Wireless Internet ワーキンググループ 2013 年度活動報告

大江 将史　　島 慶一　　Wireless Internet ワーキンググループ

1　はじめに

Wireless Internet ワーキンググループは、無線通信ネットワークを前提とした、より堅牢で柔軟なインターネット構成技術を研究している。2013 年度は、事前プランニングと可視化技術を活用し、現場での細かい調整を削減する無線メッシュネットワークの構築運用技術の実証実験、またバックホールチャネルを複数利用することで多重化による性能向上を目指した運用実証実験を行った。また、WiFi デバイスにおけるネットワークへの接続性を高める WiFi Passpoint 技術の調査を実施した。

2　WIDE プロジェクト 2013 年春合宿におけるシミュレーションと可視化による事前検証を利用した OLSR ネットワーク構築実験

2.1　背景

無線アドホックネットワーク技術は、会議ネットワークやイベントネットワークなどの一時的なネットワーク環境を構築するために有用と考えられてきた。さらに 2011 年の東日本大震災以降においては、情報通信ネットワークによる正確かつ迅速な情報提供および共有が被災者の生命に重大な影響を及ぼすことが理解され [1]、被災環境でのいち早い情報基盤提供のための手段としても注目されている。しかしながら、実用に耐える無線アドホックネットワークを構築するためには様々な設定パラメータおよび環境要因を考慮する必要があり、無線ネットワークの知識やアドホックネットワークプロトコルの動作に精通した技術者なしでは困難な状況である。本実験では、構築現場での設定や調整事項をできる限り省き、無線アドホックネットワークノードを「置くだけ」で実用的なネットワークを構築する技術の実現に挑戦した。具体的には構築現場の環境を事前にシミュレーション環境で再現し、無線アドホックネットワークノードの配置や通信パラメータをシミュレータの中で事前決定する。これにより、現場でのノード配置場所や設定パラメータの微調整を不要とし、誰もが簡単に効率的な無線アドホックネットワークを構築できる仕組みの実現を目指している。

2.2　実施手順

本実験では、WIDE プロジェクトの 2013 年春合宿研究会の会場を実験対象とし、研究会参加者向けに OLSR[2] を用いた無線アドホックネットワークを提供した。このために必要となった手順は以下の通りである。

1. 会場レイアウトの確認と無線環境の計測
2. 会場環境のシミュレータ内での再現
3. シミュレータ用 OLSR ノードの構築と設定
4. 運用モニタリングのための可視化ツールの構築
5. シミュレーション環境での運用テスト
6. 現場環境用 OLSR ノードの構築とシミュレータ用ノードからの設定移行
7. 現地におけるネットワーク構築および調整

以後、各章にて作業の詳細を述べる。

2.3　会場レイアウトの確認と無線環境の計測

会場をシミュレータ内部で再現するにあたり、部屋の配置、壁の位置など無線通信環境に影響する情報を事前に収集しておく必要がある。さらに、建物内部の物品の密集度や建材の種類などによって無線の減衰レ

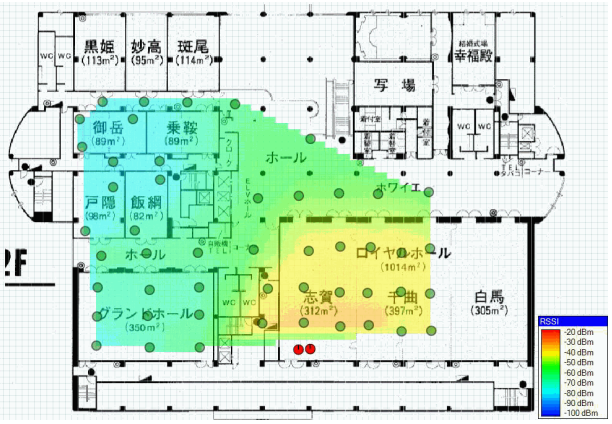

図 1: 現地における無線信号強度の事前計測

ベルが変化するため、シミュレータ環境の減衰率を設定するための基礎計測が必要となる。

部屋の配置や壁の位置は事前に図面などから把握できるが、無線環境は現地での測定が必要となる。今回は会場下見の際に利用予定の OLSR ノードを持ち込み、無線信号強度を計測した。ただし、現実的には常に会場の環境を事前測定できるわけではないため、現地の建物情報から確度の高い無線減衰率を導きだす手法を確立する必要がある。

図 1 に現地計測の結果を示す。赤丸で示した点が計測起点となる OLSR ノードを置いた場所である。アンテナ出力は 18dbm に設定し、緑丸で示した地点で計測した後、計測ソフトウェアによって信号強度を計算し描画している。図から、距離による無線信号の減衰の度合い、また壁による影響が判断できる。この情報を元に、シミュレータ環境での会場環境を調整する。

2.4 会場環境のシミュレータ内での再現

無線環境のシミュレーションには情報通信研究機構北陸 StarBED 技術センター [1] で開発されている QOMET[3][2] を用いた。

QOMET では、障害物の定義、空間環境の定義、無線ノードの定義、および無線ノードの移動軌跡の定義を元に、時系列に無線ノード間の通信状態をシミュレートする。今回、無線ノードは移動しないメッシュルータ、およびアクセスポイントとして利用しているため、無線ノードの移動情報は利用していない。

[1] http://starbed.nict.go.jp

[2] https://www.starbed.org/qoala/twiki/bin/view/QOALA/QOMET

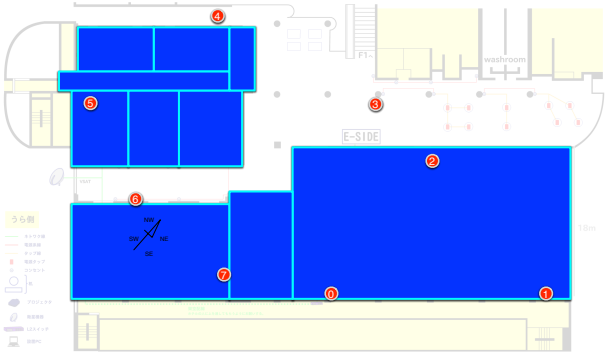

図 2: シミュレータ内部での部屋の定義と OLSR ノードの配置

表 1: 会場環境のパラメータ

場所	α	W	σ
ホワイエ、ホール	2	0	3
部屋	2	4.8	3
サービス用廊下など	3	0	7

事前に入手したフロアの見取り図から、図 2 に示す形で部屋空間を定義した。図 1 に示されている 6 つの部屋（ロイヤルホール、グランドホール、戸隠、飯綱、御岳、乗鞍）に加え、ホテルサービス用の廊下、トイレなどを壁のある閉じた空間として定義した。ホワイエおよびホールは開放空間として定義した。

QOMET では、無線信号の減衰を log-distance path loss モデル [4] で実現する。

$$Pr(d) = Pr_0 - 10\alpha log(d) - W + X_\sigma \qquad (1)$$

$Pr(d)$ は信号源から距離 d だけ離れた位置での信号受信強度、Pr_0 は信号源の信号強度である。α、W、および X_σ はそれぞれ path loss exponent、wall attenuation、標準偏差 σ におけるガウシアンランダム変数であり、環境に依存したパラメータとなる。これらの値を適切に設定することにより、現地の環境に近似した環境を構築する。今回は、第 2.3 章での実測値を参考に、表 1 に示す数値を採用した。

現在のところ、QOMET には実測された信号強度情報から、空間定義パラメータを逆算する手法が提供されていない。そのため、実測値に近い状況を再現するパ

ラメータを調べるためには、パラメータを微調整しながら随時結果を確認していく必要がある。この作業はシミュレーション結果を正確にするためには重要であるにもかかわらず、正しいパラメータを導きだすのは容易ではない。実際、今回採用したパラメータでは、無線ノードの配置戦略に関しては十分有効なシミュレーション環境として機能したものの、通信帯域のシミュレーション環境としては実際とは大きく異なる結果となっている。

図 2 にはシミュレータ内での OLSR ノードの場所も示している。各ノードは 802.11a を用いて通信する設定となっており、送信出力は 18dbm に設定してある。

本シミュレーションに用いた QOMET のシナリオファイルを付録 A に添付する。

2.5 シミュレータ用 OLSR ノードの構築と設定

シミュレーションは KVM[5] を用いた仮想環境内で実施した。実ノードとしてバッファロー社の WZR-HP-AG300 に OpenWRT を搭載した機材を想定していたため、仮想環境内でも OpenWRT(x86) によるノードを利用している。

QOMET 実装上の制限から、シミュレーションに用いるノードは Linux カーネル 2.6 ベースである必要がある。OpenWRT は最新版ですでに Linux カーネル 3 に移行してしまっているため、北陸 StarBED 技術センターで利用されている Linux カーネル 2.6 ベースの OpenWRT(x86) を用いている。

OLSR プロトコルを実装していない会議参加者のノードを収容するために、各 OLSR ノードはバックホールネットワークを構成する無線インターフェースと、会議参加者を収容する無線インターフェースの二つのインターフェースを持つ。バックホールに 802.11an、参加者収容に 802.11gn を用い、無線の干渉を抑えている。本来であれば、バックホール側に到達性に優れた 802.11gn を用いた方がより安定したネットワークを構築できると思われるが、近年増加しているスマートフォンの多くが 802.11an に対応していないため、今回のバンド選択になっている。

なお、シミュレータでは会議参加者のネットワークのシミュレーションは実施しておらず、バックホールネットワークのみを取り扱っている。

表 2: ネットワークアドレス割当

ネットワーク	アドレス空間
参加者収容	$10.0.n.1/24$ ($n : OLSR$ ノード番号)
バックホール	$10.0.255.x/24$ ($x : 100+OLSR$ ノード番号)

各 OLSR ノードは、自ノードの番号に応じてサブネットワークを持ち、全 OLSR ノードで共有しているサブネットワークで経路情報の交換およびパケットの転送を行う設定となっている。なお、ノード 0 はインターネットゲートウェイとして設定されており、このノードを通じて会議会場の他のネットワークファシリティおよび外部ネットワークに接続される。表 2 に各ノードが利用したアドレス空間を示す。

付録 B に今回用いた OpenWRT olsrd 設定ファイルを添付する。なお、インターネットゲートウェイとして動作していた OLSR ノード 0 は自ノードの HNA (Host and Network Association) 情報 (10.0.0.0/24) に加え、デフォルト経路 (0.0.0.0/0) も HNA として設定されている。

2.6 運用モニタリングのための可視化ツールの構築

OLSR を用いたネットワークが予期した通りに動作しているか、また安定して動作しているかを把握するのは容易ではない。ノード間の接続が無線であるため、近傍ノードが固定しておらず、環境的な要因によって動的にトポロジが変化するためである。そこで、今回はネットワークの現状を把握しやすくするための可視化サポートを試みている。

OLSR によって構成された経路情報は各ノードで olsrd プロセスに問い合わせることができる。この情報をノード内で加工し、OLSR ネットワークの外に置かれた可視化サーバに定期的に送信する仕組みを構築した。また、経路情報に加えて、各ノードに接続している会議参加者端末の数（ただし、シミュレーション環境には会議参加者ノードが存在しないので常に 0)、バックホールネットワークから入出力されるトラフィック量も同時に送信している。図 3 にシミュレータ環境を可視化した図を示す。

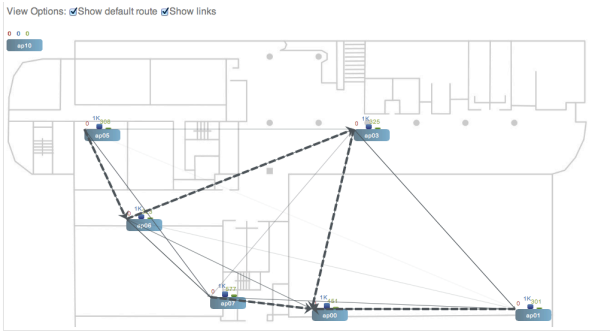

図 3: シミュレータ環境の可視化

図中、点線の矢印は各ノードからインターネット側へ抜ける経路（デフォルト経路）を示し、実線の矢印はノードの近隣関係を示している。実線の濃淡は、近隣ノードのリンク品質を示しており、色が濃いほど高品質のリンクであることを示している。各 OLSR ノードの上部に表示されている数値および棒グラフは、左からそれぞれ、そのノードに接続している会議参加者のノード数、バックホールからノードに入力されているトラフィック、ノードからバックホールに出力されているトラフィックを示す。

2.7 シミュレーション環境での運用テスト

可視化によってネットワークトポロジの構成が把握できたことにより、当初予定していたノード配置では非効率的になると考えられる点が明らかになっている。図 2 に示した通り、当初は 8 台の OLSR ノードを運用する予定としていた。しかしながら、ノード 2 とノード 4 が存在する状況下では、ノード 4 からノード 0（インターネットゲートウェイ）に向かう経路がノード 4-2-0 となり、ノード 6 からノード 0 に向かう経路がノード 6-3-0 となることが予想された。この場合ホワイエおよびロイヤルホールの空間を異なるパスが複数横断することになり、無線通信の干渉が増加すると考えられた。よって、シミュレーションからノード 2 と 4 を省き、重複するパスを減らしている。ノードを削減した結果が、前章に示した図 3 となる。

表 3: OpenWRT パッケージ一覧

OLSR 経路制御	olsrd
	olsrd-mod-arprefresh
	olsrd-mod-dyn-gw
	olsrd-mod-httpinfo
	olsrd-mod-nameservice
	olsrd-mod-txtinfo
SNMP サービス	snmpd
経路情報取得	ip
ログ記録用ストレージサポート	kmod-usb-storage
	block-mount
	kmod-fs-ext4
スクリプト記述	bash
	python
	python-json
トラフィック制御	tc
	iptables-mod-ipopt
	kmod-sched
性能計測	iperf
デバッグ	tcpdump

2.8 現場環境用 OLSR ノードの構築とシミュレータ用ノードからの設定移行

現場環境にはバッファロー社の WZR-HP-AG300 を用いた。利用した OpenWRT のバージョンは 12.09-beta である [3]。実験ノード構築のために追加で利用したパッケージを表 3 に示す。なお、表に示したパッケージはすべてトップレベルのパッケージであり、依存関係によりサブパッケージもインストールされている。

経路制御に関してはシミュレータで利用した設定を継続して利用した。シミュレータでの設定から変更した内容は以下の通りである。

- ネットワークインターフェース構成
  無線 LAN インターフェースを利用する設定に変更した。また、ノード 0 に関しては上流ネットワークに接続するために WZR-HP-AG300 に備えられている 4 ポートスイッチのひとつをポート VLAN

[3] http://downloads.openwrt.org/attitude_adjustment/12.09-beta/ar71xx/generic/

を用いて別ネットワークに設定した。ネットワーク設定ファイルは付録 D を参照。

- firewall 構成
  OpenWRT の標準設定では、lan インターフェースと wan インターフェースの間で NAT を用いたパケット転送しか許可していないため、バックホールインターフェースと利用者収容インターフェースの間でのパケット転送を許可する設定を行った。付録 E を参照。

- dhcp 構成
  利用者を収容するインターフェースで DHCP サービスを起動した。付録 F を参照。

- fstab 構成
  WZR-HP-AG300 の USB ポートに差した USB メモリをログ記録用として利用するため、自動マウントの設定を行った。付録 G を参照。

- hotplug2 構成
  WZR-HP-AG300 の Movie Engine スイッチを shutdown スイッチとして利用するため、hotplug2 を構成した。付録 H を参照。

2.9 現地におけるネットワーク構築および調整

現地で無線ノードを設置し、再度動作状況の確認を実施した結果、ボードメンバーの打ち合わせ場所として利用される「乗鞍」(図 1 の地図参照) に無線ノードを追加することとした。それに応じて、プレナリ部屋近傍に設置していた 2 台の位置を微調整している。図 4

可視化に関しては、負荷の高くなった無線アクセスポイントから、自律的に負荷の低いアクセスポイントへ移動してもらうことを期待して、アソシエーション数を数に応じた大きさで表示する工夫を追加している。

2.10 メッシュネットワークの性能

OLSR メッシュネットワーク網と有線ネットワーク境界で計測された帯域は最大で 5Mbps 程度となっていた。最も長いパスとなる、インターネットゲートウェイ (OLSR ノード番号 0) から、乗鞍のアクセスポイント (OLSR ノード番号 5) の帯域が 5Mbps 程度であったことから、合宿期間中はメッシュクライアントとバックホール間の帯域を 5Mbps にシェープしていたが、比較のために解除して運用した場合でも 5Mbps を大きく超えることはなかった。今後は、この帯域をいかに拡大するかが重要な課題となる。

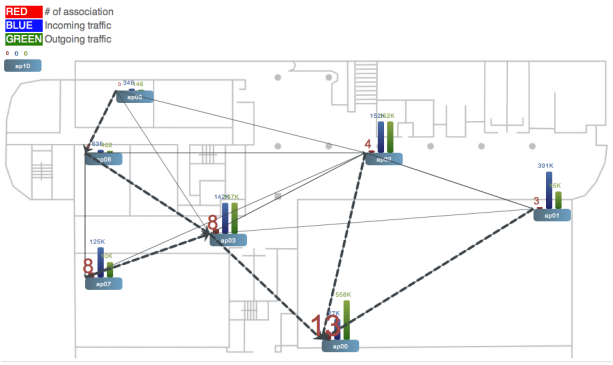

図 4: 最終的な無線ノード配置と可視化画面

2.11 まとめ

今回の実験では、無線メッシュネットワークを実環境で運用する際の障壁となっている準備作業の負荷を低減することと、運用状況の把握手段の提供を目的とした。実環境を模したエミュレーション環境を構築し、無線ノードを配置してシミュレーションを実施することで、現地での配置調整の時間を削減し、迅速な導入を目指した。今回、当初前回 (2012 年 3 月) と同様の配置でシミュレーションした結果、不要と考えられるアクセスポイントをシミュレータで事前に発見することができ、現地での構成の補助となった。また、シミュレータ上でのメッシュネットワークの状況を把握するための可視化ツールを作成し、ネットワーク環境を実時間で理解できる環境を構築した。この可視化環境は、シミュレータでも実環境でも共通して利用できるため、シミュレーション環境での経験をそのまま実運用に結びつけることができた。

現在のところ、シミュレータの結果から想定した配置が、他の配置よりも実環境で優れていたかどうかを客観的に判定できていない。実環境での比較はネットワークの構成に時間がかかることから、実際の会議に利用しているネットワークで異なる構成を運用するこ

とが困難であるためである。また、仮に構成変更ができたとして、メッシュネットワークの客観的な性能を示す指標が存在しないため、優劣の判定が困難であることも理由である。

シミュレータ環境に用いた QOMET は、IP フィルタなどの機能を用いて L2 環境を擬似的に L3 で実現する仕組みだが、実環境に近いパラメータを設定するためには数種類のパラメータを微調整して試行を繰り返す必要がある。それでもなお、今回シミュレータ内に構築した環境は特にスループットの再現性において実環境と大きく異なっていた。現場の無線環境の測位結果から、現実的なパラメータを導きだす手法の確立が重要な課題であると思われる。

今後、メッシュ環境の性能評価のための指標を提案するとともに、より効率的なネットワーク環境を、より負担の少ない作業量で構築するための手法の提案と実証を継続していく。

3 WIDE プロジェクト 2013 年秋合宿における OLSR ネットワーク構築実験

前章の実験からの継続実験として、9 月に実施された WIDE プロジェクト 2013 年秋合宿で OLSR による無線メッシュネットワークの運用実験を実施した。

前回からの変更点は以下の通りとなる。

- 運用モニタリングのための可視化ツールの改良
- アクセスポイント数削減による干渉の軽減
- バックホールチャネルの多重化運用
- アクセスポイントの通信パラメータの改善

3.1 可視化ツールの改良

春合宿用に開発した可視化ツールを元に、今回の実験に合わせて可視化ツールの改良を実施した。主な変更点は以下の通りである。

- 2013 年秋合宿のアクセスポイント配置への追随
- バックホールチャネル多重化トポロジへの対応

図 5 に可視化ツールのスクリーンショットを示す。

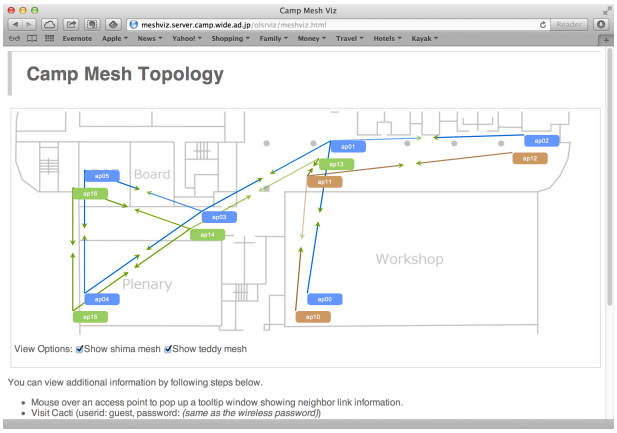

図 5: 2013 年秋合宿用可視化ツール

3.2 アクセスポイント配置

図 5 に示すように、今回の運用ではアクセスポイントの配置場所を 6 箇所に減らしている (2013 年春合宿では 7 箇所 (図 4))。2013 年春で利用していた会場上部の部屋「御岳」(図 1) が今回利用されなかったため、上部へネットワークを延伸する必要がなかったためである。

アクセスポイント数が減ることにより、お互いのバックホールチャネルでの競合が軽減されることが期待されるが、今回はバックホールに複数チャネルを活用する実験を並列して実施したため、その効果は実証できていない。

3.3 バックホールチャネル多重化

今回の運用では、バックホールチャネルを複数利用することで、全体の帯域の向上を目指した。図 5 で色分けされているアクセスポイントおよびリンクがバックホールチャネルに相当する。以下の 3 つのチャネルが設定された。

- 802.11a 36HT+ (図 5 中の茶色)
- 802.11a 48HT+ (図 5 中の緑色)
- 802.11a 149HT+ (図 5 中の青色)

全体をカバーする青色のトポロジーは、前回の春合宿と同じ運用となる。今回は、このネットワークに重ねて、別のメッシュネットワークが運用された。さら

に、追加されたネットワークは会場の左右でも異なるチャネルを用いて、さらなる干渉低減を目指した。

しかしながら、実際の運用では目立った効果がみられなかった。これは、チャネルの異なる複数のアクセスポイントを近接して配置したため、使用チャネルに関わらず互いに干渉が発生していたためであることが、後日の追加実験により確認された。今後、マルチチャネルで運用する場合の参考としたい。

3.4 アクセスポイントの通信パラメータ

今回の運用では、干渉低減と帯域の維持を目指して、以下のパラメータ設定を実施した。

- 低速クライアントの排除
- バックホールチャネルの MCS 下限設定

802.11b では通信帯域の上限が 11Mbps となっている。802.11agn が普及している現状、11Mbps では同じデータ量の通信に長い時間がかかってしまい、他の通信を抑制してしまう原因になる。今回の運用では、アクセスポイントからのビーコン信号を 18Mbps で送信し、かつビーコンに含まれるサポート帯域を 18Mbps 以上に限定することで、暗黙の低速クライアント排除を実施した。クライアントが明示的に SSID を指定して接続すれば、18Mbps より低い帯域でも通信が出来るため、厳密な排除とならないが、意図せず低い帯域で接続してしまう問題には対応できる。

この設定は、ビーコン信号が占有してしまう無線通信時間の削減にも貢献している。通常ビーコンは 1Mbps で 100ms 毎に送信されるが、18Mbps で送信することでビーコン送信にかかる時間が理論上 18 分の 1 に短縮され、その結果他の通信が利用できる時間が増えることになる。ビーコンの送信帯域の効果は、別途運用していた無線モニターツールによって確認された。

また、帯域の下限設定はバックホールチャネルを構成するアクセスポイント間でも実施した。バックホールは 802.11an を用いて構成されているが、プロトコルの仕様上、通常は無線環境が悪化すると低速の通信に切り替わり、接続を維持するようになっている。今回、通信帯域のプロファイルとして利用される MCS (Modulation and Coding Scheme) インデックスを明示的に指定することで、低速でのリンク確立を排除した。この場合、リンクが確立できないとアクセスポイント間の通信が途絶することになるが、メッシュ構成で運用しているため、自動的に別経路での通信に切り替わることを想定した設定となっている。

3.5 メッシュネットワークの性能

性能向上のための調整を実施してみたにもかかわらず、前回の春合宿と比較して、今回ネットワークの性能が大きく向上したとは言えない。インターネットゲートウェイ (ap00) と有線ネットワーク間で計測した最大帯域は 9Mbps 程度が観測されたものの、最大パス長となる ap00 と ap05 間の帯域は 5Mbps 程度にとどまっており、マルチホップ構成による性能低下が依然みられる。

今回、バックホールチャネルの多重化を実施して、性能向上を目指したが、前述の通り、干渉を軽減するためのアクセスポイント配置の知見が不足していたため、十分にその効果を発揮できなかったというのも理由のひとつと思われる。今後、バックホールチャネルの多重化の効果が発揮されるような構成での再実験を予定している。

3.6 ソフトウェア入手先

本章および前章で報告した実験に用いたソフトウェアおよびその設定ファイルに関しては、WIDE プロジェクトメンバー限定で公開している。興味のある方は shima@wide.ad.jp まで連絡してほしい。

4 WiFi Passpoint の有用性調査

4.1 Passpoint とは?

WiFi Passpoint(以下、Passpoint) は、WiFi デバイスにおける WiFi ネットワークへの接続性を高める規格で、Wi-Fi アライアンスによる標準化と認証が進められている [4]。現在のホットスポットサービスにおいて、利用者は、多数の SSID の中からサブスクリプションを有するサービスの SSID を選択し、暗号化・認証の設定、場合によっては、WEB ブラウザの認証など、

[4] http://www.wi-fi.org/discover-and-learn/wi-fi-certified-passpoint™

様々な作業が必要となっている。一部のサービスでは、専用のサプリカントにより独自にその煩雑さを軽減するなどの対応をとっている。

この課題に対して、Passpoint では、ホットスポットサービスをレルム（認証ポリシーやサービス状況など定めたもの）として取扱い、ビーコンやプローブを用いレルムごとの認証方式やサービス状況などの報知と取得を容易にしている。これは、デバイスが、アクセスポイントから提供されるレルム情報とデバイスの有するサブスクリプションなどに基づくプロファイルより、サービスの選択を実現する。たとえば、携帯キャリアが出荷するする端末に、ホットスポットサービスの情報を定義したプロファイルを含めておけば、窓口での煩雑な手続きなしにホットスポットサービスが利用でき、ハンドオーバーも容易である。また、認証や暗号化は、すでに標準化された IEEE802.1x や EAP-SIM/AKA を認証プロトコルなどを利用することにより、実装の容易性も高くなっている。

このように、Passpoint は、現在の Wi-Fi との互換性を有しつつ、そのうえで高度化サービスを提供することで、現在のホットスポットサービスにおける問題を解決することを狙った規格である。

4.2 Passpoint の現状と課題

Passpoint の実装は、端末および、アクセスポイントについて、共に限られた範囲での実装となっている。Android OS の場合、サムソン社の Galaxy シリーズの標準ファームウェアにて対応しており、アクセスポイント設備側は、Cisco の WLC シリーズであれば、利用可能となっている。Galaxy の場合は、`cred.conf` ファイルに、レルムごとのプライオリティや認証方式、証明書、パスワードなどを記載することにより、Passpoint が利用可能となる。

我々は、国内 3 か所で統合運用されている国立天文台ワイヤレスネットワーク上に、構成を行い、機能性の検証やローミングの検証などを継続中である。現状、対応端末が少ない点や、対応ファームウェアが公開されていないといった状況から、実用的なホットスポットサービスを構成するのは難しい状況である。

一方で、本技術は、ホットスポットサービスを利用する側や提供する側も含めて、デバイスにおける煩雑なオペレーション・コストを大幅に削減できる。これは、日常のホットスポット運用もさることながら、緊急時のホットスポット解放や仮設ホットスポット開設といった状況でも、利用情報の告知や設定コストをかけることなく、ホットスポットサービスを利用してもらうことが可能である。

有用な技術であると考えていることから、継続して、機能検証や標準化動向の調査を行う予定である。

5 まとめ

2013 年度は合宿研究会での運用実験を中心に、無線メッシュネットワークの性能向上を目指した実証実験を中心に活動した。これまでの運用経験から、100 名程度の参加者を収容するメッシュネットワークの運用は、家庭用のブロードバンドルータを活用して安価に構築可能であることがわかってきた。ただし、メッシュネットワーク特有の、マルチホップによる性能低下は大きな問題として残っており、これを以下に軽減するかが大規模運用に欠かせないことも実証された。

また、WiFi デバイスの接続性を向上する技術規格である WiFi Passpoint を調査し、その有効性を検討した。将来性が見込める技術として調査を継続する。

今後、より高性能かつ自律運用可能なメッシュネットワークの構成技術を目指して研究をすすめていく。

参考文献

[1] 植原啓介, 大江将史. 震災復興インターネット. 情報処理学会論文誌, Vol. 52, No. 9, pp. 1068–1069, September 2011.

[2] Thomas Heide Clausen and Philippe Jacquet. *Optimized Link State Routing Protocol (OLSR)*. IETF, October 2003. RFC3626.

[3] Razvan Beuran, Lan Tien Nguyen, Khin Thida Latt Junya Nakata, and Yoichi Shinoda. QOMET: A Versatile WLAN Emulator. In *21st International Conference on Advanced Information Networking and Applications (AINA'07)*, pp. 348–353, 2007.

[4] Theodore S Rappaport. *Wireless communications: principles and practice*. IEEE press, 1996.

[5] R.A. Harper, A.N. Aliguori, and M.D. Day. KVM: The Linux Virtual Machine Monitor. In *Proceedings of the Linux Symposium*, pp. 225–230, 2007.

A QOMET シナリオファイル

```
<qomet_scenario start_time="0" duration="300" step="0.5">

<node name="ap00" x="51" y="45" z="1">
  <interface name="interface0" Pt="18"
        ip_address="10.0.255.100" />
</node>
<node name="ap01" x="85" y="45" z="1">
  <interface name="interface0" Pt="18"
        ip_address="10.0.255.101" />
</node>
<node name="ap02" x="67" y="24" z="1">
  <interface name="interface0" Pt="18"
        ip_address="10.0.255.102" />
</node>
<node name="ap03" x="58" y="15" z="1">
  <interface name="interface0" Pt="18"
        ip_address="10.0.255.103" />
</node>
<node name="ap04" x="33" y="1" z="1">
  <interface name="interface0" Pt="18"
        ip_address="10.0.255.104" />
</node>
<node name="ap05" x="13" y="15" z="1">
  <interface name="interface0" Pt="18"
        ip_address="10.0.255.105" />
</node>
<node name="ap06" x="20" y="30" z="1">
  <interface name="interface0" Pt="18"
        ip_address="10.0.255.106" />
</node>
<node name="ap07" x="34" y="43" z="1">
  <interface name="interface0" Pt="18"
        ip_address="10.0.255.107" />
</node>

<environment name="hotel" alpha="2" sigma="3" W="0"
        noise_power="-100" />
<environment name="room" alpha="2" sigma="3" W="4.8"
        noise_power="-100" />
<environment name="backyard" alpha="3" sigma="7" W="0"
        noise_power="-100" />

<environment name="ap00_env" is_dynamic="true" />
<environment name="ap01_env" is_dynamic="true" />
<environment name="ap02_env" is_dynamic="true" />
<environment name="ap03_env" is_dynamic="true" />
<environment name="ap04_env" is_dynamic="true" />
<environment name="ap05_env" is_dynamic="true" />
<environment name="ap06_env" is_dynamic="true" />
<environment name="ap07_env" is_dynamic="true" />

<!-- Hotel -->
<object name="HOTEL" type="building" environment="hotel"
        x1="0" y1="0" x2="100" y2="55" height="4" />

<!-- WS room -->
```

```
<object name="WS" type="building" environment="room"
        x1="46" y1="23" x2="90" y2="47" height="4" />

<!-- plenary -->
<object name="plenary" type="building" environment="room"
        x1="11" y1="32" x2="36" y2="47" height="4" />

<!-- BoF1 -->
<object name="bof1" type="building" environment="room"
        x1="11" y1="14" x2="20" y2="26" height="4" />

<!-- BoF2 -->
<object name="bof2" type="building" environment="room"
        x1="20" y1="14" x2="28" y2="26" height="4" />

<!-- BoF3 -->
<object name="bof3" type="building" environment="room"
        x1="24" y1="4" x2="36" y2="11" height="4" />

<!-- Board -->
<object name="board" type="building" environment="room"
        x1="12" y1="4" x2="24" y2="11" height="4" />

<!-- WR and EV -->
<object name="WR-EV" type="building" environment="backyard"
        x1="36" y1="30" x2="46" y2="47" height="4" />

<!-- EV -->
<object name="EV" type="building" environment="backyard"
        x1="28" y1="14" x2="38" y2="26" height="4" />

<!-- Backyard -->
<object name="backyard-board-bof1" type="building"
        environment="backyard"
        x1="9" y1="11" x2="36" y2="14" height="4" />
<object name="backyard-top-of-EV" type="building"
        environment="backyard"
        x1="36" y1="4" x2="40" y2="14" height="4" />

<connection from_node="ap00" to_node="auto_connect"
            through_environment="ap00_env"
            standard="802.11a"/>
<connection from_node="ap01" to_node="auto_connect"
            through_environment="ap01_env"
            standard="802.11a"/>
<connection from_node="ap02" to_node="auto_connect"
            through_environment="ap02_env"
            standard="802.11a"/>
<connection from_node="ap03" to_node="auto_connect"
            through_environment="ap03_env"
            standard="802.11a"/>
<connection from_node="ap04" to_node="auto_connect"
            through_environment="ap04_env"
            standard="802.11a"/>
<connection from_node="ap05" to_node="auto_connect"
            through_environment="ap05_env"
            standard="802.11a"/>
<connection from_node="ap06" to_node="auto_connect"
```

```
            through_environment="ap06_env"
            standard="802.11a"/>
<connection from_node="ap07" to_node="auto_connect"
            through_environment="ap07_env"
            standard="802.11a"/>

</qomet_scenario>
```

B OpenWRT olsrd 設定

C /etc/config/olsrd

ノード 0 以外では以下のスクリプトから生成される設定を利用した。

```
#!/bin/sh

hostid=${1}

hna4=`printf 10.0.%d.0 ${hostid}`

cat > /etc/config/olsrd <<EOF
config olsrd
        # uncomment the following line to use a custom
        # config file instead:
        #option config_file '/etc/olsrd.conf'

        option IpVersion '4'

config Hna4
        option netaddr ${hna4}
        option netmask 255.255.255.0

config LoadPlugin
        option library 'olsrd_arprefresh.so.0.1'

config LoadPlugin
        option library 'olsrd_dyn_gw.so.0.5'

config LoadPlugin
        option library 'olsrd_httpinfo.so.0.1'
        option port '1978'
        list Net '0.0.0.0 0.0.0.0'

config LoadPlugin
        option library 'olsrd_nameservice.so.0.3'

config LoadPlugin
        option library 'olsrd_txtinfo.so.0.1'
        option accept '0.0.0.0'

config Interface
        list interface 'backhaul'
        option HelloInterval 5.0
        option HelloValidityTime 30.0
EOF
```

D OpenWRT network 設定

D.1 /etc/config/network

ノード 0 以外では以下のスクリプトから生成される設定を利用した。

```
#!/bin/sh

hostid=${1}
backhaul_addr=`printf 10.0.255.%d $((100 + ${hostid}))`
lan_addr=`printf 10.0.%d.1 ${hostid}`

cat > /etc/config/network <<EOF
config interface 'loopback'
        option ifname 'lo'
        option proto 'static'
        option ipaddr '127.0.0.1'
        option netmask '255.0.0.0'

config interface 'backhaul'
        option ifname 'wlan1'
        option proto 'static'
        option ipaddr '${backhaul_addr}'
        option netmask '255.255.255.0'

config interface 'lan'
        option ifname 'eth0'
        option type 'bridge'
        option proto 'static'
        option ipaddr '${lan_addr}'
        option netmask '255.255.255.0'

config interface 'wan'
        option ifname 'eth1'
        option proto 'dhcp'

config switch
        option name 'eth0'
        option reset '1'
        option enable_vlan '1'

config switch_vlan
        option device 'eth0'
        option vlan '1'
        option vid '1'
        option ports '0 1 2 3 4'
EOF
```

E OpenWRT firewall 設定

E.1 /etc/config/firewall

```
config defaults
        option syn_flood        1
        option input            ACCEPT
```

```
        option output           ACCEPT
        option forward          ACCEPT
#       option forward          REJECT
# Uncomment this line to disable ipv6 rules
#       option disable_ipv6     1

config zone
        option name             lan
        option network          'lan'
        option input            ACCEPT
        option output           ACCEPT
        option forward          ACCEPT
       #option forward          REJECT

config zone
        option name             wan
        option network          'wan'
        option input            REJECT
        option output           ACCEPT
        option forward          REJECT
        option masq             1
        option mtu_fix          1

config forwarding
        option src              lan
        option dest             wan

# We need to accept udp packets on port 68,
# see https://dev.openwrt.org/ticket/4108
config rule
        option name             Allow-DHCP-Renew
        option src              wan
        option proto            udp
        option dest_port        68
        option target           ACCEPT
        option family           ipv4

# Allow IPv4 ping
config rule
        option name             Allow-Ping
        option src              wan
        option proto            icmp
        option icmp_type        echo-request
        option family           ipv4
        option target           ACCEPT

# Allow DHCPv6 replies
# see https://dev.openwrt.org/ticket/10381
config rule
        option name             Allow-DHCPv6
        option src              wan
        option proto            udp
        option src_ip           fe80::/10
        option src_port         547
        option dest_ip          fe80::/10
        option dest_port        546
        option family           ipv6
        option target           ACCEPT
```

```
# Allow essential incoming IPv6 ICMP traffic
config rule
        option name             Allow-ICMPv6-Input
        option src              wan
        option proto    icmp
        list icmp_type          echo-request
        list icmp_type          echo-reply
        list icmp_type          destination-unreachable
        list icmp_type          packet-too-big
        list icmp_type          time-exceeded
        list icmp_type          bad-header
        list icmp_type          unknown-header-type
        list icmp_type          router-solicitation
        list icmp_type          neighbour-solicitation
        list icmp_type          router-advertisement
        list icmp_type          neighbour-advertisement
        option limit            1000/sec
        option family           ipv6
        option target           ACCEPT

# Allow essential forwarded IPv6 ICMP traffic
config rule
        option name             Allow-ICMPv6-Forward
        option src              wan
        option dest             *
        option proto            icmp
        list icmp_type          echo-request
        list icmp_type          echo-reply
        list icmp_type          destination-unreachable
        list icmp_type          packet-too-big
        list icmp_type          time-exceeded
        list icmp_type          bad-header
        list icmp_type          unknown-header-type
        option limit            1000/sec
        option family           ipv6
        option target           ACCEPT

# include a file with users custom iptables rules
config include
        option path /etc/firewall.user
```

F OpenWRT dhcp設定

F.1 /etc/config/dhcp

```
config dnsmasq
        option domainneeded     1
        option boguspriv        1
        # enable for dial on demand
        option filterwin2k      0
        option localise_queries 1
        # disable if upstream must serve RFC1918 addresses
        option rebind_protection 1
        # enable for RBL checking and similar services
```

```
        option rebind_localhost 1
        # whitelist RFC1918 responses for domains
        #list rebind_domain example.lan
        option local    '/lan/'
        option domain   'camp.wide.ad.jp'
        #option domain  'lan'
        option expandhosts      1
        option nonegcache       0
        option authoritative    1
        option readethers       1
        option leasefile        '/tmp/dhcp.leases'
        option resolvfile       '/etc/resolv.conf.camp-1303'
        #option resolvfile      '/tmp/resolv.conf.auto'
        #list server            '/mycompany.local/1.2.3.4'
        #option nonwildcard     1
        #list interface         br-lan
        #list notinterface      lo
        #list bogusnxdomain     '64.94.110.11'

config dhcp lan
        option interface        lan
        option start    100
        option limit    150
        option leasetime        12h

config dhcp wan
        option interface        wan
        option ignore   1
```

G OpenWRT fstab 設定

G.1 /etc/config/fstab

```
config global automount
        option from_fstab 1
        option anon_mount 1

config global autoswap
        option from_fstab 1
        option anon_swap 0

config mount
        option target   /root/log
        option device   /dev/sda1
        option fstype   ext4
        option options  rw,sync
        option enabled  1
        option enabled_fsck 0

config swap
        option device   /dev/sda2
        option enabled  0
```

H OpenWRT hotplug2 設定

H.1 /etc/hotplug2.rules

```
$include /etc/hotplug2-common.rules

SUBSYSTEM ~~ (^net$|^input$|button$|^usb$|^ieee1394$\
|^block$|^atm|^zaptel$|^tty$) {
        exec /sbin/hotplug-call %SUBSYSTEM%
}

DEVICENAME == watchdog {
        exec /sbin/watchdog -t 5 /dev/watchdog
        next-event
}
```

H.2 /etc/hotplug.d/button/buttons.sh

```
#!/bin/sh

if [ ${BUTTON} == "BTN_7" -a ${ACTION} == "released" ]; then
        /bin/sync
        /bin/echo 1 > \
/sys/class/leds/buffalo:blue:movie_engine/brightness
        /sbin/halt
fi
```