卒業論文 2011年度 (平成23年度)

被災地における WiFi-based Long Distance Network の構築 ネットワーク設置位置推薦システムの設計と実装

> 慶應義塾大学 環境情報学部 村上 滋希

被災地における WiFi-based Long Distance Network の構築 ネットワーク設置位置推薦システムの設計と実装

2011 年 3 月 11 日に発生した東日本大震災において,電子メール, Web サービス, Skype を初めとするインターネットを介したサービスが活用され家族の安否確認や被害状況,交通情報などの様々な情報のやり取りに大きな役割を果たした.インターネットを介することで,情報の双方向性が成立し,地方自治体や政府機関だけでなく一般市民のレベルで被害情報を集約/発信/拡散された.

被災地でよりインターネットを活用するためには、インフラの寸断や被災地特有のネットワークの需要を考慮すると、被災者自らの手でネットワークを構築する必要がある。本研究は WiFi-based Long Distance Network(以下 WiLD Net と呼ぶ) に着目し被災地において、建物の屋上などに機材を設置し、メッシュネットワークを構築を考える。 WiLD Net を構築する際の障壁として、機材の設置場所問題があげられる。 見通しや建物への立ち入り、電源の調達、ネットワークの引き込みなど、設置する場所の制約に加え、干渉の低減や伝送距離の短縮、マルチホップネットワークにおけるホップ数の低減を考慮したネットワークの設計の双方を同時に考慮する必要がある。

本研究では、これらの問題点を整理した上で、被災地における WiLD Net 構築を支援するシステムの設計と実装をする。事前に入手した地図情報をデータベース化し、見通しの計算、ネットワークの設計を同時に行い、機材の設置場所の推薦モジュールや被災状況によって変化する状況を手動で更新しデータベースに反映する、データベース更新モジュールを初めとする5つのモジュールとデータベースが連携することで、被災地において高スループットの WiLD Net の迅速な構築を可能にする.

本システムにおける計算上の見通しと実際の写真から目視で確認できる見通しを比較した見通しの正答率は建物データを入力により更新した状態で、52.9%(写真から建物を特定出来ないものは 36.0% でこれを含む) であり、最新の地図データを用いれば、コンピュータ上で見通しの計算が出来る。システムの計算時間は、南北約 1.7km、東西約 1.8km、面積 $3.06km^2$ 、データベースのレコード数 774,877 のデータを用い、設置数 20 箇所計算した場合約 71 秒設置数 100 箇所で計算した場合

約346秒であり、この時間において、ユーザーの入力時間を加えても、見通しの確認や場所の制約、ネットワークの設計を考慮し、設置位置を決定することができるため、現地調査やネットワークの設計にかかる時間を削減やネットワーク全体のスループットの低下を防ぐことが期待できる。本研究は被災地におけるWiLD Net を活用した迅速なネットワーク構築に貢献が期待できる。

キーワード

1. WiLD Net, 2. Long Distance Wifi, 3. Wireless Mesh Network, 4. 被災地, 5. 減災

慶應義塾大学 環境情報学部 村上 滋希

WiFi-based Long Distance Inter-Rooftop Network in Disaster Area: Design and Implementation of Location Recommendation System to Deploy Network

On March 11th, 2011, Japan was devastated by the massive Tohoku Earthquake and Tsunami. While many people tried to gather information about the latest situation, or the whereabouts about their loved ones during the chaos, there is no question about the great contributions of Internet-based services (e.g. web, email, and Skype) upon connecting people. Through the Internet, bidirectional communication channels were established, and enabled not only the local government, but also the general public to disseminate and exchange necessary information with each other.

However, it is likely that communication infrastructures would be damaged when disasters occur. Therefore, enabling Internet connectivity at the disaster site requires the establishment of temporary communication infrastructures by the people at the actual site. This research would focus on assistive methods for establishing temporary mesh networks using WiFi-based long distance networking technologies.

The main obstacle upon establishing WiFi-based Long Distance Networks (WiLD Net) would be the decision making of the peers' location. In order to make the WiLD Net operational, the visual clearance, distance, number of hops between peers must be taken into consideration. In this thesis, a map-based WiLD Net establishment support system, which enables the rapid deployment of high throughput WiLD Net, is proposed and evaluated. This system would calculate the visual clearance, decide the location of the peers, and automatically designs networks at the users' demands.

As a result of evaluations, the system succeeded in calculating peer locations within 71 seconds, for a network with 20 peers in a experimental environment which consists of 774,877 peer-candidate-locations, extending 1.7km south to north and 1.8km east to west. Consid-

ering the amount of days it would require to manually survey the peer-candidate-locations, this system has the potential to enables extremely rapid design of the WiLD Net in disaster scenarios.

${\bf Keywords}:$

1. WiLD Net, 2. Long Distance Wifi, 3. Wireless Mesh Network 4. Disaster Area,

5. Disaster Prevention

Keio University Faculuty of Environment and Information

Shigeki Murakami

目次

第1章	はじめに	1
1.1	研究の背景	1
1.2	目的	2
1.3	構成	2
第 2 章	WiLD Net の特性と構築における問題点	3
2.1	WiLD Net の運用実験	3
	2.1.1 手法	3
	2.1.2 結果	5
	2.1.3 考察	7
2.2	RSSI とネットワークパフォーマンスの関係性測定実験	10
	2.2.1 手法	10
	2.2.2 結果	12
	2.2.3 考察	14
2.3	無線ネットワークの特性	14
	2.3.1 スループット低下の要因	15
2.4	WiLD Net 構築における考察	15
2.5	まとめ	16
第3章	関連研究/事例/製品	17
3.1	被災地における WiLD Net 構築事例	17
	3.1.1 Lifeline Station/PDRNET	17
	3.1.2 Inveneo/Nethope の取り組み	17
3.2	WiLD Net の構築支援	18

	3.2.1 ネットワークトポロジーの自動生成	19
	3.2.2 電波伝搬シミュレーション	20
3.3	考察	21
3.4	まとめ	21
第4章	アプローチ	23
4.1	WiLD Net 構築の課題の整理	23
4.2	機能要件	24
4.3	アプローチ	24
4.4	まとめ	24
第5章	設計	26
5.1	システム全体の設計	26
5.2	各モジュールの詳細	27
5.3	各モジュールの連携	28
5.4	まとめ	28
第6章	· · · · · · · · · · · · · · · · · · ·	30
6.1	各モジュールの実装....................................	30
	各モジュールの実装	30 30
	6.1.1 データベース	30
	6.1.1 データベース	30 30
	6.1.1 データベース	30 30 31
	6.1.1 データベース	30 30 31 32 33
	6.1.1 データベース 6.1.2 地図データ取得モジュール 6.1.3 建物検出モジュール 6.1.4 ユーザーインターフェース 6.1.5 データベース更新モジュール	30 30 31 32 33
6.1	6.1.1 データベース 6.1.2 地図データ取得モジュール 6.1.3 建物検出モジュール 6.1.4 ユーザーインターフェース 6.1.5 データベース更新モジュール 6.1.6 位置情報推薦モジュール	30 30 31 32 33 33
6.1	6.1.1 データベース. 6.1.2 地図データ取得モジュール. 6.1.3 建物検出モジュール. 6.1.4 ユーザーインターフェース. 6.1.5 データベース更新モジュール. 6.1.6 位置情報推薦モジュール. 動作概要.	30 30 31 32 33 33 36
6.1 6.2 6.3	6.1.1 データベース. 6.1.2 地図データ取得モジュール. 6.1.3 建物検出モジュール. 6.1.4 ユーザーインターフェース. 6.1.5 データベース更新モジュール. 6.1.6 位置情報推薦モジュール. 動作概要 実装環境.	30 30 31 32 33 33 36 36
6.1 6.2 6.3 6.4	6.1.1 データベース 6.1.2 地図データ取得モジュール 6.1.3 建物検出モジュール 6.1.4 ユーザーインターフェース 6.1.5 データベース更新モジュール 6.1.6 位置情報推薦モジュール 動作概要 実装環境 まとめ	30 30 31 32 33 36 36 37

	7.2.1 見通し計算の評価	. 39
	7.2.2 計算量/計算時間/メモリの消費量の評価	. 42
7.3	まとめ	. 46
第8章	おわりに	47
8.1	本研究のまとめ	. 47
8.2	今後の課題と展望	. 48
	8.2.1 機能の追加	. 48
	8.2.2 防災/減災への応用	. 48
参考文献	武	49
謝辞		51
付録 A	建物 ID 付与のソースコード	53
付録 R	見诵し確認のソースコード	56

図目次

2.1	ネットワークトポロジー	4
2.2	会場と栗駒総合支所の位置関係とネットワークの敷設経路	5
2.3	栗駒総合支所のネットワーク	6
2.4	産業まつり会場のネットワーク	7
2.5	パケットの遅延計測結果	8
2.6	1分毎のパケットロス計測結果	9
2.7	Long Distance Wifi キット	9
2.8	計測用ネットワークの構成	11
2.9	計測用ネットワークの様子	12
2.10	RSSI ごとのスループット	13
2.11	RSSI 毎のパケットロス率の関係 (1)	13
2.12	RSSI 毎のパケットロス率の関係 (2)	14
2.13	スループット低下の要因の整理	16
3.1	Life Line Station の概要図	18
3.2	ハイチ地震時に構成されたネットワーク (1)	19
3.3	ハイチ地震時に構成されたネットワーク (2)	20
5.1	システム設計図	29
6.1	地図情報取得	31
6.2	建物検出アルゴリズム・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・	32
6.3	ユーザーインターフェースの外観	33
6.4	見通し計算	34

6.5	動作概要	36
7.1	見通し評価データベース上にある全ての建物のマッピング	40
7.2	屋上からの見通し	41
7.3	レコード数 487,627 のプログラムの実行時間	44
7.4	レコード数 774,877 のプログラムの実行時間	44
7.5	レコード数 487,627 の場合のメモリの消費量	45
7.6	レコード数 774,877 の場合のメモリの消費量	45

表目次

2.1	利用機材	4
2.2	計測帯域	8
2.3	利用機材	11
2.4	減衰量と Access Point の受信電力の関係	12
3.1	解決手法の比較 (設置場所の制約)	21
3.2	解決手法の比較 (ネットワークの考慮)	22
6.1	地図情報を格納するテーブルの設計	30
6.2	テーブルの設計	35
6.3	実装環境	37
7.1	写真から判断できる見通し可能な建物の数	39
7.2	データベース更新後の写真から判断できる見通し可能な建物の数	42
7.3	利用データ	43

第1章 はじめに

1.1 研究の背景

2011 年 3 月 11 日に発生した東日本大震災において、電子メール、Web サービス、Skype を初めとするインターネットを介したサービスが、家族の安否確認や被害状況、交通情報などのやり取りに大きな役割を果たした。インターネットを介することで情報の双方向性が成立し、地方自治体や政府機関だけでなく一般市民のレベルで被害情報を集約/発信/拡散された。

被災地域でよりインターネットを活用するためには、被災地域の人々自らの手で断絶したネットワークの復旧やネットワークの需要がある場所 (例えば、体育館などの避難者への情報提供/屋外に設置された災害対策窓口での業務/2 次災害が発生しそうな場所のモニタリングへの利用など) への設置をすることが考えられる。ネットワークが構築されることで、情報伝達による救援の効率化、被害状況の収集、被災地からの情報発信、河川や土砂崩れなどのモニタリングが可能となり、早期救援や2次災害の防止、行政機能の効率化が期待される。

被災地域において, ネットワークを構築する研究として, Life Line Station[1], Cisco 社の Cisco IP VSAT Satellite WAN Network Module [2] を始めとし, 衛星通信, MANET, DTN などの分野が研究されている.

本研究では、WiFi-based Long Distance Network (以下 WiLD Net と呼ぶ) に着目する。WiLD Net は安価に構築でき、免許が必要なく、長距離伝送が可能であり、マルチホップさせることで広範囲にネットワークを構築できるという特性がある。被災時においても有効に活用でき、ハイチ地震での inveneo[3] の取り組みや東日本大震災での PDRNET[4] において活用された。

これらのプロジェクト報告より WiLD Net を仮設的に被災地で構築する場合, 第 2 章で述べるように, 被災地域で建物の形状や倒壊状況による設置の可否や見通し, 電源の有無, ネットワークの引き込みが可能かなど場所の制約があり, 無線の基地局の設置場所を決定することが難しい. さらに,

高スループットの WiLD Net を構築するためには、リンク間の実空間上の距離、ホップ数、無線のチャネル重複による干渉を考慮したネットワークの設計が必要である.

要するに、被災地で WiLD Net の構築にあたって、場所の制約とネットワークの設計の双方を考慮する必要があり、被災地で WiLD Net を構築する上で大きな障壁となる.

1.2 目的

本研究の目的は、被災地において WiLD Net を構築する際の課題を整理し、整理された要件に基づき、WiLD Net 構築のための機材の設置位置の推薦をするシステムの設計と実装をすることで、被災地域において高スループット WiLD Net の迅速な構築を支援するツールの開発することである.

1.3 構成

本論文は全8章で構成される。第2章では、WiLD Net の基本的な特性について明らかにし、問題点の整理を行う。第3章では、関連研究/事例/既存の製品をまとめそれぞれについて特性を比較する。第4章では、第2章、第3章を踏まえ、本研究のアプローチを示す。第5章では、データベースの構築から位置推定までシステム全体の設計を述べる。第6章では、第5章で述べた設計に基づいた実装の詳細を述べる。第7章では、本提案手法を評価する。第8章では、結論と今後の課題について述べる。

第2章

WiLD Net **の特性と構築における問** 題点

本章では、第 1 章で述べた WiLD Net の設置/運用の難しさを明らかにする。WiLD Net の構築 手順や運用に際するネットワークの計測結果について、第 2.1 節 WiLD Net を構築/運用実験でまとめる。第 2.1 節を受けて、設置場所の制約とネットワークの設計について、RSSI とスループットの関連性の計測追実験(第 2.2 節)や既存の WiLD net や Wireless Mesh Network の研究事例(第 2.3 節)を通じてまとめる。

2.1 WiLD Net **の運用実験**

本実験では WiLD Net の構築/運用をし WiLD Net の構築手順や構築における問題点の確認, 実環境での伝送におけるスループットやパケットロス率,遅延などネットワークの特性を明らかに する.

2.1.1 手法

本実験は 2011 年 10 月 29 日, 30 日の栗原市産業まつり会場においてまつり会場とワンセグ送出機の間のネットワーク (距離にして 350m) を, WiLD Net と UTP ケーブルの配線により, ネットワークを構築し行った. ネットワークは, アクセスポイントに八木アンテナを接続し, WDS モードで無線ブリッジとして利用し, 祭り会場とワンセグコンテンツ送出機の間を結んだ. 詳細なネットワーク設計として, 構築したネットワークトポロジーを図 2.1 に, 設置した産業まつり会場の栗原市岩ケ崎馬場通と栗駒総合支所の位置関係, 及びネットワークの敷設経路を図 2.2 に示す. また設営の

表 2.1 利用機材

八木アンテナ	WLE-WG-DYG
	BUFFALO アクセスポイント
無線アクセスポイント	WAPM-APG300N 2 台
同軸ケーブル	WLE-CC5 2本
	ThinkPad(ubuntu 10.04 LTS)
PC	MacBook(MacOSX SnowLeopard)

様子を図 2.3, 2.4 に示す. テント-アンテナ間, アンテナ-屋内間, 屋内配線は UTP を配線した. 利用機材の型番を表 2.1 に示す.

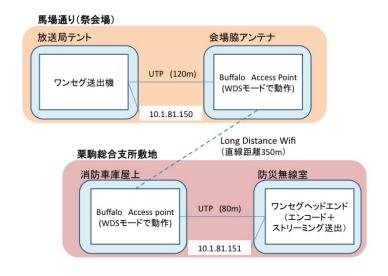


図 2.1 ネットワークトポロジー

構築したネットワークにおいて、アクセスポイントでの測定値の $RSSI^{*1}$ の取得、ping によるパケットロス率の測定、iperf によるスループットの測定をした.ping はパケットサイズ 64byte 送出間隔 1 秒で送信し計測した.iperf は TCP 利用の window size 256kbyte で 10 回計測した.

^{*1} 受信信号強度

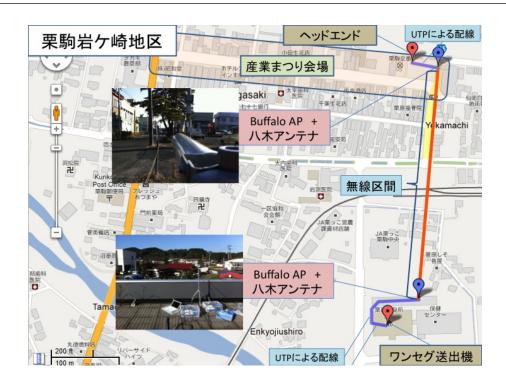


図 2.2 会場と栗駒総合支所の位置関係とネットワークの敷設経路

2.1.2 結果

ネットワーク構築

ネットワークの構築手順は以下の4項目にまとめることができる.

Step1. 事前調査

ネットワークの需要の調査をしネットワークを構築すべきエリアを決める. エリア間を結ぶ ために, アンテナを設置する場所が立ち入りが可能であるのか, 無線で結ぶ 2 点間の見通し があるのか, 電源の確保ができるのか, 以上 3 点を確認した上でアンテナを設置できるか判 断する. 設置不可能であれば, 再度設置位置の再検討をする.

Step2. 設置位置の検討/ネットワークの設計

利用可能なアクセスポイントの数, アンテナの数, UTP ケーブルの長さ, 接続用の PC の数を把握する. 地域においてネットワークが必要な場所を割り出し, 地図に書き出し, ブリッジ用のアクセスポイントの位置, アンテナの設置位置, 配線, サービス用アクセスポイントの配



図 2.3 栗駒総合支所のネットワーク

置を考慮する。全ての機材が配置できることを確認した上で、設営場所の決定とネットワーク設計をする。

Step3. 設置/アンテナ調整/導通確認

決定された場所に機材を展開し設置する. RSSI を計測しながらアンテナの方向合わせをし、 最良点で固定し、導通を確認する. その後、必要に応じて建物内の配線をし、ネットワーク サービスを提供可能な環境を構築する.

Step4. 運用

設置されたネットワークの導通確認やスループットなどネットワークの状態を計測すること で障害への対応をする.

RSSI とスループットの計測

10月29日のネットワークの設営直後に RSSI とスループットを計測した. RSSI の最良値を示す方向にアンテナを調整し, 計測した結果 RSSI は web インターフェースの目視確認で-74dBm であった. スループットは連続して 10回の計測をし, 平均32.32Mbps の帯域であった. 10回のス



図 2.4 産業まつり会場のネットワーク

ループット測定の結果を表 2.2 に記載する.

Round Trip Time とパケットロス率の測定

Round Trip Time とパケットロス率の測定結果を記載する. Round Trip Time の時系列の変化を図 2.5, 1 分間毎のパケットロス率の時系列の変化を図 2.6 に示す. Round Trip Time に関して, -1 はパケットロスしたことを示している.

2.1.3 考察

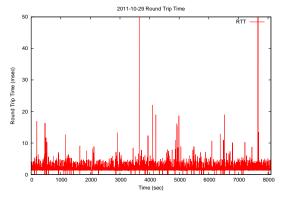
第 2.1.2 項より、WiLD Net 構築の際に予想される問題を以下の 3 点にまとめる.

問題点 1:事前機器設定

現地での機器の設定は多くの時間を要する. そこで, 事前準備として機器の設定はアクセスポイントの設定と導通確認をする必要がある. 今回の運用実験に際して, 用いる機器は, アクセスポイントを設定し, 導通確認をしたものに加え, アンテナやケーブル, UPS 付きスイッ

丰	2.2	計測帯域
11	Z.Z	

Interval	Transfer	Bandwidth
0.0-10.0sec	40.5MBytes	33.9 Mbits/sec
0.0 - 10.0 sec	41.5MBytes	34.8Mbits/sec
0.0 - 10.0 sec	41.3MBytes	34.6Mbits/sec
0.0 - 10.0 sec	31.6MBytes	26.5 Mbits/sec
0.0 - 10.0 sec	38.9MBytes	32.6 Mbits/sec
0.0 - 10.0 sec	38.8MBytes	32.6Mbits/sec
0.0 - 10.0 sec	39.0MBytes	32.7 Mbits/sec
0.0 - 10.0 sec	35.0MBytes	29.4Mbits/sec
0.0-10.0sec	39.5MBytes	33.1Mbits/sec
0.0-10.0sec	39.4MBytes	33.0 Mbits/sec



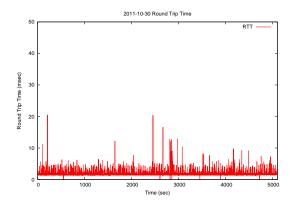


図 2.5 パケットの遅延計測結果

チをまとめた WiLD Net キット (図 2.7) として準備をした. このように事前の準備や機器 設定により簡略化が可能であることが明らかになった.

問題点 2:設置場所

設置場所の下見に多くの時間を要した. 構築運用実験をする前に下見/事前調査を行い, 実際にアンテナを設置し導通確認をした上で, 設置場所の決定をした. 建物への立ち入りが可能であるか, 設置地点と反対側の地点との見通しが可能であるか, 電源の位置, ネットワークの

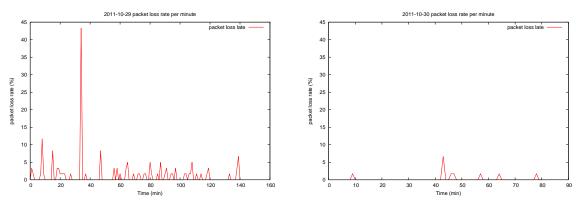


図 2.6 1分毎のパケットロス計測結果

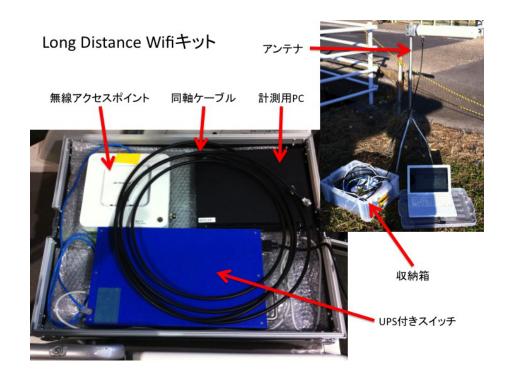


図 2.7 Long Distance Wifi キット

引き込みが可能であるかの4点の場所の制約が事前準備において困難であった.

問題点 3:導通確認

障害物や電波伝搬特性によって、実際に電波測定をし、導通確認をしなければ、ネットワークを構築可能か判断できない。電波強度計測ツールによる計測や実際にアンテナを設置し導通確認をする必要がある。

以上より、WiLD Net を構築する要素として、場所の制約が設置位置の決定を困難にしている.このため、入念な現地調査が必要となり、多くの時間を費やしてしまうため、迅速にネットワークを構築することは難しい.

また、事前にアクセスポイントの設定、電波測定機器の活用、持ち運び可能なバッテリーの事前準備があれば、ネットワークが構築可能であると考えられる.

表 2.2 に示されたスループットの計測では、10 回の計測で最小 26.5Mbits/sec から最大 34.8Mbits/sec と差が出たが、パケットロスが観測されなかったため、RSSI の低下によるアクセスポイントの送信レートの低下が、 *2 スループットの低下に影響していると考えられる.

同一機材で同一の場所に設置したネットワークで 2 日間の運用中の計測では,図 2.5,図 2.6 に見られるように,それぞれの間でパケットロスのパターンや遅延の発生の仕方が見られた.両日とも同一の場所に設置し,天候も両日とも晴れだったため,指向性アンテナの向きのずれによる RSSI の減衰,見通し上の障害物への抵触による RSSI の減衰があげられる.

2.2 RSSI とネットワークパフォーマンスの関係性測定実験

本節では、第 2.1 節で明らかになった、RSSI の低下がネットワークのスループットやパケットロス率に与える影響について追実験により明らかにする.

2.2.1 手法

WiLD Net におけるアクセスポイント間を無線伝送した場合,得られる RSSI の値とネットワークのスループットやパケットロス率が変化する.本実験では,RSSI 毎のパケットサイズとパケットロス率の関係,RSSI 毎のパケット送出間隔とパケットロス率の関係,RSSI 毎のスループットを計ることで,ネットワークのパフォーマンスの低下の要因を調べる.ネットワークの構成は図 2.8 に示すように,計測は外部要因を除くために,無線区間は同軸ケーブルで接続し,間にアッテネーターを挟むことで擬似環境を構築した.利用機材は表 2.3 に実験の様子は図 2.9 に示す.

BUFFALO アクセスポイントは RSSI を表示する web インターフェースが存在する. 減衰量を変化させた時の, アクセスポイントにおける RSSI をそれぞれデータを取得した. 両端の PC の片方から, ping で ICMP 2000 パケットを送信しパケットロスを計測する. ping の送出間隔を 1 秒, 0.1 秒, 0.001 秒と変化させ, さらに各々で ICMP のパケットサイズを 64byte, 500byte,

^{*&}lt;sup>2</sup> cisco の RSSI と送信レート

表 2.3 利用機材

八木アンテナ	WLE-WG-DYG	
無線アクセスポイント	BUFFALO アクセスポイント WAPM-APG300N 2 台	
同軸ケーブル	WLE-CC5 2 本	
アッテネーター	Agilent(固定 40dB 減衰) 2 個	
	Agilent(可変) 1 個	
PC	ThinkPad (ubuntu 10.04 LTS)/ MacBook (MacOSX SnowLeopard)	

1000byte, 1500byte と変化させパケットロス率を測定した. その後 iperf を用い TCP の帯域を計測する. 帯域は 10 回計測しその平均を用いた.

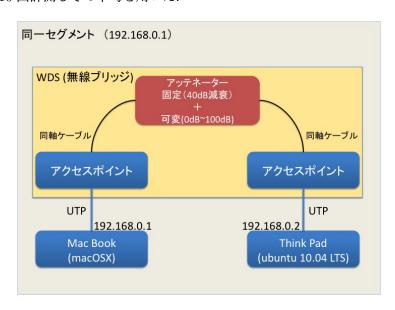


図 2.8 計測用ネットワークの構成

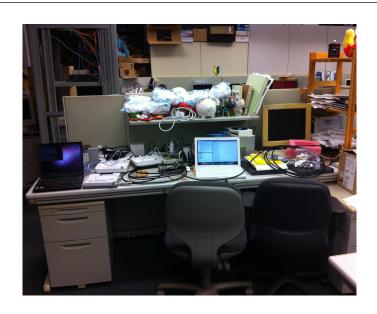


図 2.9 計測用ネットワークの様子

表 2.4 減衰量と Access Point の受信電力の関係

可変アッテネーターの減衰量	受信アクセスポイントにおける RSSI 表示値		
25 dB	-70 dBm		
40 dB	-80 dBm		
45 dB	-82 dBm		
50 dB	-85 dBm		
52 dB	-82 dBm (計測不能)		
55 dB	-82 dBm (計測不能)		

2.2.2 結果

可変アッテネーターの減衰量に対するアクセスポイントでの RSSI の測定値を表 2.4 に記載する. 図 2.10, 図 2.11, 図 2.12, のグラフの減衰量はアッテネーターの値のため, 変換には表 2.4 を用いアクセスポイントでの測定値を参照する.

減衰量を増加させた (RSSI の低下させた) 時のスループットの変化を示す。RSSI ごとの帯域を図 2.10 に記載する。尚,図 2.10 の横軸は可変アッテネーターの減衰量であり,固定アッテネーターの減衰量は含まない。

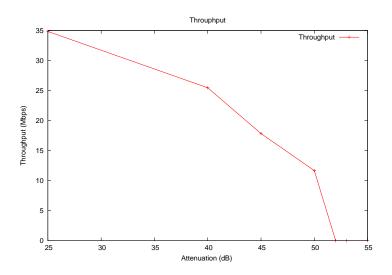


図 2.10 RSSI ごとのスループット

RSSI 毎のパケットサイズとパケットロス率の関係, RSSI 毎のパケット送出間隔とパケットロス率の関係を図 2.11, 図 2.12 に示す.

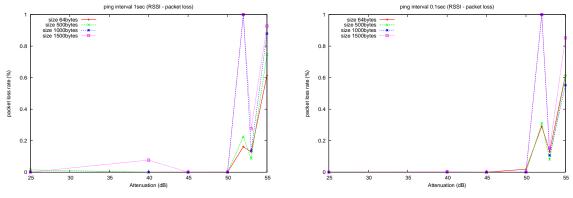
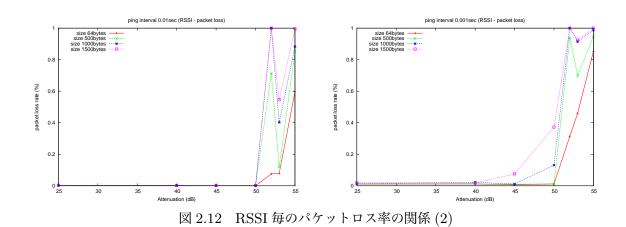


図 2.11 RSSI 毎のパケットロス率の関係 (1)



2.2.3 考察

図 2.10 より可変アッテネーターの値が 50dB を境に、性質が異なることが分かる.

可変アッテネーターの値が 50dB より小さい場合において、減衰量を増やすごとに緩やかなスループットの低下が見られる。これはアクセスポイントの RSSI によるデータレートの低下によって、スループットが低下したと考えられる。RSSI に対して送信するデータレートは変調やエラー訂正によって決定されるためである。

図 2.11, 図 2.12 より可変アッテネーターの値が 50dB より大きい場合において, スループットが 急激に低下し, パケットロス率が増大していることが分かる. これは通信できる RSSI を得られてい ないため, パケットロスが発生したと考える. また, パケットサイズが大きい方が全体としてパケットロスが大きく, 送出間隔が短いほどパケットロスが大きい.

WiLD Net を運用する上で、安定した通信をするためには十分な RSSI を得る必要がある. 十分な RSSI を得られない場合、スループットの低下や設営時に導通しないことや降雨減衰 [5] など通信状況が突然悪化し通信断絶を引き起こすことが明らかになった.

2.3 無線ネットワークの特性

本節では、WiLD Net やその他無線マルチホップネットワークの研究事例から、WiLD Net の設 営場所を決定するための要素を明らかにする.

2.3.1 スループット低下の要因

Wildnet: Design and implementation of high performance wifi based long distance networks[6] は、スループット低下の要因に干渉と衝突があると述べている。 FOMA の無線ネットワーク設計概要 [7] では、ネットワークのエリアの設計やビームの向きや幅によって、干渉が発生し容量の低下が発生することも述べられている。 5 チャネル以上近接したチャネルを利用すると相互に干渉が発生し、スループットの低下が発生する。 また反射波の到達など地形や電波伝搬特性によって干渉が引き起こされる可能もある。 チャネルの分離やアンテナの方向、設置場所、送信出力の設計が必要である。

また、CSMA/CA による衝突回避のアルゴリズムでは、DIFS*3の関係上、リンク間の距離が増大すると衝突回避が困難になる。802.11b の場合は 15km, 802.11g の場合は 8.4km を超えると衝突回避が困難になる。

Architecture and evaluation of an unplanned 802.11 b mesh network[8] や Packet loss characterization in WiFi-based long distance networks[9] では無線マルチホップを行う際に、ホップ数の増加はスループットの低下に繋がると述べられている。stop-and-wait を基本とする OSI 参照モデル第 2 層の IEEE 802.11 のプロトコルは、WiLD Net の環境において伝搬遅延や CSMA/CA における待ち時間が増大する。ホップ数が増加するにつれ、待ち時間や伝搬遅延が蓄積 することで伝送遅延が増大する。このためタイムアウトの発生による再送などからスループットが 低下する。

2.4 WiLD Net **構築における考察**

第 2.1 節より場所の制約として建物への立ち入り、見通しの可否、電源の調達、目的地までのネットワークの引き込みの可否の考慮が必要であることが明らかになった。またそれらを調査するために、現地調査に多くの時間を割く必要があった。第 2.2 節より十分な RSSI が必要であることが明らかになった。RSSI は電波伝搬特性により伝送距離が長いほど低下することが明らかになっている。第 2.3 節より干渉、衝突、ホップ数の低減が必要であることが明らかになった。これらを低減するた

^{*3} Distributed InterFrame Space:CSMA/CA 搬送波の有無を認識する際に, ビジー状態のチャネルから 信号電力が検出されなくなり, アイドル状態に変化したと判断されるまでの時間間隔 (802.11b は 50msec)

めに、チャネル設計や伝送距離の低減、ホップ数を考慮しネットワークの設計が必要である。ネットワークのスループットの低下の要因を図 2.13 で整理する.

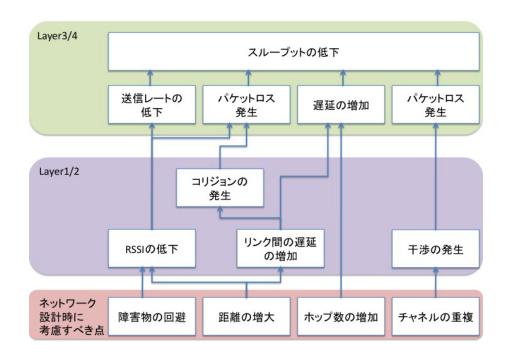


図 2.13 スループット低下の要因の整理

図 2.13 より、ネットワークのスループットは、場所の制約に関わる見通しが可能や距離、ホップ数、チャネルの重複が密接に関係し、ネットワークトポロジーのみでなく、実空間上の場所の制約に影響を受ける。要するに、場所の制約とネットワーク設計を同時にする必要があり、これらを全て満たす適切な場所に設置場所を決め、WiLD Net を構築することは難しく、ネットワークを設置する場所の数が増加するに伴い困難になる。

2.5 まとめ

本章では、WiLD Net の設置/運用の難しさを WiLD Net の構築/運用や基本的な特性の追実験やサーベイにより明らかにし整理をした。高スループットの WiLD Net を構築するためには場所の制約とネットワークの設計の双方の考慮が必要でありこの点が課題であった。これらを踏まえ、第3章では WiLD Net の設置について関連研究や関連事例を挙げながら、本研究の位置づけをする。

第3章

関連研究/事例/製品

本章では、WiLD Net 構築について被災地での実際の運用やネットワーク構築を支援するシステムについて述べ、本研究との比較をする.

3.1 **被災地における** WiLD Net **構築事例**

3.1.1 Lifeline Station/PDRNET

Life Line Station[1](図 3.1) は、被災直後に専門家でない人が衛星通信や WiFi、ワンセグなどを 用いて、ネットワークを即座に構築できるシステムである。 Life Line Station で培った技術を用い、 PDRNET[4] プロジェクトでは東日本大震災の被災地で、衛星通信/3G WiFi Bridge/WiLD Net などを活用し、インターネットは被災地にて構築された。ネットワークに加え、PC や IP 電話など の端末の提供もし、2012 年 1 月現在も必要な拠点に対し支援を行っている。

本プロジェクトでは衛星通信や 3G-WiFi ブリッジをゲートウェイにし, 無線のアクセスポイントや UTP の配線, WiLD Net などを使い, ネットワークの接続性と拡張性が示された.

一方で、現地調査ではネットワークの需要の調査と設置可能性の調査に時間を費やした。また、建物間のネットワークを WiLD Net で接続を試みたが、構築の際に十分な RSSI を得られず構築を断念した。

3.1.2 Inveneo/Nethope の取り組み

Inveneo/Nethope のチームは, 2010 年 1 月 12 日に発生したハイチ地震 (死者約 31 万 6000 人) において, WiLD Net をハイチの首都ポルトープランスを中心に構築した [3]. ネットワークの規模



図 3.1 Life Line Station の概要図

は、最大伝送距離 82km、1.6Mbps 程度のスループット、無線基地局を最大 3 ホップのネットワークを構築した。主たる構成を図 3.2、図 3.3 に示す。

Inveneo の記事"How to Deploy Long-Distance WiFi in Haiti"[10] では、WiLD Net を構築 する際に現地調査に時間を割き、アンテナの設置場所を決定することに時間をかけたと述べられて いる。また既存の地図は建物の倒壊などで利用できず、ボランティアが更新し続けたオープンマップ が役立った。電波伝搬をシミュレートする Radio Mobile[11] や Google Earth[12] を用い、ネット ワークの設計を行った。

3.2 WiLD Net の構築支援

WiLD Net の設計や構築に利用可能な研究やツールをとりあげる.

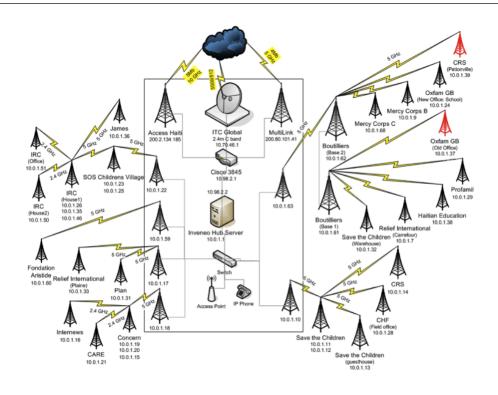


図 3.2 ハイチ地震時に構成されたネットワーク (1)

3.2.1 ネットワークトポロジーの自動生成

Long Distance Wireless Mesh Network Planing: Problem Formulation and Solution[13] において、Long Distance Wireless ネットワークを構築する上で、アンテナ設営コストなど金銭的コストの最小化とネットワークパフォーマンスを最大化させるネットワークの生成を自動的にするアルゴリズムを提唱している。トポロジーの組み方、アンテナの高さ、アンテナの種類、出力電力の大きさをメトリックとして、ヒューリスティックアルゴリズムを用い、ネットワークトポロジーや利用アンテナの種類、高さ、出力電力を決定している。ノード数が増えても対応可能で WiLD Net を構築する際に、有効な研究である。



図 3.3 ハイチ地震時に構成されたネットワーク (2)

3.2.2 電波伝搬シミュレーション

アジア航測株式会社の製品電波伝搬シミュレーションソフト [14] は見通しの計算や干渉計算など電波伝搬に関して高い精度でシミュレーションが出来る. 利用地図データは, 全国対応で 250m メッシュ/50m メッシュであり小エリア対応の 4m メッシュ地図も利用している. 地理情報のプロファイルなどを活用し, 実際の現場でどのくらいの RSSI が取得出来るかなど, シミュレートでき, ネットワークの設計を支援できる. 他にも, HF 帯の電離層反射伝搬を Ham Cap[15] VHF 帯以上で山岳回折や反射を考慮した伝搬予測 Radio Mobile は [11] も活用可能である.

NTT DoCoMo においても電波伝搬推定のための遺伝的アルゴリズムを用いたレイトレーシング 法 [16] において電波伝播を高速演算し、基地局からの電波伝搬推定をする.

	場所の制約			
				ネットワークの
	建物への立ち入り	見通し	電源の調達	引き込み
Inveneo/NetHope	現地調査	現地調査	現地調査	現地調査
PDRNET	現地調査	現地調査	現地調査	現地調査
Mesh Network Planning	-	0	-	-
電波伝搬シミュレーター	-	0	-	-

表 3.1 解決手法の比較 (設置場所の制約)

3.3 考察

第2章で述べた場所の制約の考慮とネットワークの設計について,第3.1.1節,第3.1.2節の事例では,現地調査と手作業で行い, 事前の準備や調査時間がかかることが問題となる.また接続数が増えるにつれ,ネットワークの適切な設計が難しくなる.

第3.2.1 節で述べた,トポロジーの自動生成の研究は,鉄塔の建設など定常運用に対して効果を発揮する.本研究では,被災地で建物の屋上に機材を設置することを考えるため,場所の制約を考慮できない.

第 3.2.2 節で述べた,電波伝搬シミュレーションは,電波伝搬特性からアンテナの利得や送信電力, 位置/高度情報からそれぞれのアクセスポイントの RSSI を,シミュレーションにより推定できる.

被災地における WiLD Net 構築の観点からこれらの手法の比較をする. 場所の制約への解決手法を表 3.1, ネットワークの設計に対する解決手法を表 3.2 に示す.

表 3.1, 表 3.2 より, 被災地においてマルチホップさせる WiLD Net を構築する場合において, 場所の制約とネットワークの設計を考慮したものは存在しない.

3.4 まとめ

本章では、WiLD Net 構築について関連研究/事例/製品について述べた.表 3.2、表 3.1 より場所の制約とネットワークの設計の双方を考慮し、WiLD Net を構築する必要があることが明らかになった.

表 3.2 解決手法の比較 (ネットワークの考慮)

	ネットワークの考慮			
	RSSI	伝送距離	ホップ数	チャネル分け
Inveneo/NetHope	計測	手動	手動	手動
PDRNET	計測	手動	手動	手動
Mesh Network Planning	0	0	0	0
電波伝搬シミュレーター	0	-	-	-

第4章

アプローチ

本章では、問題点について、第2章で述べた WiLD Net を被災地で構築する場合の課題や第3章で述べた関連研究/事例/製品と比較し、本研究のアプローチを述べる.

4.1 WiLD Net **構築の**課題の整理

第 2 章, 第 3 章で述べた WiLD Net 設計における課題は被災地を考慮した WiLD Net 構築, 場所の制約とネットワーク全体の設計であった.

被災地を考慮した WiLD Net 構築の問題点は, 鉄塔など建造物の建築が難しく, 建物の屋上など に機材を設置するため, 見通しが現存の建物に依存し各地点において接続が可能か地図上では判断が難しい. このため, 建物の制約や見通しに応じたネットワークの設計が必要となり, 事前の現地調査に時間を割く必要がある. また, ネットワークトポロジーも設置場所の見通しに依存するため, 高スループットの WiLD Net の設計が難しい. 場所の制約の主な要素は以下の4つであった.

- 建物への立ち入り
- 見通し
- 電源の調達
- ネットワークの引き込みが可能か

また、ネットワークの設計で考慮すべき点は以下の4点であった.

- 見通し
- 伝送距離
- ホップ数
- チャネル分け

4.2. 機能要件

場所の制約とネットワークの設計それぞれの要素が、被災地において迅速に高スループットのWiLD Net の構築を難しくする要因となる.

4.2 機能要件

本節では、これらの点を解決するための機能要件について述べる。第 4.1 節より機能要件は以下の 2 点である.

- 1. 場所の制約とネットワーク設計を考慮した WiLD Net の設計
- 2. 計算時間が現地調査や手作業での設計よりも高速である

4.3 **アプローチ**

本節では,第4.2節の機能要件より本研究において構築する WiLD Net 機材設置支援システムのアプローチを述べる.

WiLD Net を設置する地域の位置情報を予め地図情報から取得しデータベース化をする. 与えられた WiLD Net を設置したい拠点から, 事前に取得してある地理情報を元に, 見通し計算をし接続可能かを評価する.

場所の制約は、被災地の状況を知る利用者の知識を織り交ぜながら、対話しながら設置位置を決定する. ネットワークの設計は孤立ノード数、ホップ数、距離、チャネルの分割を評価し、ツリー構造を構築し、スループットの低下要因が最も少ないトポロジーを形成し、設置地点と共に地図上に表示する.

また、地理情報を持っているデータベースを公開し、データベースに書き込めるユーザーインターフェースを構築することで、被災地で刻一刻と変化する状況に対応し、随時情報更新を行なう.

4.4 まとめ

本章では、WiLD Net 機材設置場所についての問題点を整理した上で、WiLD Net 機材設置支援システムに求められる 4 つの機能要件を明らかにした。第 5 章では WiLD Net 機材設置支援ツールのシステム設計について述べる。

被災地における WiLD Net 機材設置支援ツールの機能要件は,被災地の状況に合わせた情報更新,場所の制約とネットワーク設計,設置数の増加への対応,一般の方でも分かる情報提示であった.

4.4. まとめ

本章ではこれらの機能要件を満たすために、地図情報をデータベース化し、ユーザー対話しながら 設置位置を決定する.

次章では、システムの詳細な設計について述べる.

第5章

設計

本章では、第4章を踏まえ WiLD Net 構築支援システムの設計を行なう.

5.1 システム全体の設計

第 4.2 節の要件を満たすために、本システムはデータベースと以下の 5 つのモジュールを実装する.

- 1. 地理データ取得モジュール
- 2. 建物検出モジュール
- 3. ユーザーインターフェース
- 4. データベース更新モジュール
- 5. 位置推薦モジュール

地理情報取得モジュールは、事前に地図情報から緯度/経度/建物の高さ/地表の高さの情報を取得し、データベースに格納する。データベースより、建物の識別しそれぞれに建物 ID を付与し、データベースに格納する。この 2 つのモジュールにより、計算するための基礎データの取得/加工をする役割をもつ。

ユーザーインターフェースはデータベース更新モジュールや位置推薦システムと連携をし, ユーザーに地図ベースの視覚情報で情報を提供する.

データベース更新モジュールは、建物の倒壊や新設など実空間とデータベース上の地図情報の乖離を更新によって防ぐ役割を持つ。ユーザーインターフェースより、更新したい場所を指定し、高さの指定や倒壊など状況を入力し、更新することで変化する被災地の状況を最新のものに更新する。

位置推薦モジュールは、見通しの計算やユーザーの知識の利用により、場所の制約を考慮し、ネットワークの設計の自動計算により、地図上に設置場所を明確に表示することで、実際に構築可能かつ高スループット WiLD Net の迅速な構築の支援をする。ユーザーインターフェースより受け取った、ネットワークを設置したい拠点及び設置場所を地図上で指定する。

5.2 各モジュールの詳細

本節では、各モジュールの詳細を述べより細部の設計を明らかにする。データベースと 5 つのモジュールの詳細を以下に示す。

データベース

本データベースは、地図データに依存せず、本システムを動作させるための計算の基礎データとして構築する. 対象地域の緯度/経度/地表の高さ/建物の高さを結びつけて格納する. また建物を建物検出モジュールを利用して、建物を一意に識別するための ID を付与し格納する. 更に被災地域でも活用できるように、データベース更新モジュールを利用し、建物の倒壊状況をステータスとして格納する. 本データベースは、各モジュールからアクセスされ読み書きをする.

地図データ取得モジュール

本モジュールは、地図データを取得し、データベースに格納出来る形に変換し格納する.

建物検出モジュール

データベースに格納された緯度/経度/建物の高さ/地表の高さの各点のデータから建物と識別する. 建物の高さと地表の高さの差が 1m を超えるものを建物として識別し, 隣り合う点が建物だと識別された場合同一の建物として一意に識別できる ID を付与し, データベースに格納する.

ユーザーインターフェース

地図を中心に視覚にわかりやすい形でユーザーインターフェースを構築する. 地図を利用して入力された地点の緯度/経度や入力された高さをデータベース更新モジュールへの送信やWiLD Net を設置したい場所や中継点の緯度/経度を位置推薦モジュールに送信し, 計算させるさせ, 計算結果を地図上で表示するなどそれぞれのモジュールと連携をはかる.

データベース更新モジュール

データベースをインターネット上で共有し、衛星写真などから被災地域の建物の破壊などを ハイチ地震での事例のようにボランティアによるオープンマップの更新 [3] のような仕組み で、更新する. ユーザーインターフェースより入力された緯度/経度からデータベース内ので同一の建物 ID を持つ建物の建物の倒壊フラグを倒壊に変更する. また、同様にデータベースに格納されていない建物も地図上で範囲を指定し、建物の高さを入力することで、指定された範囲内に入力された高さの建物のデータを格納も出来る.

位置推薦モジュール

ユーザーインターフェースから受け取った,設置候補地点から位置推薦をするためには,場所の制約やネットワーク全体の設計を考慮し,設置位置や中継点を決定する必要がある.2 地点間の緯度/経度からその2地点間の見通し計算,中継点候補の評価とユーザーの知識の利用による中継点の決定,設置位置の決定の後のネットワークトポロジーの自動生成し,設置位置とネットワークのトポロジーを地図情報として,ユーザーインターフェース上で表示する.

5.3 各モジュールの連携

システム図の概要を図 5.1 に示す。第 5.2 節で述べたモジュールが図 5.1 のように連携し動作する。ユーザーが入力した地点から計算し、機材の設置位置と WiLD Net のトポロジーを形成し、地図上で表示する。

5.4 まとめ

本章では、第4.2節の機能要件を満たすために必要なシステムを明らかにし、それぞれの設計について述べた。第6章では設計を実現するための詳細な実装について述べる。

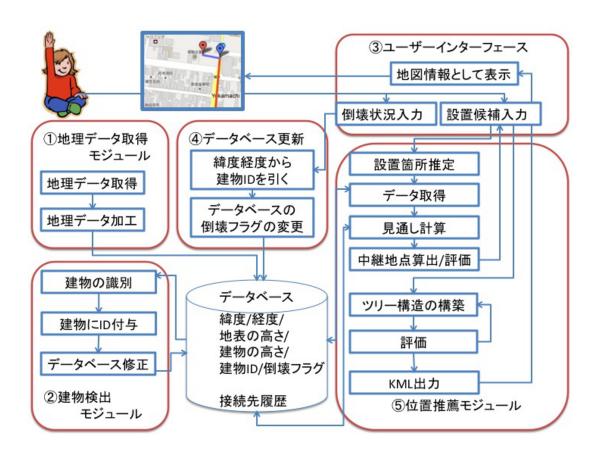


図 5.1 システム設計図

第6章

実装

6.1 各モジュールの実装

第5章の設計を実現させるための実装の詳細を述べる.

6.1.1 データベース

全てもモジュールからアクセスされるデータベースは MySQL により構築する. テーブルの設計 は表 6.1 に緯度/経度/地表の高度/建物の高度を地図から得られるデータ, 建物 ID/建物ステータスは, 建物検出モジュールやデータベース更新モジュールより更新されるデータとして, 紐付け各地点の情報を保持する.

6.1.2 地図データ取得モジュール

google earth3D plugin を用い, ある地点の緯度/経度/地表の高度/建物の高度を取得する. 緯度/経度/建物の高さ/地表の高さは getLatitude/getLongitude/getAltitude/getGroundAltitude

表 6.1 地図情報を格納するテーブルの設計

データ	緯度	経度	地表の高度	建物の高度	建物 ID	建物ステータス
カラム	lat	lng	ground	height	id	status
データ型	double	double	double	double	int	int

関数より取得する.このデータの取得はマウスでクリックした部分のみ変数に格納される仕様のため,ブラウザで表示した地図情報を,2011 年 12 月現在 google earth3D plugin は MacOS と Windows のみ対応しているため,MacOSX の PC を用いクリックをエミュレートするソフトウェア cliclick[17] を活用し,ブラウザ上でカーソルを 1 ピクセルずつずらしながら,ブラウザに表示されている地図をクリックし,データを変数に格納する.データは HTTP の POST を利用してサーバーに送信しデータを保持する.本モジュールの動作概要を図 6.1 に示す.送信されるデータは,クリックされた場所の緯度/経度/地面の高さ/建物の高さである.

送信されたデータは, 小数点以下 5 桁に桁合わせ (5 桁目は偶数に値を丸める) 格納する. この桁は緯度/経度から距離を求めるヒュベニの公式より, 東経 139 度, 北緯 35 度付近では, 南北方向約 2.22m, 東西方向約 1.82m の幅となる.

本研究では、実験として慶應義塾大学三田キャンパス付近を中心に南北約 1.7km、東西約 1.8km のデータベースを構築し実験する.



図 6.1 地図情報取得

6.1.3 建物検出モジュール

データベースに格納された緯度/経度/建物の高さ/地表の高さの各点のデータから建物と識別する. 識別した建物に対して一意に識別できる ID を付与し, データベースに格納する. 建物識別のためのアルゴリズムは, 地表のデータと建物の高さのデータの差が 1m 以下のものを建物とみなし, 隣

り合う高さ 1m 以上の緯度/経度の地点を同一の ID を付与し格納する. データベースに格納された 地面の高さと建物の高さのデータの差分が 1m 以上である部分は, 建物であると認識する. 緯度/経 度を平面とした座標として扱う. 隣り合った緯度/経度が建物であると認識された場合, それらの場 所は同一の建物とみなし同一の建物 ID を付与する. アルゴリズムを図示したものを図 6.2 に示す.

緯度		40.0		222						ľ
件及	10.5	10.2	9.2	8.5	0.1	0.1	6.1	6.5	6.5	
1	10.2	10.1	9.2	8.4	0.3	0.2	6.2	6.3	6.2	値は地表の高さと 建物の高さの差
	10.2	10.1	8.9	8.6	0.2	0.9	6.3	6.4	6.3	
1	3.1	3.0	2.0	2.0	0.4	0.1	6.1	6.5	6.9	
	0.1	0.3	0.2	0.1	0.1	0.1	0.2	0.1	0.2	
	10.5	ID:35	5-135	3.5	0.1	0.2	0.6	0.5	0.2	
	10.5	10.8	9.3	8.3	0.2	0.2	6.6	6.4	6.4	
	12.0	10.2	9.2	8.5	0.2	0.4	6.8	6.2	6.2	
35.0000	10.2	10.1	9.2	8.6	0.1	0.2	6.5	6.1	6.9	
	135.000	0							→	経度

図 6.2 建物検出アルゴリズム

隣り合うものの高さが1m以上なら同一の建物とみなしIDをつける 緯度/経度が最も小さい部分の緯度/経度からIDを生成する

具体的なソースコードを付録 A に示す.

6.1.4 ユーザーインターフェース

図 6.3 にユーザーインターフェースの外観を示す. 汎用的なデータベースを用いているため, 変換をするのみで様々な地図に対応できるが, 本システムでは, 図 6.3 のように, CGI を用いユーザーに提示することを考えるため google maps を活用する. 図 6.3 の地図上をクリックすることで, 設置場所の候補地点を入力でき, 左側にあるボタンを押すことで, 機材の入力や中継点の候補算出, 設置場所の決定とネットワークトポロジーの生成をするモジュールに, データを送信し, その結果を地図に表示する. 表示された設置地点やトポロジーを表した線をクリックすることで, アンテナの方角や

設定するチャネルを表示する.動作概要は第6.2節で述べる.



図 6.3 ユーザーインターフェースの外観

6.1.5 データベース更新モジュール

ユーザーインターフェースより、通信部分は XMLHttp を利用し渡された緯度/経度から建物 ID を引き、同一の建物 ID を保持している建物ステータスを変更する. 同様に、緯度/経度情報と高さの情報から、それらを直線で結び領域としてとらえ、その内部にあるデータの高さと建物 ID を変更する.

6.1.6 位置情報推薦モジュール

位置推薦モジュールは、ユーザーインターフェースから取得した、設置場所の緯度/経度より設置 地点/中継点を決定し、適切なネットワークトポロジーを形成し、ユーザーインターフェースに返す。 この際に、見通しの確認、中継点算出、ネットワークの設計のためのツリー構造によるトポロジーの 生成をする。また計算結果をデータベースに格納することで、同一地点の結果は再度計算せず、デー タベースのデータを活用し、計算量を削減する接続可能位置記憶の構築をする.

見通し

2 地点間の見通しを計算に概要は、図 6.4 のように、2 地点間の緯度/経度からその 2 地点間の間にある高さのデータを取得し、2 点間の直線をその間にある高さが超えなければ、見通しがあるとみなす。

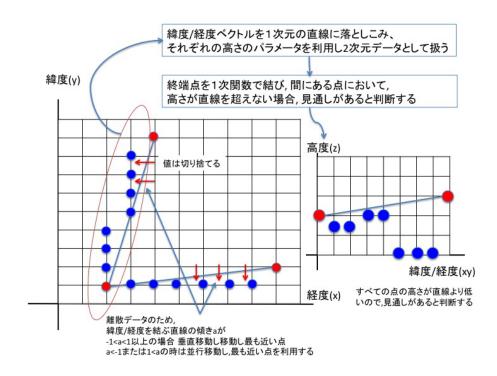


図 6.4 見通し計算

付録 B のソースコードは、Perl 言語で書かれた関数である。見通し確認関数は、2 地点の緯度/経度を引数として与えると、データベースのデータを用い、2 点間の見通しがとれるかどうかを判断し、見通しが取れる場合は返り値 1、見通しが取れない場合は返り値 0 を返す。データベースから読み込んだ緯度/経度をキーにして、建物の高さ/建物 ID が %height_latlng %id_latlng に格納されいる。これを利用し、直線上にデータを抽出し、高さと地表の 2 次元データとして扱い、2 地点間を直線で結ぶときに、間に障害物があるか調べる.

表 6.2 テーブルの設計

データ	建物 ID 1	建物 ID 2	ステータス	
カラム	$id_{-}1$	id_2	status	
データ型	int	int	int	

中継点候補算出

見通し確認関数を用い、指定された設置地点から全方向に全ての見通し可能な建物を算出する. その他の指定されたそれぞれの設置地点で同様の計算を行い、2点以上から見通し可能と判定された建物を中継点候補とする. その際に、中継点のおすすめ度をスコアとして算出する. スコアの算出の手法は、その中継点における設置地点の接続可能数を利用し、中継点をユーザーにであることを示す.

ツリー構築/評価

ユーザーインターフェースより、設置地点と中継点が与えられるとそれぞれの地点同士の見通しを計算し、見通し可能な場所を隣接行列として表現する。この隣接行列を用い、ネットワークの設計をするためのツリー構築/評価を行なう。隣接行列として与えると、干渉を防ぐための1地点の最大接続数を3つの制約の上で、形成される接続の数で孤立ノードが発生*1、頂点からのツリーの深さによるホップ数、実空間での平均経路長による接続距離、を評価し、ツリー構造を構築する。同時に、設置場所の緯度/経度からアンテナの方角を割り出しやツリー構築時にチャネルの設計をする。

接続可能位置記憶

中継点算出によって算出された場所や WiLD Net が一度構築された場所をデータベースに格納することで、再度計算をする手間を省く. 具体的には、接続可能な場所の建物 ID を紐付けデータベースに格納する. テーブルの設計は表 6.2 のように、接続可能な建物同士をヒモ付て格納する. ステータスはユーザー入力によるものかシステム上の計算によるものであるかを、識別するものである.

^{*1} 孤立地点が発生する場合は 1 地点の最大接続数 3 つの制約を無視する

6.2 動作概要

ユーザーの操作によるシステムの動作にについて図 6.5 に示す. 図 6.5 のように, ユーザーはネットワークを設置したい場所をクリックし, 家のアイコンを地図上に置き, その後ボタンを押すと中継点候補のデータが返却される. 返却された中継点候補を更にユーザーが選択し, ボタンを押すと機材の設置場所と接続方向, ネットワークのパスが地図上に表示される.

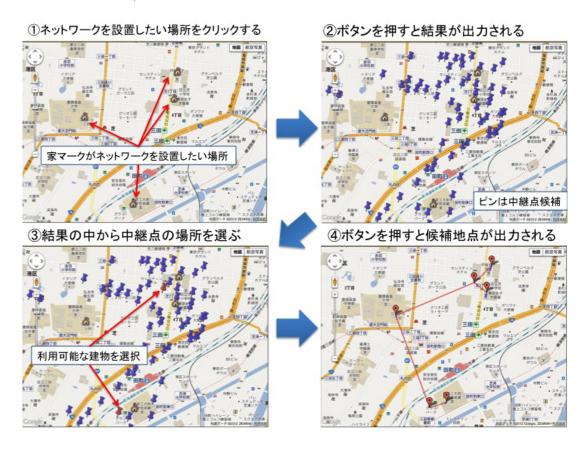


図 6.5 動作概要

6.3 実装環境

実装環境は Ubuntu サーバー上で perl 言語を用い開発した. またデータベースに mysql を用い, 地図情報の取得を google Earth 3D Plugin, ユーザーインターフェースの地図を google map API

表 6.3 実装環境

Mac OS X	10.6.8		
Ubuntu	$10.04.3~\mathrm{LTS}$		
MySQL	5.1.41-3ubuntu12.10		
perl	5.10.1		

をそれぞれ web ブラウザ上の javascript から操作し実装した. 詳細なバージョンを表 6.3 に記載する.

6.4 まとめ

実装環境は Ubuntu サーバー上で perl 言語を用い開発した。本章では、第5章の設計を実現するための実装の詳細を述べた。第7章では、本システムを評価する。

第7章

評価

本章では、第6章において構築されたシステムを評価する.

7.1 評価項目

第4.2節で述べた機能要件を満たすことを評価する.

被災地を考慮した WiLD Net 構築の評価として, 事前調査時間の短縮を評価するために以下の 2 つを評価する. システムが正しく見通し計算の正しさを示すことにより, 計算機による計算で見通しが確認できるを示す. 計算時間を評価する. 設置地点数を増加させた場合とデータベースの地図データのデータ量を増加させた場合のプログラムの実行時間及びメモリの消費量を評価する.

被災状況の考慮について、地図情報の鮮度の影響やデータベース更新モジュールの効果は見通し計算の評価の一部として評価する. 基礎的な地図データは建物の倒壊などにより情報が古くなる可能性がある. 本システムでは、第 6.1.5 項のデータベース更新モジュールによって、人によるデータベース更新をし地図情報を更新する.

高スループットのネットワークを構築は、設計の時点で機能要件を満たすように設計をした.場所の制約/ネットワークの設計の考慮は設置したい拠点や中継点の選択をインタラクティブに行い、ユーザーの知識を盛り込むことで、場所の制約を解決する.特に被災地においては、実際の現地を調査や建物の構造を理解している人の知識を利用し情報を得る必要があるため、これらの知識の活用し場所の制約の解決するように設計したことで、機能要件を満たす。またネットワークの設計は、中継点が選択されると、チャネルの分割、ホップ数、伝送距離を評価し、最適なものを選択するため、1設置場所において3つのチャネルのみの使用に加え、ホップ数最小の構成の中、伝送距離の平均が最小になるように設計するため、機能要件を満たしていると言える。RSSIは距離に反比例し減衰する

表 7.1 写真から判断できる見通し可能な建物の数

	建物数	割合
見通し可能	99	31.9%
見通し不可	143	46.1%
写真からの建物特定不可	68	21.9%

ため、伝送距離の最小化によって、一部機能要件を満たしていると言える. 以上より評価項目は見通し計算と計算量/計算時間/メモリの消費量を評価する.

7.2 評価結果

本節では、第7.1節で述べた評価項目に対する評価結果を記載する.

7.2.1 見通し計算の評価

手法

慶應義塾大学三田キャンパスメディアセンターの屋上から,本システムを用い1度ずつ360度全方向に見通し可能な建物を算出する.システムが判断し見通し可能な建物各々について,同屋上から360度全方向に撮影した写真を用い,実際に見通しが可能か比較し確認をし評価する.

結果

表 7.1 は、本システムが判断した見通し可能な建物数と写真から判断出来る見通しの出来る建物の数の比較である。本システムが見通し可能と判断したものの中で、実際に写真から見通しが確認できたものを見通し可能、見通しが確認できないものを見通し不可、写真から建物を特定出来ない*1場合は写真からの建物特定不可としている。表 7.1 より見通しの正確さは 31.9% となる.

^{*1} google maps の衛星写真を利用し実際の写真を位置座標系に落とし込んだため, 色や形に特徴のない建物 はどの建物か判断できない

考察

表 7.1 において、建物特定不可なものをを含め、見通しの正確さが 31.9% と低い. この原因は、データを取得している Google Earth において、3D データが無い建物が存在することである. 図 7.1 は、構築した地図データベースにおいて、建物検出モジュールにシステムによって検出された建物を全てマッピングしたものである. 図 7.1 より、地図上の全ての建物が検出されていないことが分

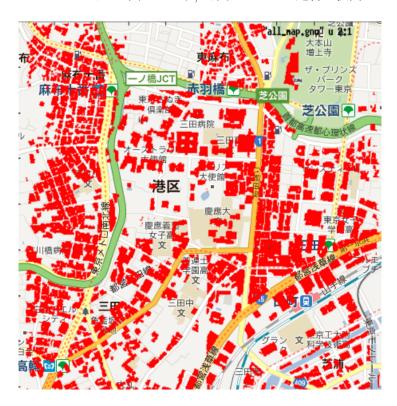


図 7.1 見通し評価データベース上にある全ての建物のマッピング

かる.

更に、慶應義塾大学三田キャンパスメディアセンターの南西に位置する、南校舎は建て替えられ 2011 年 3 月に竣工した。図 7.2 の屋上からの写真よりメディアセンターの屋上よりも高い一方で google earth のデータでは南校舎よりメディアセンターの方が高い*2. つまり、google earth の保持する 3D データが古い情報が最新の状況とずれがあることが分かる.

^{*2 2011}年1月調査

以上, 3D データの欠損, 最新でないことの 2 点が主な誤差を生む原因と考えられる.

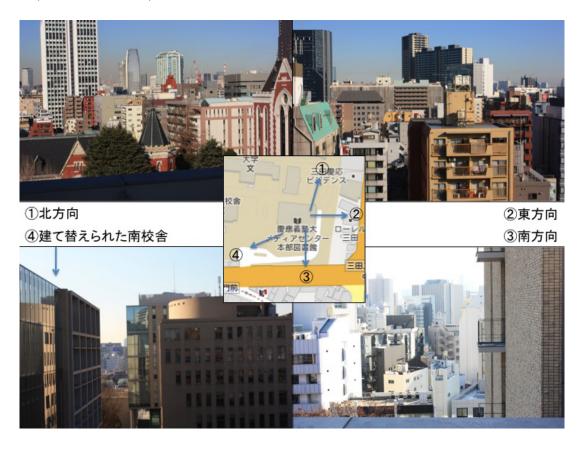


図 7.2 屋上からの見通し

この点に関して、データベース更新モジュールを利用し、慶應義塾大学三田キャンパス南校舎を追加し、再度見通しの評価を行った.

表 7.2 が更新モジュールを利用し、南校舎を追加し評価をした結果である.

見通しの正確さが少なくとも 52.9% 確保されていることが表 7.2 より分かる. これはハイチ地震の際にオープンマップがボランティアの手で, 建物の倒壊状況のなどが更新され続けたものが有効に活用されたことから, 地図情報の更新については, 実際の被災地環境でも効果があることが期待できる. また, 衛星写真から建物の倒壊状況を判定する研究 [18] 建物の立ち入りなど場所の制約については考慮できないため, 他の研究と比較しても本アプローチが有効であることも分かる. また, 地図メーカー大手の株式会社ゼンリン [19] における住宅地図の更新頻度は主要都市で概ね 1 年に 1 度改訂, それ以外の地域は 2 年に 1 度, または 3 年に 1 度改訂されている. 地図の更新頻度は概ね年単

表 7.2 データベース更新後の写真から判断できる見通し可能な建物の数

	建物数	割合
見通し可能	100	52.9%
見通し不可	21	11.1%
写真からの建物特定不可	68	36.0%

位であるため、慶應義塾大学三田キャンパス南校舎のように新しく建築された建物にもデータベース更新モジュールを活用しデータを更新が期待できる.

見通し不可である 11.1% の見通し不可な場所が見通し可能と判定されてしまう原因は, 森や林など地図データに無い障害物が存在することやデータベースの地図データの欠損による影になる建物の検出されてしまうことが主な原因である.

まとめ

これらの結果より、被災状況でない場合でも地図データの古さやデータの欠損が大きな原因となる.また、地図データにない森なども見通し判定を誤る原因となる.

これらはデータベース更新モジュールを利用し建物データを更新することで見通しの正確さを向上させることが可能であった。また、本研究では価格の観点から google earth のデータを用いたが、より精度の高いデータの活用やオープンマップ/データベースの仕組みを活用したユーザー/ボランティアによる情報の更新などを活用し、データベースが実世界に近づけられれば十分に活用可能である。

7.2.2 計算量/計算時間/メモリの消費量の評価

本節では、設置数を増加させた時の計算量、計算時間とそれに伴うメモリの消費量を評価する.また、データベースのデータ量を増加させた時の計算量、計算時間とそれに伴うメモリの消費量を評価する.

表 7.3 利用データ

南北	東西	面積	データ量
約 1.1km	約 1.7km	$1.87km^2$	487,627
約 1.7km	約 1.8km	$3.06km^{2}$	774,877

手法

表 7.3 に記載するデータを利用し、それぞれについて設置数を 2 から 100 まで 1 つずつ増加させ、ユーザ CPU 時間を計測し、同時にプロセスのメモリの最大消費量を評価する.ハイチ地震の活動 [3] や既存の WiLD Net 構築事例より機材の設置数が 100 を超えないため、設置数 100 地点までを計測対象とする.

本システムは、中継点を算出した際にデータベースに結果を格納するため、初期状態の実行時間を計測する必要がある。データベースのデータが空の状態で、ランダムな建物を設置数選択し、中継点候補の算出をさせる。最も計算量が多い設置候補地点選択から中継点候補算出における実行時間を計測する。中継点候補算出において、算出結果がデータベースに格納され、その後の計算に利用される。中継点をユーザーに選択させ、中継点候補算出から設置場所推薦までは、データベースのデータを利用するため、計算に時間に時間を要しない。したがって、中継点候補算出までの実行時間の計測とする。

結果

計算時間/計算量の結果を示す. グラフ中の node の数は設置数と対応する. 結果を図 7.3, 図 7.4 に示す.

見通し可能な場所を記憶するデータベースを利用せずに計測し、実行時間は、図 7.3 よりデータベースのレコード数 487,627 において、設置数 20 で約 50 秒、設置数 100 で計算で約 240 秒レコード数 774,877 において、図 7.4 より設置数 20 で約 71 秒、設置数 100 で計算で約 346 秒となる.一度計算した場所は建物毎に記憶され、データベースに保持されるため、計算時間はこれより少なくなる.

同時に計測した設置数を増加させた時のメモリの消費量を図7.5、図7.6に示す.図7.5では、メ

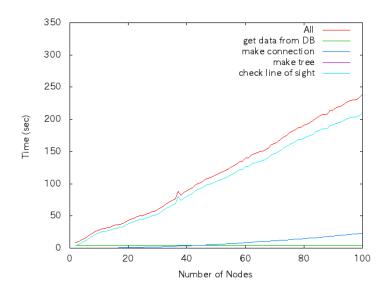


図 7.3 レコード数 487,627 のプログラムの実行時間

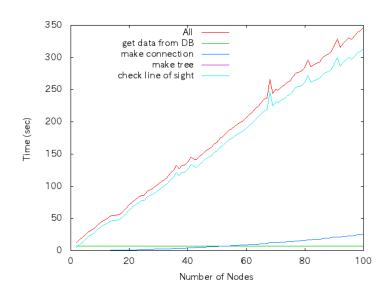


図 7.4 レコード数 774,877 のプログラムの実行時間

モリの使用量は 80Mbyte あたりを推移している。図 7.6 では、メモリの使用量は 130Mbyte あたりを推移している。

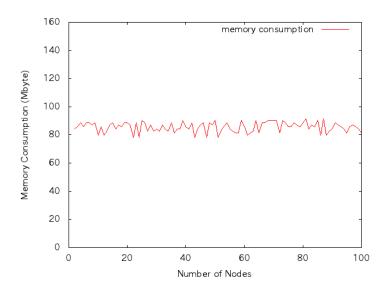


図 7.5 レコード数 487,627 の場合のメモリの消費量

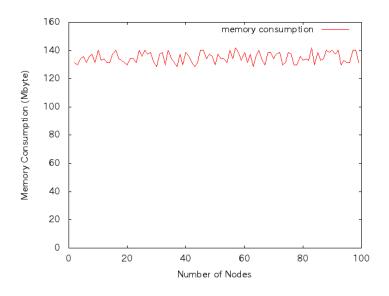


図 7.6 レコード数 774,877 の場合のメモリの消費量

考察

図 7.3, 図 7.4 において、All は全実行時間、残りはそれぞれの関数の実行時間である。 make connection 関数は、初期設置地点において、直接見通し可能かを確認する関数のため、設置数を n と すると n 行 n 列の隣接行列を走査するため、 $O(n^2)$ のオーダーとなる。一方で、check line of sight 関数は 360 度全方向に操作し、見通し可能な建物を全て抽出する関数は、設置数に対して O(n) の オーダーである。 ノード数 100 程度であれば、全実行時間は O(n) と見なすことができる。

これは、またメモリの消費量は、データサイズに比例し大きくなっている。計算を高速化するためにデータベースへの問い合わせ回数を減らすために、メモリ上に全データを読み込み計算をしていることに起因している。十分計算可能なメモリ消費量であるが、データベースのサイズが大きくなる場合は、実行時間との兼ね合いで、メモリ使用量を減らすアルゴリズムに変更が必要である。

実際の現地調査では、建物から見通しの確認や電源の取り回しの確認をし、見通しが取れない場合、別の建物を調査し、それを繰り返し、設置位置を決定する。慶應義塾大学三田キャンパスでの調査において、見通し確認をするために、一つの建物において最低30分以上の時間を要した事例より、*3. 本システムは、計算時間(ユーザーの入力時間を除く)で、見通しの調査の軽減やネットワーク設計の時間を省くことが出来、事前調査の時間短縮を満たす。100地点以上利用した大規模な被災地でのWiLD Netの構築事例はなく、計算時間は十分に早く、ノードの増加への対応は満たしている。

7.3 まとめ

本章では、第6章で実装したシステムが機能要件を満たしているか評価した. 本研究で、場所の制約とネットワークの設計の双方を考慮し、安価なシステムを構築できたと言える. これにより、被災地における WiLD Net 構築において、迅速なネットワークの構築に貢献が期待できる.

第8章では、本研究のまとめをする、

^{*3} 事前に建物の立ち入りの許可をもらい, 案内していただいた場合の時間

第8章

おわりに

8.1 本研究のまとめ

本研究はの目的は、被災地において WiLD Net を構築する際の課題を整理し、整理された要件に基づき、WiLD Net 構築のための機材の設置位置の推薦をするシステムの設計と実装をすることで、被災地域において高スループット WiLD Net の迅速な構築を支援するツールの開発することであった.

被災地において WiLD Net をを構築する際に、場所の制約とネットワークの設計の双方を考慮し、機材の設置場所を決定しなければならず、被災地における WiLD Net 構築にあたって、事前調査やネットワーク設計に多くの時間を費やす.

本研究では、事前に取得できる地図情報や現地の方の知識を活用し、機材の設置場所を推定し、地図として表示するシステムを構築した.機能要件は、場所の制約とネットワークの設計を考慮や現地調査やネットワークにかかる時間よりも早く計算できることであった.

評価は計算機上で見通し判定か可能かについて, 実空間上で撮影された写真と計算上で判定された建物を比較することで評価し, システムの実行時間として計算量と計算時間とメモリの消費量を評価した.

見通し計算では、手動でデータベースを更新した後に 52.9%*1の見通しが正確であった.一方、誤差の原因は google earth の建物の 3D データが網羅的でないことや最新でないため誤差が生じる.この点は、データベース更新モジュールを利用やユーザーとのインタラクティブな中継点算出の算出により解決する.計算時間は、データベースのレコード数 487,627、設置数 20 で約 50 秒、設置数 100 で約 240 秒程度でありレコード数 774,877、設置数 20 で約 71 秒、設置数 100 で約 346 秒程度

^{*1} 写真から建物が特定できない場所 36.0% を含む

で計算可能である.この時間において、ユーザーの入力時間を加えても、見通しの確認や場所の制約、ネットワークの設計を考慮し、設置位置を決定することができるため、現地調査やネットワークの設計にかかる時間を削減が期待できる.

本研究は被災地における被災者による被災者のためのネットワークの構築として、WiLD Net を活用して迅速ネットワーク構築に貢献が期待できる.

8.2 今後の課題と展望

8.2.1 機能の追加

より安定したネットワークを構築するために、電波伝搬特性モデルを追加ネットワークの設計時の評価に加える必要がある。また、google earth の 3D の地図情報を利用できる地域が少ないため、他の 3D の地図データの活用も検討する必要がある。データ量の増加や計算機資源に応じたアルゴリズムの設計と利用アルゴリズムの動的変更機構を実装する必要がある。

8.2.2 防災/減災への応用

被災地域において、本システム以外にも被害状況収集や運送、交通な様々なデータの基礎データとして、被災地の状況に即した地理情報が必要となる。地図情報を常時更新し続け、様々なサービスとの連携できるような地図データベースの設計/構築が必要である。ボランティアの手を用いたり、衛星写真 [18] から建物の状況を判断するなど、様々な研究と連携し、データベースの設計から更新の手法まで、被災状況下で活用可能な設計で構築しなければならない。

WiLD Net を被災者の手で自由に構築できれば、一般の方からの被害情報収集や情報発信、二次災害防止のためのモニタリングセンサーのデータを収集するネットワークや、ロボットなどを遠隔で動かすために利用など様々な需要に対応したネットワークの構築に役立つ。実際の被災地で、WiLD Net を含めたネットワーク機器を用い、被災者自らで通信環境を構築するための、モデル作りや機器設計、システム設計、運用体系の確立まで、総合防災の中で通信を生かせる環境の構築が必要である。

参考文献

- [1] Kotaro Kataoka, Keisuke Uehara, and Jun Murai. Lifeline station: A quickly deployable package for post disaster communications. *Internet Conference 2009*, 2009. 1.1, 3.1.1
- [2] Cisco Systems. Cisco ip vsat satellite wan network module for cisco integrated services routers. http://www.cisco.com/en/US/prod/collateral/modules/ps2797/ps6989/product_data_sheet0900aecd804bbf6f.html. 1.1
- [3] Wayan Vota. 7 steps to deploy long-distance wifi in haiti. http://www.inveneo.org/. 1.1, 3.1.2, 5.2, 7.2.2
- [4] 震災復興インターネット事務局. 震災復興インターネット. http://pdrnet.wide.ad.jp/. 1.1, 3.1.1
- [5] K. Chebrolu, B. Raman, and S. Sen. Long-distance 802.11 b links: performance measurements and experience. In *Proceedings of the 12th annual international conference on Mobile computing and networking*, pages 74–85. ACM, 2006. 2.2.3
- [6] R. Patra, S. Nedevschi, S. Surana, A. Sheth, L. Subramanian, and E. Brewer. Wildnet: Design and implementation of high performance wifi based long distance networks. NSDI, 2007. 2.3.1
- [7] 石川 義裕, 岩村 幹生, and 林 貴裕. Foma の無線ネットワーク設計概要 その 2 無線ネットワーク設計とパラメータ最適化の事例. NTT DoCoMo テクニカル・ジャーナル, 12(3):51-55, 2004. 2.3.1
- [8] J. Bicket, D. Aguayo, S. Biswas, and R. Morris. Architecture and evaluation of an unplanned 802.11 b mesh network. In *Proceedings of the 11th annual international conference on Mobile computing and networking*, pages 31–42. ACM, 2005. 2.3.1
- [9] A. Sheth, S. Nedevschi, R. Patra, S. Surana, E. Brewer, and L. Subramanian. Packet loss characterization in wifi-based long distance networks. In *INFOCOM 2007. 26th*

- IEEE International Conference on Computer Communications. IEEE, pages 312–320. Ieee, 2007. 2.3.1
- [10] Inveneo. How to deploy long-distance wifi in haiti. http://www.inveneo.org/haiti-wifinetwork. 3.1.2
- [11] radio mobile. http://www.cplus.org/rmw/english1.html. 3.1.2, 3.2.2
- [12] google. google earth. http://www.google.co.jp/intl/ja/earth/index.html. 3.1.2
- [13] B. Sen, S.andRaman. Long distance wireless mesh network planning:problem formulation and solution. In *Proceedings of the 16th international conference on World Wide* Web, pages 893–902. ACM, 2007. 3.2.1
- [14] アジア航測株式会社. 電波伝搬シミュレーションソフト. http://www.ajiko.co.jp/service/e-municipality/14_electric.html. 3.2.2
- [15] Alex Shovkoplyas. Ham cap. http://www.cplus.org/rmw/english1.html. 3.2.2
- [16] 今井 哲朗. 電波伝搬推定のための遺伝的アルゴリズムを用いたレイトレーシング法. NTT DoCoMo テクニカル・ジャーナル, 15(3):20-26, 2007. 3.2.2
- [17] Carsten Blum MAC DEVELOPMENT. cliclick. http://www.bluem.net/de/mac/cliclick/. 6.1.2
- [18] S. Hinz, F. Kurz, A. Baumgartner, and P. Wasmeier. The role of shadow for 3d object reconstruction from monocular images. In 5th Conference on Optical, pages 354–363, 2001. 7.2.1, 8.2.2
- [19] 株式会社ゼンリン. ゼンリン住宅地図. http://www.zenrin.co.jp/. 7.2.1

謝辞

本卒業論文を執筆するにあたり、多くの人の助力をいただきました。

はじめに、本論文執筆にあたりご指導頂きました慶應義塾大学環境情報学部教授 村井純博士、同学部准教授 楠本博之博士、同学部教授の中村修博士、同学部専任講師 Rodney D. Van Meter 博士、同学部准教授 植原啓介博士、同学部准教授 三次仁博士、同学部教授 武田圭史博士政策メディア研究科特任講師佐藤雅明博士、同研究科特任助教片岡広太郎博士、同研究科講師 Achmad Husni Thamrin 博士に感謝します.

特に環境情報学部准教授植原啓介博士,政策メディア研究科特任講師佐藤雅明博士,同研究科特任助教片岡広太郎博士,同研究科講師 Achmad Husni Thamrin 博士には本論文執筆に辺り多大なご指導を頂きました. 日々の研究活動から宮城県栗原市のプロジェクトまで, 私自身が成長する大きなきっかけを頂き,また見守ってくださりました. ご多忙の中,親身にご指導いただけたことを深謝致します.

慶應義塾大学徳田/村井合同研究室の鈴木詩織氏,波多野敏明氏,澤田暖氏,倉田彩子氏,谷口悠氏,服部将真氏,上野幸杜氏,木本瑞希氏,三條場直希氏,木村翔氏,船柳孝明氏,OB/OGである中村友一氏,片岡友香氏,鵜飼佑氏に感謝します.様々な研究活動の中で皆様から勉強させて頂いたことは一生の財産です.

また、田崎創博士、水谷正慶博士、六田佳祐氏、上原雄貴氏、永山翔太氏、重松邦彦氏、佐藤弘崇氏、 広石達也氏、野島良和氏、中村遼氏、横石雄大氏、佐藤友紀氏、加藤碧氏、吉原洋樹氏、佐藤淳哉氏、鴻 野弘明氏、水谷伊織氏に感謝します。 ふとした会話やアドバイスの中から様々なアイディアが生まれ、様々な考え方を身に付け、世界を広げられることができました.

特に鈴木詩織氏、澤田暖氏、鵜飼佑氏にはお世話になりました. 先輩方の背中を見て、様々なことを吸収、勉強させて頂きました. 楽しいこと、辛いことを様々なことを共有し学んだことは、これからの世界をより良くするための原動力として、生かしていきたいと思います.

また、中村遼氏、横石雄大氏、上野幸杜氏は、WIDE 合宿や Open Rearch Folum, Internet of

Things などのネットワーク構築など様々なイベントでお世話になりました。また一緒に仕事できることを楽しみにしています。

研究のサポートをして頂いた, 荒井成美氏, 遠藤ちひろ氏, 南方真氏に感謝します. 差し入れを頂いたことや論文の構成/英訳/日本語チェックなど研究の手助けをして頂きました.

研究や出張, 論文執筆など様々な活動の辛さを理解してくれ, いつも励ましてくれた福嶋愛美氏に 感謝します.

不規則な生活の中,精神面や健康面で支えてくれた,両親 村上弘毅氏,孝子氏に感謝します. 良き師,先輩,仲間,後輩に出会えたことを心から感謝します.

最後に、2011年3月11日の東日本大震災で犠牲になられた方のご冥福を祈り、また、被災された皆様が平穏無事な生活に戻れるよう祈らさせていただきます。今後の私の研究で、災害において犠牲になる人を減らし、いち早く、平穏な生活に戻れるような社会を構築できるように、研究に邁進したいと思います。

2012年1月村上滋希

付録A

建物 ID 付与のソースコード

データベースに保存されている地図情報 (緯度/経度/建物の高さ/地表の高さ) から建物を検出し建物 ID を付与するためのソースコードを記載する.

ソースコード A.1 建物 ID の付与

```
#緯度方向への走査
mv = 0.00002
for(my $i=$lat_min;$i<=$lat_max;$i=$i+$mv){</pre>
       i = i/100000;
       $where_name = "${table_name}.lat_=_$i";
       select_name = "s{table_name}.lat_, s{table_name}.lng_, s{table_name}.
            height<sub>□</sub>,${table_name}.ground";
       $sql_command = "SELECT_\$select_name_FROM_\$table_name_WHERE_\${\text{where_name}}
            } UORDER BY $\table_name}.lng asc";
       $sth = $db->prepare($sql_command);
       $sth->execute;
       $num_rows = $sth->rows();
       $seq_flag=0;
       for(my $j=0;$j<$num_rows;$j++){</pre>
               @line = $sth->fetchrow_array();
               if(!defined($line[3])){
                       $seq_flag = 0;
                      next;
               }
               $diff = $line[2] - $line[3];
               if ($diff > 1) { #差分が 1以上の場合
```

```
if($seq_flag == 0){
                                                                                                $ID = &updateID(); #新たな重複しないID を生成
                                                                                                $seq_flag = 1;
                                                                        }
                                                                         $update_name = "${table_name}.id_=\subseteq:\subseteq:\subseteq:\subseteq:\subseteq:\subseteq:\subseteq:\subseteq:\subseteq:\subseteq:\subseteq:\subseteq:\subseteq:\subseteq:\subseteq:\subseteq:\subseteq:\subseteq:\subseteq:\subseteq:\subseteq:\subseteq:\subseteq:\subseteq:\subseteq:\subseteq:\subseteq:\subseteq:\subseteq:\subseteq:\subseteq:\subseteq:\subseteq:\subseteq:\subseteq:\subseteq:\subseteq:\subseteq:\subseteq:\subseteq:\subseteq:\subseteq:\subseteq:\subseteq:\subseteq:\subseteq:\subseteq:\subseteq:\subseteq:\subseteq:\subseteq:\subseteq:\subseteq:\subseteq:\subseteq:\subseteq:\subseteq:\subseteq:\subseteq:\subseteq:\subseteq:\subseteq:\subseteq:\subseteq:\subseteq:\subseteq:\subseteq:\subseteq:\subseteq:\subseteq:\subseteq:\subseteq:\subseteq:\subseteq:\subseteq:\subseteq:\subseteq:\subseteq:\subseteq:\subseteq:\subseteq:\subseteq:\subseteq:\subseteq:\subseteq:\subseteq:\subseteq:\subseteq:\subseteq:\subseteq:\subseteq:\subseteq:\subseteq:\subseteq:\subseteq:\subseteq:\subseteq:\subseteq:\subseteq:\subseteq:\subseteq:\subseteq:\subseteq:\subseteq:\subseteq:\subseteq:\subseteq:\subseteq:\subseteq:\subseteq:\subseteq:\subseteq:\subseteq:\subseteq:\subseteq:\subseteq:\subseteq:\subseteq:\subseteq:\subseteq:\subseteq:\subseteq:\subseteq:\subseteq:\subseteq:\subseteq:\subseteq:\subseteq:\subseteq:\subseteq:\subseteq:\subseteq:\subseteq:\subseteq:\subseteq:\subseteq:\subseteq:\subseteq:\subseteq:\subseteq:\subseteq:\subseteq:\subseteq:\subseteq:\subseteq:\subseteq:\subseteq:\subseteq:\subseteq:\subseteq:\subseteq:\subseteq:\subseteq:\subseteq:\subseteq:\subseteq:\subseteq:\subseteq:\subseteq:\subseteq:\subseteq:\subseteq:\subseteq:\subseteq:\subseteq:\subseteq:\subseteq:\subseteq:\subseteq:\subseteq:\subseteq:\subseteq:\subseteq:\subseteq:\subseteq:\subseteq:\subseteq:\subseteq:\subseteq:\subseteq:\subseteq:\subseteq:\subseteq:\subseteq:\subseteq:\subseteq:\subseteq:\subseteq:\subseteq:\subseteq:\subseteq:\subseteq:\subseteq:\subseteq:\subseteq:\subseteq:\subseteq:\subseteq:\subseteq:\subseteq:\subset
                                                                         $where_name = "${table_name}.lat_=_$line[0]_AND_${
                                                                                      table_name\}.lng_{\sqcup}=_{\sqcup}$line[1]";
                                                                         $sql_command = "UPDATE_${table_name}_SET_${update_name}_
                                                                                      WHERE<sub>□</sub>${where_name}";
                                                                         $sth2 = $db->prepare($sql_command);
                                                                         #print OUT "$diff\t:$sql_command\n";
                                                                         $sth2->execute;
                                                                        $sth2->finish;
                                                }else{
                                                                        $seq_flag = 0;
                                                }
                        $sth->finish;
#経度方向への走査
for(my $i=$lng_min;$i<=$lng_max;$i=$i+$mv){</pre>
                        $where_name = "${table_name}.lng_=_$i";
                        $select_name = "${table_name}.lat,_${table_name}.lng,_${table_name}.id"
                        $sql_command = "SELECT_\${select_name}_\FROM_\$table_name_\WHERE_\${
                                      where_name}_\u0RDER\u8Y\u${table_name}.lat\uasc";
                        $sth = $db->prepare($sql_command);
                        $sth->execute;
                        $num_rows = $sth->rows();
                        $seq_flag=0;
                        for(my $j=0;$j<$num_rows;$j++){</pre>
                                                @line = $sth->fetchrow_array();
                                                if(!defined($line[2])){
                                                                        $seq_flag = 0;
                                                                        next;
                                                }
```

```
if($line[2] > 0){#
                      if($seq_flag == 0){
                             $ID = $line[2];
                             $seq_flag = 1;
                      if($line[2] != $ID){
                             $update_name = "${table_name}.id_=_$ID";
                             $where_name = "${table_name}.id_=_$line[2]";
                             $sql_command =
                                            "UPDATE_\${table_name}_\SET_\${
                                                 update_name}_WHERE_\${
                                                 where_name}";
                             $sth2 = $db->prepare($sql_command);
                             $sth2->execute;
                             $sth2->finish;
                      }
              }else{
                      $seq_flag = 0;
              }
       }
       $sth->finish;
$db->disconnect;
```

付録 B

見通し確認のソースコード

2 点間の緯度/経度を与えると 2 点間の見通し可能かを確認する. 引数はそれぞれの地点の緯度/経度である.

ソースコード B.1 見通し確認関数

```
# 予めデータベースを読み込み以下の連想記憶配列に
# 緯度/経度をキーとした高さと建物ID のデータが格納されている.
$height_latlng{$lat}{$lng}; # 高さが格納 キーは緯度/経度
$id_latlng{$lat}{$lng}; # 建物ID が格納 キーは緯度/経度
#使用方法
#&line_of_sight($lat1,$lng1,$lat2,$lng2); # 引数は2地点の緯度経度
#返り値 -1:同一地点の指定 ○:見通しなし 1:見通しあり
sub line_of_sight{
      my \ \mbox{$mv = 0.00002};
      my $lank = 50000;
      my @lat_los = undef;
      my @lng_los = undef;
      my @height_los = undef;
      my @id_los = undef;
      my @ground_los = undef;
      my p = mv/2;#
      my $cmpflag_x;
      my $cmpflag_y;
```

```
my $x_tmp;
     my $y_tmp;
     if($_[1] < $_[3]){ # 入力
             \frac{cmpflag_x = 0}{cmpflag_x}
     }else{
             \frac{cmpflag_x = 1}{cmpflag_x}
     if($_[0] < $_[2]){ # 入力
             $cmpflag_y = 0;
     }else{
             $cmpflag_y = 1;
     }
     my $x;
     my $y;
     my $x1 = $_[1];
my \$y1 = \$_[0];
my $x2 = $_[3];
my $y2 = $_[2];
     my  dx = (x1 - x2);
     my  $dy = ($y1 - $y2);
     my $num_array=0;
     # 2点間の直線上の高さのデータを取り出す
     if($dx == 0 && $dy == 0){ #同一地点が指定された場合
            return -1;
     elsif(dx == 0){
                           # x 軸に平行な直線の場合
             if($cmpflag_y == 0){
                    for(my $y=$y1;$y<=$y2;$y=$y+$mv){</pre>
                           x_t = int(x1*slank)/slank;
                           $y_tmp = int($y*$lank)/$lank;
                           if(!defined($height_latlng{$y_tmp}{$x_tmp})){
                                   last;
                           }
                           if(!defined($id_latlng{$y_tmp}{$x_tmp})){
                                   last;
                           }
```

```
$lat_los[$num_array] = $y_tmp;
                     $lng_los[$num_array] = $x_tmp;
                     $height_los[$num_array] = $height_latlng{$y}{$x1}
                     $id_los[$num_array] = $id_latlng{$y}{$x1};
                     $num_array++;
              }
      }else{
              for(my $y=$y1;$y>=$y2;$y=$y-$mv){
                     x_t = int(x1*slank)/slank;
                     $y_tmp = int($y*$lank)/$lank;
                     if(!defined($height_latlng{$y_tmp}{$x_tmp})){
                            last;
                     if(!defined($id_latlng{$y_tmp}{$x_tmp})){
                            last;
                     }
                     $lat_los[$num_array] = $y_tmp;
                     $lng_los[$num_array] = $x_tmp;
                     $height_los[$num_array] = $height_latlng{$y}{$x1}
                     $id_los[$num_array] = $id_latlng{$y}{$x1};
                     $num_array++;
              }
      }
}else{ #傾きを定義出来る場合
       $a = $dy/$dx; #傾きを求める
       if(-1 < $a && $a < 1){ # x 軸とのなす角が 45 度以下の場合
              if($cmpflag_x == 0){
                     for(my $x=$x1;$x<=$x2;$x=$x+$mv){</pre>
                            #my $y = $a * ($x - $x1) + $y1;
                            y = a * (x - x1) + y1;
                            x_t = int(x*\t)/\t
                            $y_tmp = int($y*$lank)/$lank;
                            if(!defined($height_latlng{$y_tmp}{$x_tmp})
                                })){
                                   last;
```

```
if(!defined($id_latlng{$y_tmp}{$x_tmp})){
                            last;
                     }
                     $lat_los[$num_array] = $y_tmp;
                     $lng_los[$num_array] = $x_tmp;
                     $height_los[$num_array] = $height_latlng{
                         $y_tmp}{$x_tmp};
                     $id_los[$num_array] = $id_latlng{$y_tmp}{
                         $x_tmp};
                     $num_array++;
              }
       }else{
              for(my $x=$x1;$x>=$x2;$x=$x-$mv){
                     #my $y = $a * ($x - $x1) + $y1;
                     y = a * (x - x1) + y1;
                     x_t = int(x*\t)/\t
                     $y_tmp = int($y*$lank)/$lank;
                     if(!defined($height_latlng{$y_tmp}{$x_tmp})
                         })){
                            last;
                     if(!defined($id_latlng{$y_tmp}{$x_tmp})){
                            last;
                     }
                     $lat_los[$num_array] = $y_tmp;
                     $lng_los[$num_array] = $x_tmp;
                     $height_los[$num_array] = $height_latlng{
                         $y_tmp}{$x_tmp};
                     $id_los[$num_array] = $id_latlng{$y_tmp}{
                         $x_tmp};
                     $num_array++;
              }
       }
}else{ #x 軸とのなす角が 45 度を超えた場合
       if($cmpflag_y == 0){
```

```
for(my $y=$y1;$y<=$y2;$y=$y+$mv){</pre>
              my $x = (y - y1)/a + x1;
              x = (y - y1)/a + x1;
              x_t = int(x*\theta)/\theta
              $y_tmp = int($y*$lank)/$lank;
              if(!defined($height_latlng{$y_tmp}{$x_tmp})
                  })){
                     last;
              }
              if(!defined($id_latlng{$y_tmp}{$x_tmp})){
                     last;
              }
              $lat_los[$num_array] = $y_tmp;
              $lng_los[$num_array] = $x_tmp;
              $height_los[$num_array] = $height_latlng{
                  $y_tmp}{$x_tmp};
              $id_los[$num_array] = $id_latlng{$y_tmp}{
                  $x_tmp};
              $num_array++;
       }
}else{
       for(my $y=$y1;$y>=$y2;$y=$y-$mv){
              my \ x = (y - y1)/a + x1;
              x = (y - y1)/a + x1;
              $x_tmp = int($x*$lank)/$lank;
              $y_tmp = int($y*$lank)/$lank;
              if(!defined($height_latlng{$y_tmp}{$x_tmp})
                  })){
                     last;
              if(!defined($id_latlng{$y_tmp}{$x_tmp})){
                     last;
              }
              $lat_los[$num_array] = $y_tmp;
              $lng_los[$num_array] = $x_tmp;
              $height_los[$num_array] = $height_latlng{
                  $y_tmp}{$x_tmp};
```

```
$id_los[$num_array] = $id_latlng{$y_tmp}{
                                    $x_tmp};
                                $num_array++;
                          }
                   }
             }
      }
      # 直線上に収集されたデータから見通しを確認する
      $b = $height_los[0];
      my $tail = $num_array - 1;
      if(!defined($height_los[$num_array-1])){
             return 0;
      }
      # 1次関数を用い、2点間に見通し線を超える高さがないか確認
      a = (\frac{1}{2} - \frac{1}{2} - \frac{1}{2} - \frac{1}{2} - \frac{1}{2} 
      for(my $i=1;$i<$num_array-1;$i++){</pre>
             if($id_los[0] == $id_los[$i]){ # 同一の建物は見通しの阻害点に含めな
                   next;
             }
             my $diff = $height_los[$i] - (($a * $i) + $b);
             if ($diff > 0){ # diff が 0 を越えると見通し線を超える
                   return 0; # 見通しが取れないため 0を返す
             }
      }
      return 1;
                   # 見通しがとれるため 1を返す
}
```