30, 2010-12-16.
- Shingo Yasuda, Kunio Akashi, Tomoya Inoue, Toshiyuki Miyachi, Shinsuke Miwa, Ken-ichi Chinen, Yoichi Shinoda, “Requirements of Large Data Distribution Mechanism for Large-Scale Network Testbed”, Proceedings of the 2nd European Conference of Computer Science (ECCS '11), ISBN:978-1-61804-056-5, pp.315-322, Tenerife, Spain, Dec, 2011.
- Shingo Yasuda, Kunio Akashi, Masatoshi Enomoto, Shinsuke Miwa, Yoichi Shinoda, “Technical Requirements of Experiment Support Software for Huge-Scale Network Environment”, International Journal of Computers and Communications, Issue 4 Volume 7, 2013.
- 安田 真悟, 小林 博樹, 崔 舜星, 工藤宏美, 篠田陽一, “動物指向クラウドネットワークの設計と課題”, CSISDAYS2012 研究アブストラクト集, pp.20 2012-11.

- 安田 真悟, 宮地 利幸, ラズバン ベウラン, 牧野 義樹, 明石 邦夫, 井上 朋哉, 三輪 信介, 宇多 仁, 丹 康雄, 篠田 陽一, “空間情報に基づく災害時シナリオを用いたネットワークエミュレーション基盤の構築”, CSISDAYS2012 研究アブストラクト集, pp.61-2012-11.
- Shingo Yasuda, Kunio Akashi, Toshiyuki Miyachi, Razvan Beuran, Yoshiki Makino, Tomoya Inoue, Shinsuke Miwa and Yoichi Shinoda, “Emulation-based ICT System Resiliency Verification for Disaster Situations”, Workshop on Resilient Internet based Systems (REIS 2013), to be appeared, SITIS 2013, Kyoto, Japan, 2-5 December 2013.
- Toshiyuki Miyachi, Razvan Beuran, Yoshiki Makino, Kunio Akashi, Shingo Yasuda, Tomoya Inoue, Shinsuke Miwa, Satoshi Uda, Yasuo Tan, Yoichi Shinoda, “On network system evaluation under various failures”, VALUETOOLS 2012: 226-227.
- 小林博樹, 崔舜星, 安田 真悟, 松浦知史, “計算機を介した人と生態系のインタラクションの研究～イリオモテヤマネコの保全現場支援から被爆生態系の再生過程解明まで～”, インターネットアーキテクチャ研究会, 招待講演, 2012-2.
- 井上 朋哉, 安田 真悟, 高野 祐輝, 宇多 仁, 篠田 陽一, “PacketX: エンドホストにおけるアプリケーション指向 IP パケット制御機構”, インターネットコンファレンス 2008 論文集, 日本ソフトウェア科学会, ISSN 1341-870X, pp.99-107, 2008.10.
- 上野 隆資, 安田 真悟, 井上 朋哉, 篠田 陽一, “アプリケーションベース MapTB マーカの提案”, 日本ソフトウェア科学会 第 8 回 インターネットテクノロジーワークショップ, 2007.6.
- 鍛冶 祐希, 安田 真悟, 知念 賢一, 篠田 陽一, “SpringOS M-extension: 超大規模実証実験環境を想定したリソース指向実験ノード管理手法”, Dicom 2012, pp.1513-1519, 2012.

参考文献

- [1] ソラテナ. <http://weathernews.jp/door/soratena/>.
- [2] ドコモ環境ライブ. <http://docomo-live.jp>.
- [3] 坂村健. TRON からの発想. 岩波書店, 1987.
- [4] Mark Weiser. The computer for the 21st century. *Scientific American*, Vol. 265, No. 3, pp. 66–75, January 1991.
- [5] StarBED Project. <http://www.starbed.org>.
- [6] SMART DUST Project.
<http://robotics.eecs.berkeley.edu/~pister/SmartDust/>.
- [7] V. Cerf, S. Burleigh, A. Hooke, L. Torgerson, R. Durst, K. Scott, K. Fall, and H. Weiss. Delay-Tolerant Networking Architecture, RFC4838. April 2007.
- [8] K. Scott and S. Burleigh. Bundle Protocol Specification, RFC5050. November 2007.
- [9] 宮地利幸. 大規模実証環境の実現と実験支援によるネットワークサービスの検証技術. PhD thesis, 北陸先端科学技術大学院大学, 2007.
- [10] Toshiyuki Miyachi, Takeshi Nakagawa, Ken ichi Chinen, Shinsuke Miwa, and Yoichi Shinoda. Starbed and springos architectures and their performance. In *TRIDENT-COM*, Vol. 90, pp. 43–58, 2011.
- [11] R. Beuran, L. T. Nguyen, and Y. Shinoda. QOMB wireless network emulation testbed: Evaluation and case study. In *5th ACM International Workshop on Wireless Network Testbeds, Experimental Evaluation and Characterization (WiNTECH 2010)*, in conjunction with *MobiCom 2010, Chicago, Illinois, USA*, pp. 1–8, Sep 2010.

- [12] R. Beuran, J. Nakata, Y. Tan, and Y. Shinoda. Ieee 802.15.4 network emulation testbed. In *25th IEEE International Conference on Advanced Information Networking and Applications (AINA 2011), Biopolis, Singapore*, pp. 451–458, Mar 2011.
- [13] ONE (Opportunistic Network Environment).
<http://www.netlab.tkk.fi/tutkimus/dtn/theone/>.
- [14] OpenStreetMap. <http://www.openstreetmap.org>.
- [15] FABEE Project. <http://fabee.jp>.
- [16] 板生清. ウェアラブル環境情報ネット技術とその応用. FED ジャーナル Vol.11 No.3, pp. 25–33, 2000.
- [17] 川口敏. 死物学の観察ノート 身近な哺乳類のプロファイリング. PHP 研究所, 2001.
- [18] Toshiyuki Miyachi, Razvan Beuran, Yoshiki Makino, Kunio Akashi, Shingo Yasuda, Tomoya Inoue, Shinsuke Miwa, Satoshi Uda, Yasuo Tan, and Yoichi Shinoda. On network system evaluation under various failures. In *VALUETOOLS*, pp. 226–227, 2012.
- [19] Shingo Yasuda, Kunio Akashi, Toshiyuki Miyachi, Razvan Beuran, Yoshiki Makino, Tomoya Inoue, Shinsuke Miwa, and Yoichi Shinoda. Emulation-based ict system resiliency verification for disaster situations. In *9th International Conference on Signal Image Technology & Internet based Systems (SITIS 2013), Workshop on Resilient Internet based Systems (REIS 2013) SITIS*, pp. 875–882, Dec 2013.
- [20] K. Kanchanasut, A. Tunpan, A. M. Awal, K. D. Das, T. Wongsardsakul, and Y. Tsuchimoto. Dumbonet: a multimedia communication system for collaborative emergency response operations in disaster-affected areas. *International Journal of Emergency Management*, Vol. 4, No. 4, pp. 670–681, 2007.
- [21] S. Schildt, J. Morgenroth, W.-B. Pottner, and L. Wolf. Ibr-dtn: A lightweight, modular and highly portable bundle protocol implementation. *Electronic Communications of the EASST*, Vol. 37, pp. 1–11, January 2011.

- [22] 奥村義久, 大森英二, 河野十三彦, 福田倚治. 陸上移動無線における伝搬特性の実験的研究. *通研実報*, Vol. 16, No. 9, pp. 1705–1764, 1967.
- [23] M. Hata. Empirical formula for propagation loss in land mobile radio services. *IEEE Trans. on Vehicular Technology*, Vol. 29, pp. 317–325, 1980.
- [24] Shuttle Radar Topography Mission. <http://www2.jpl.nasa.gov/srtm/>.
- [25] A. G. Longley and P. L. Rice. Prediction of tropospheric radio transmission loss over irregular terrain. *ESSA Tech. Report ERL65-ITS67. Available from NTIS*, pp. 676–874, 1968.
- [26] 原岡充. 中山間地域のための無線によるボランティア非常通信網整備を目的とした電波伝搬シミュレーションの試みについて. 東京大学 空間情報科学研究センター シンポジウム, 2008.
- [27] Razvan Beuran, Lan Tien Nguyen, Khin Thida Latt, Junya Nakata, and Yoichi Shinoda. Qomet: A versatile wlan emulator. In *IEEE International Conference on Advanced Information Networking and Applications (AINA-07), Niagara Falls*, pp. 21–23, 2007.
- [28] Daniel B. Faria. Modeling signal attenuation in ieee 802.11 wireless lans - vol. 1. Technical Report TR-KP06-0118, Kiwi Project, Stanford University, jan 2006.
- [29] 藤原明広, 巳波弘佳. Bluetooth&wi-fi モバイル無線通信実験とすれ違い頻度分布の乗則: 一期一会の影響 (nw 制御 (1)). 電子情報通信学会技術研究報告. IN, 情報ネットワーク, Vol. 110, No. 449, pp. 139–144, feb 2011.
- [30] 宮地利幸, 中田潤也, 知念賢一, ラズバン・ベウラン, 三輪信介, 岡田崇, 三角真, 宇多仁, 芳炭将, 丹康雄, 中川晋一, 篠田陽一. Starbed 大規模ネットワーク実験環境. Vol. 49, No. 1, pp. 1–14, 2008.
- [31] 寺田真敏, 重本倫宏, 仲小路博史. P2p ファイル交換ソフトウェア winny を対象としたオーバーレイネットワークの制御実験. Vol. 2009-CSEC-45, No. 22, May 2009.
- [32] Shinsuke Miwa, Mio Suzuki, Hiroaki Hazeyama, Satoshi Uda, Toshiyuki Miyachi, Youki Kadobayashi, and Yoichi Shinoda. Experiences in emulating 10k as topology

- with massive vm multiplexing. In *The First ACM SIGCOMM Workshop on Virtualized Infrastructure Systems and Architectures (VISA '09)*, Aug 2009.
- [33] iLO(Integrated Lights-Out 2).
<http://h18004.www1.hp.com/products/servers/management/remotemgmt.html>.
- [34] 宮地利幸, 知念賢一, 篠田陽一. Starbed および springos の性能評価. 日本ソフトウェア科学会 研究会資料シリーズ No.36, pp. 81–88, 2007.
- [35] Sally Floyd, Van Jacobson, Ching-Gung Liu, Steven McCanne, and Lixia Zhang. A reliable multicast framework for light-weight sessions and application level framing. *IEEE/ACM Trans. Netw.*, Vol. 5, No. 6, pp. 784–803, December 1997.
- [36] Miwa Shinsuke, Miyachi Toshiyuki, Eto Masashi, Yoshizumi Masashi, and Shinoda Yoichi. Design issues of an isolated sandbox used to analyze malwares. In *IWSEC*, Vol. 4752 of *Lecture Notes in Computer Science*, pp. 13–27. Springer, 2007.
- [37] Noriyuki Suzuki, Jane Louie Fresco Zamora, Shigeru Kashiwara, and Suguru Yamaguchi. Using sos message propagation to estimate the location of immobilized persons. In *Proceedings of the 18th Annual International Conference on Mobile Computing and Networking*, Mobicom '12, pp. 455–458, New York, NY, USA, 2012. ACM.
- [38] 株式会社ゼンリンデータコム : 混雑統計[®].
<http://www.zenrin-datacom.net/business/other/>.
- [39] 株式会社NTTドコモ : モバイル空間統計に関する情報.
https://www.nttdocomo.co.jp/corporate/disclosure/mobile_spatial_statistics/.
- [40] NS-2. http://http://nslam.isi.edu/nslam/index.php/User_Information.
- [41] QualNet. <http://network.kke.co.jp/products/qualnet/>.
- [42] 前田久美子, 佐藤和基, 小西一樹, 内山彰, 廣森聡仁, 山口弘純, 安本慶一, 東野輝夫. 携帯端末の現実的な移動モデルを表現可能なネットワークシミュレータの設計と実装 (マルチメディアアプリケーション). 情報処理学会研究報告. マルチメディア通信と分散処理研究会報告, Vol. 2004, No. 89, pp. 21–27, sep 2004.

- [43] MobiREAL. http://www.mobireal.net/index_j.html.
- [44] I. Rhee, M. Shin, S. Hong, K. Lee, and S. Chong. On the levy-walk nature of human mobility. In *Proc. INFOCOM 2008. The 27th Conference on Computer Communications. IEEE*, pp. 924–932, 2008.
- [45] A. Fujihara and H. Miwa. Homesick levy walk and optimal forwarding criterion of utility-based routing under sequential encounters. *Internet of things and inter-cooperative computational technologies for collective intelligence, Studies in Computational Intelligence*, Vol. 460, pp. 207–231, 2013.
- [46] 池田透, 遠藤将史, 村野紀雄. 野幌森林公園地域におけるアライグマの行動圏. 酪農学園大学紀要. 自然科学編, Vol. 25, No. 2, pp. 311–319, feb 2001.
- [47] METEOR. https://github.com/k-akashi/qomet_linux.
- [48] googleMapsAPI. <https://developers.google.com/maps/?hl=ja>.