

# 修士論文

## IEEE 802.11n に基づく シームレスなハンドオーバー手法

2013 年 2 月 6 日

指導教員 相田 仁 教授



東京大学  
大学院工学系研究科  
電気系工学専攻

37-116466

出水 達也

## 概要

---

近年の無線 LAN の普及はめざましい。しかし1つアクセスポイント (AP) でカバーできる範囲が狭いため、無線 LAN 端末所持者が物理的に移動しながら通信する場合、頻繁に AP を切り替えることになる。ここで、AP を切り替えると通信が一旦切断され、サービス品質の低下につながる。

そこで本論文では、物理層におけるシームレスなハンドオーバーのための新手法を提案した。ここで“シームレス”とは、「パケットが欠落しないこと」「極端な遅延が発生しないこと」を意味する。本手法は、最近になって標準化が完了した IEEE 802.11n の特性である省電力モード (Spatial Multiplexing Power Save) と複数アンテナ通信 (MIMO) 環境を巧みに利用することで、既存のネットワーク構成にほとんど手を加えることなく、移動端末側の工夫でハンドオーバーを実現するものである。なお、無線 LAN の AP が設置されているビルなど屋内での使用を想定している。

本論文では、ソフトウェア無線機を用いて提案手法の実現可能性を確認した。また、ノート PC に搭載されている一般的な IEEE 802.11n 対応無線 LAN モジュールと、一般的な市販の IEEE 802.11n 対応無線 LAN AP を用いて、実環境下での使用を目指して提案手法の環境を実装した。さらに提案手法のために、Mobile IP の概念をもとにした簡易的な IP 層ハンドオーバーソフトウェアを開発した。

以上のように物理層・IP 層両方においてハンドオーバーが実現できる環境を整えて動作実験を行ったことで、提案手法が理論上のみならず実環境下でも有効であることを確かめることができた。さらにハンドオーバーにかかる時間を計測した結果より、AP の配置間隔や端末の最大移動可能速度についての1つの目安を提示した。また、より効率よくハンドオーバーを行うために、Mobile IP や IEEE 802.11n に対する拡張案を提示した。

# 目次

---

第 1 章	序論	1
1.1	はじめに	2
1.2	本論文の構成	3
第 2 章	研究の背景	5
2.1	関連研究	6
2.1.1	PDMA [3]	6
2.1.2	MultiNet [5]	6
2.1.3	IEEE 802.11r [6]	7
2.1.4	Fast Sleep Scan [8]	8
2.2	IEEE 802.11n [9]の概要	10
2.2.1	IEEE 802.11n の MAC フレーム	10
2.2.2	MIMO	11
2.2.3	省電力モード (SM Power Save)	12
2.2.4	変調方式・アンテナ数・通信レートの設定	14
2.3	Mobile IP / Mobile IPv6 の概要	15
第 3 章	提案手法	18
3.1	ハンドオーバー手順	19
3.2	送信用アンテナ数と受信用アンテナ数の違い	20
3.3	省電力モードの活用	20
3.4	MCS Index 指定の変更の活用	21
3.5	ハンドオーバー手法のまとめ	22
第 4 章	実験と評価	23
4.1	ソフトウェア無線機を用いた実験	24
4.1.1	実験機器	24
4.1.2	実験準備	25
4.1.3	フレーム処理速度の問題	27
4.1.4	物理層ハンドオーバーに向けた実験	28
4.1.5	物理層ハンドオーバーの実験結果と評価	30

4.2	市販の無線 LAN モジュールを用いた実験	31
4.2.1	移動端末用ノート PC への市販の無線 LAN モジュールの搭載	31
4.2.2	全体の構成	33
4.2.3	ハンドオーバーを制御するプログラムの開発	33
4.2.4	物理層ハンドオーバーの実装	38
4.2.5	IP 層ハンドオーバーの実装	41
4.2.6	物理層・IP 層両方のハンドオーバーの実験と評価	47
第 5 章	考察と議論	53
5.1	実験結果に基づく考察	54
5.2	アンテナが 3 本以上ある場合の通信用アンテナ増減方法	55
5.3	提案手法に適切な Mobile IP / Mobile IPv6 の実装と経路暗号化	57
第 6 章	結論	58
6.1	結論	59
6.2	今後の研究の指針	59
	謝辞	61
	参考文献	62
	発表一覧	64

## 目次

図 1 : IEEE 802.11b と PDMA の使用周波数帯 [1] [2].....	6
図 2 : 2 つの無線 LAN モジュールを用いた時と MultiNet を用いた時のスループットの比較 [3] .....	7
図 3 : IEEE 802.11r による高速 AP 間移動手法の手順 [5] .....	8
図 4 : Fast Sleep Scan のフローチャート [6].....	9
図 5 : Fast Sleep Scan における AP 探索結果 [6].....	9
図 6 : IEEE 802.11n における MAC フレームの構造 (HT-Mixed Mode) [7] .....	11
図 7 : 2x2 MIMO 通信の概念図.....	12
図 8 : Association Request フレームの構造 [7] .....	13
図 9 : SM Power Save フレームの構造 [7].....	13
図 10 : サポートするレート等を伝えるための Management フレームの構造 [7] 14	
図 11 : Mobile IP や Mobile IPv6 を用いない場合 .....	16
図 12 : Mobile IP や Mobile IPv6 におけるパケット転送の基本的な概念図 .....	17
図 13 : 提案手法の模式図.....	19
図 14 : 中間周波数変換の模式図.....	24
図 15 : 2 台の USRP を 1 つの無線 LAN モジュールに見立てるための機器構成 . 26	
図 16 : G.729 とパケットサイズをあわせた実験用パケット .....	26
図 17 : 電波暗室内での USRP の配置 .....	26
図 18 : 電波暗室内でのソフトウェア無線機による実験の様子 .....	27
図 19 : ハンドオーバー前の MIMO 通信中の状態 .....	30
図 20 : SISO に切り替えてハンドオーバー中の状態 .....	30
図 21 : USRP によるハンドオーバー実験の結果.....	30
図 22 : 本体内蔵の逆 F 型板金アンテナ .....	32
図 23 : 追加した逆 F 型フィルムアンテナ .....	32
図 24 : 各アンテナの配置.....	32
図 25 : IP 層ハンドオーバーも含めた提案手法の動作実験のネットワーク構成....	33
図 26 : 物理層・IP 層両方のハンドオーバーの流れ図 (1) .....	36
図 27 : 物理層・IP 層両方のハンドオーバーの流れ図 (2) .....	37
図 28 : thread 方式による Home Agent を経由したときと直接通信した時の Ping による RTT の比較.....	45
図 29 : IP 層ハンドオーバープログラムの制御パケット .....	47
図 30 : 電波暗室内での AP 類の配置.....	47

図 31 :	Correspondent Node で受信された Mobile Node からのテストパケットの数 .....	48
図 32 :	Correspondent Node で受信された Mobile Node からのテストパケット処理 時間の揺らぎ .....	48
図 33 :	Mobile Node で受信された Correspondent Node からのテストパケットの数 .....	50
図 34 :	Mobile Node で受信された Correspondent Node からのテストパケット処理 時間の揺らぎ .....	50
図 35 :	Correspondent Node と Mobile Node 間で UDP 通信をしながらハンドオー バーを行った際の経過時間とその内訳 .....	51
図 36 :	端末の最大移動可能速度と AP 通信可能範囲の重複距離の関係 .....	55
図 37 :	SM Power Save Frame (図 9) の SM Power Save 部分の拡大図 [7] ....	56

# 表目次

---

表 1: MCS Index (0~76) のうち, サポートしなければならないものの一覧 [9] .....	15
表 2: IEEE 802.11n に向けた Hydra の実装 [19].....	25
表 3: Ping による USRP 間での RTT (10 回平均) .....	27
表 4: USRP による実験に用いた機器の構成.....	29
表 5: 市販の無線 LAN モジュールによる実験に用いた機器の構成.....	35
表 6: 複数インターフェースを用いるプログラムの方式.....	45
表 7: thread 方式による Home Agent を経由したときと直接通信した時の Ping に よる RTT の比較.....	45
表 8: Correspondent Node と Mobile Node 間で UDP 通信をしながらハンドオー バーを行った際の経過時間とその内訳.....	52

# 第1章 序論

---



### 1.1 はじめに

近年の無線 LAN の普及はめざましい。今ではファストフード店、駅、空港といったあらゆる場所で公衆無線 LAN サービスが利用できるし、ノートパソコン、ネットブック、スマートフォンといった移動端末にも無線 LAN インターフェースが搭載されている。“いつでも、どこでも”インターネットに接続できる環境になったことはもはや周知の事実である。

無線 LAN はその機器自体の価格が比較的安く、通信速度も高速でしかも誰でも簡単に設置できることがこのように急速に普及してきた理由の一つとも言える [1]。しかし1つアクセスポイント (AP) でカバーできる範囲が狭いため、通常は専用のツールを使って無線 LAN AP のカバー範囲を診断 (サイトサーベイ) し、電波が隅々に行き渡るように複数の AP を設置していく [2]。ただしこれはあくまでも“なにかしら電波がつかめるように” AP を設置するだけであり、無線 LAN 端末所持者が物理的に移動しながら (たとえば歩きながら) 通信する場合、頻繁に AP を切り替えることになる。ここで、AP を切り替えると通信が一旦切断され、通信できない時間が生じてしまう。よって VoIP による音声通話やストリーミング動画閲覧といったリアルタイム性が求められる状況では途中で通信が途切れてしまい、これがサービス品質の低下につながる。

そこで本論文では、AP を切り替える際の通信切断時間をなくす、すなわち無線 LAN AP 間を移動する際のシームレスなハンドオーバーについて新手法を提案した。ここで“シームレス”とは、「パケットが欠落しないこと」「極端な遅延が発生しないこと」を意味する。

従来までもハンドオーバーについては盛んに研究されてきたが、この新手法は適用する際に既存のネットワーク構成に大きな改修を施さなくても良いため、導入が比較的容易であると考えられる。なお、無線 LAN の AP が設置されているビルなど屋内での使用を想定している。この新手法は物理層におけるハンドオーバーを実現するものである。最近になって標準化が完了した IEEE 802.11n の特性である、省電力モード (Spatial Multiplexing Power Save)、複数アンテナ通信 (MIMO) 環境を巧みに利用している。

ところで、実環境下では物理層のみならず、IP 層のハンドオーバーも必要である。従って本手法の有効性を示すためには、IP 層のハンドオーバーも意識した実証実験を行う必要がある。IP 層ハンドオーバー手法としてはすでに Mobile IP が存在するが、本論文では簡易的な Mobile IP の概念を実装し、それをを用いて“物理層・IP 層の両方においてハンドオーバーを実現する”という実証実験を行った。

### 1.2 本論文の構成

本論文は、以下の 6 章で構成されている。

#### 第 1 章 序論

序章として、本研究に至る動機および本論文の構成を示す。

#### 第 2 章 研究の背景

提案手法に関連する研究として、パケットが欠落することなくハンドオーバーを実現するための既存の手法を紹介し、それらの問題点を明らかにする。

また、IEEE 802.11n の仕様のうち、本論文で提案する物理層ハンドオーバーの新手法において重要な点を中心に概要を解説する。

さらに、IP 層ハンドオーバーの例として Mobile IP および Mobile IPv6 を紹介した上で、IP 層ハンドオーバーにおいて満たすべき条件を明らかにする。

#### 第 3 章 提案手法

パケットの欠落および極端な遅延時間の増加のない、シームレスな物理層ハンドオーバーについて新手法を提案し、解説する。

#### 第 4 章 実験と評価

ソフトウェア無線機を用いたシームレスな物理層ハンドオーバーの実験を行い、得られた結果を評価する。このことで提案手法の有効性を確認する。

次に無線 LAN モジュールを搭載したノート PC を移動端末に見立てて、提案手法の有効性を示すために必要な実験とその詳細を説明する。それによってシームレスな物理層ハンドオーバーのためのプログラムと IP 層ハンドオーバーのためのプログラムをそれぞれ実装し、物理層・IP 層両方のハンドオーバーを同時に行う。得られた結果を評価し、実環境下での提案手法の有効性を確認する。

#### 第 5 章 考察と議論

実験で得られた結果を考察し、実環境においてハンドオーバーを成功させるために必要な機器配置について議論する。また、提案手法をより効率よく応用するために、既存の IEEE 802.11n や Mobile IP 等に対する拡張を提案する。

## 第6章 結論

本論文をまとめ、現状の問題点、解決すべき課題と今後の研究の指針を述べる。

# 第2章 研究の背景

---

本章では、研究に至る背景として関連する研究について簡単に述べる。また、IEEE 802.11n および Mobile IP / Mobile IPv6 のなかでも、提案手法において重要な点を中心に述べる。

### 2.1 関連研究

今までも高速ハンドオーバー手法については盛んに研究されてきた。そのうちのいくつかをここで簡単に紹介するが、いずれも実用面を考えると課題が残るものばかりである。

#### 2.1.1 PDMA [3]

PDMA は Packet Division Multiple Access の略であり、パケット多重通信を示す。IEEE802.11b などでは周波数を複数のチャンネルに分けて通信するが、PDMA では図 1 のようにチャンネル分けをせず 1 つのチャンネルとして扱う。これにより、従来必要であった Channel Scanning が不要になり、ハンドオーバーの高速化が期待できる。このときのパケット衝突回避は従来の無線 LAN と同様に CSMA/CA を用いる。

ただし、PDMA は図 1 のように既存の無線 LAN のチャンネルすべての周波数帯を使用するため、既存の無線 LAN の AP と競合してお互いに悪影響を及ぼす恐れがある。今までの IEEE 802.11 では新規格の策定の際に常にレガシーな規格との共存を強く意識してきているため本方式の運用は難しく、また実用化を踏まえても現実的とはいえない。

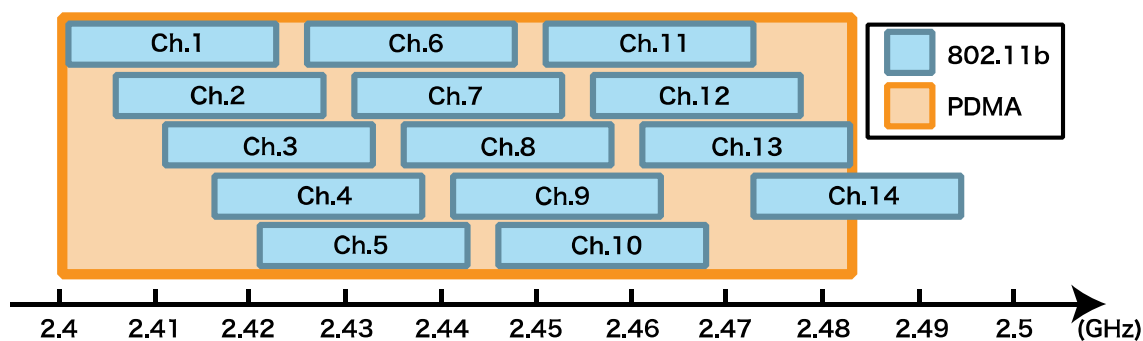


図 1： IEEE 802.11b と PDMA の使用周波数帯 [3] [4]

#### 2.1.2 MultiNet [5]

MultiNet は一つの無線 LAN インターフェースを仮想化して、複数のインターフェースに見せかける技術である。[5]によれば無線 LAN インターフェースの制御は、そのファームウェアよりもむしろほとんどを OS のドライバに依存している。よってドライバを修正して複数の仮想インターフェースとして OS 認識させ、物理的なインターフェースのパワーセ

ープモードを利用して、時間多重で仮想インターフェースに割り当てる。パワーセーブモードでは AP から MN にパケットが送出されないため、パケットの損失を抑えることができる。

また消費電力もインターフェース 1 つ分だけでよく、[5]では 2 つの物理インターフェースを用いた場合に比べて最大で 50%程度の消費電力の削減になると報告されている。さらに物理的にはインターフェースは 1 つしかないため現実的であるといえる。しかし 1 つのインターフェースを共有するため、図 2 に示すようにパフォーマンスの低下は避けられない。さらに、[5]は Ad hoc ネットワークと Infrastructure ネットワークに同時につなぐ場合を想定しており、AP 間ハンドオーバーについては触れられていない。

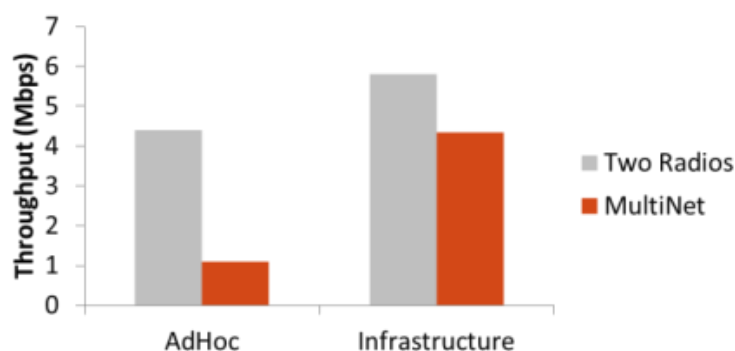


図 2： 2 つの無線 LAN モジュールを用いた時と MultiNet を用いた時のスループットの比較 [5]

### 2.1.3 IEEE 802.11r [6]

IEEE802.11r は最近になってようやく標準化された AP 間高速ハンドオーバー手法である。IEEE802.11r では AP 切断前に移動先 AP とあらかじめ認証を済ませておくことで、ハンドオーバーの高速化をはかるものである。これにより手順を図 3 のように簡素化することができる。また、移動先 AP の候補が複数あるときは、AP の負荷状況を IEEE 802.11e で定義される QoS サービスに従って決定し、最も快適に（電波強度が強く混雑していない環境で）通信できる AP を決定する。

一般に VoIP による通話では 50 ms 以内の遅延であれば通話の継続が可能であるといわれており、[7]によれば、かろうじてこの要件を満たしているといえる。またパケット損失も通常の場合に比べて小さい。ただし、これは VoIP に限った話であり、ストリーミング動画配信では画面の乱れが生じる可能性がある。

ここで、IEEE802.11r の遅延は AP を探索し情報を取得する時間によるものである。先に述べたように最も快適な AP を決定する機構は有しているが、そのための AP 探索および

一覧取得方法については IEEE802.11r では定義されていない。

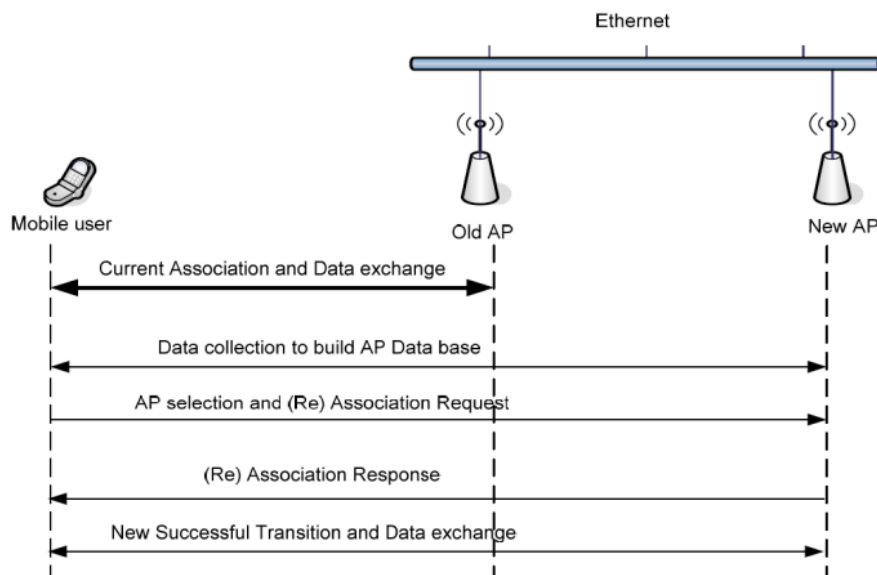


図 3： IEEE 802.11r による高速 AP 間移動手法の手順 [7]

#### 2.1.4 Fast Sleep Scan [8]

最後に、MN 単体で AP の検索と通信継続を両立する技術を紹介する。無線 LAN インターフェースをパワーセーブモード (PS モード, 11n における SM Power Save ではない) にすると AP からパケットが送出されず、AP のバッファに蓄えられる。これによりパケット損失を抑えることができる。ただしその間は通信できなくなるため、非常に短い時間だけ PS モードにし、その間に AP を探索し、すぐに通常モードに復帰する。

動作を図 4 に示す。ハンドオーバーをするための AP 一覧を構築するため、まず PS モードに移行する。そして 1 つのチャンネルで AP を探索する。一定の時間がたてば通常モードに復帰し、AP にたまっていたパケットを受け取る。移動先 AP を見つけることができなかつたときは、同様に別チャンネルで AP を探索する。

この手順を移動先 AP が見つかる、あるいはすべてのチャンネルに対して実行することで、通信と AP 探索の両方を並行して行うことができる。

10 個の AP を設置し 20 Mbps で通信しながら AP 探索実験をした結果を図 5 に示す。最もうまく探索できた場合でも AP は 10 個中 7 個までしか見つけることができなかつたことがわかる。また、後述するように一般に VoIP で品質の良い通信を確保するためにはパケットあたり 50 ms 程度の遅延までが許されるが、50 ms 以下を探索時間 (=通信できない時間) の目標とすれば、およそ 3 ~ 4 個の AP しか見つかっておらず、最適な解決策とは言いがたい。

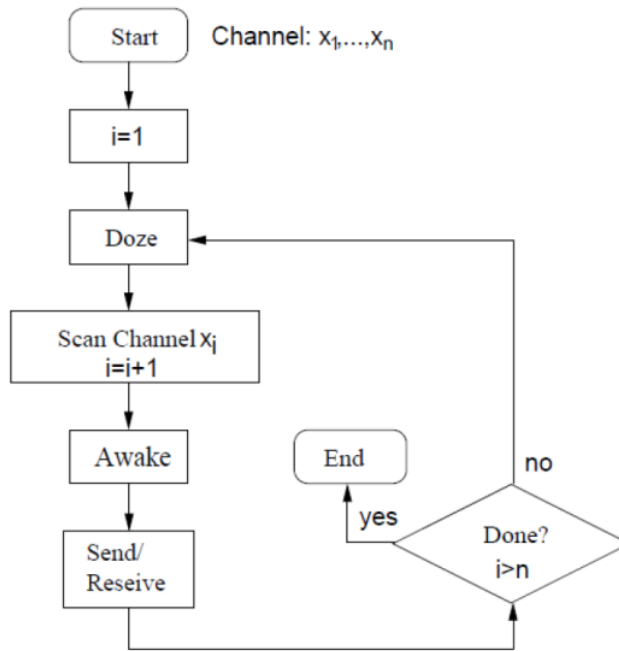


図 4 : Fast Sleep Scan のフローチャート [8]

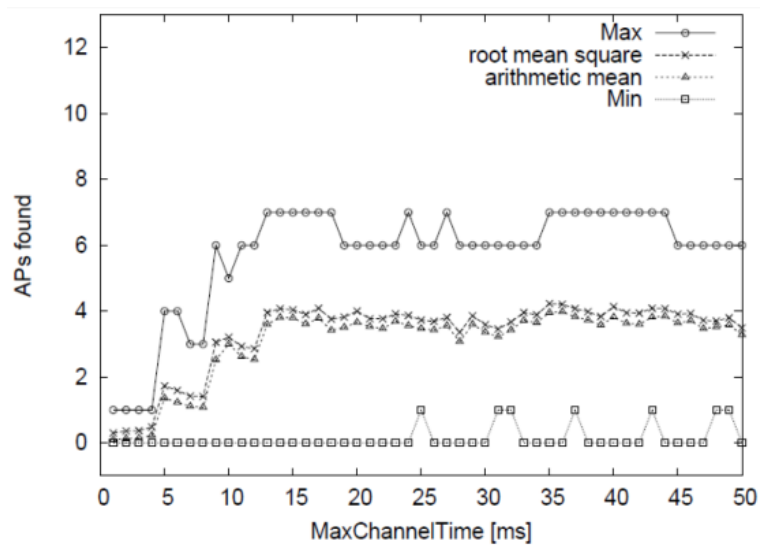


図 5 : Fast Sleep Scan における AP 探索結果 [8]

MaxChannelTime は AP からのフレーム待ち時間設定。設置 AP 数は 10。



### 2.2 IEEE 802.11n [9]の概要

IEEE 802.11n は 2009 年に標準化が完了した新しい無線 LAN 規格である。しかし標準化がすむ前からドラフト規格として公開されており、市販の多くの AP がこぞって対応してきたため、今では広く普及したといえる。

IEEE 802.11n では高速通信に対応するため、通信フレーム構造（データ構造のルール）が大きく変わっている。特にフレームアグリゲーションと呼ばれる技術の採用により構造が複雑になっている。そこで最初に IEEE 802.11n で用いられる一般的なフレーム構造を説明する。

また、チャンネルボンディング・MIMO・省電力モード（Spatial Multiplexing Power Save; SM Power Save）など、多くの新機能が採用されている。そこで新機能のうち、本研究で特に関連する MIMO と省電力モード（SM Power Save）について説明する。

さらに、IEEE 802.11n では 11a/b/g と異なり、複数の変調方式やアンテナ使用数などをサポートしている。よってこの指定方法についても説明する。

#### 2.2.1 IEEE 802.11n の MAC フレーム

図 6 のように、IEEE 802.11n の MAC フレームは複雑な構造になっている。これは既存の規格（IEEE 802.11a/b/g）との共存と 11n の高速通信を両立するためのものである。

図 6 の一番下が実際に電波にのせて AP と端末間で通信される情報（MAC フレーム）である。情報をやりとりするときは決まった長さごとに情報を分割して複数の MAC フレームを送ることになるが、このように送りたいデータ（Data）の前に多数のヘッダ（L-STF から最後の HT-LTF まで）をつける必要があるため、小さな Data を複数送るとパフォーマンスが低下する。特に、従来の 11a/b/g の AP や端末と共存できる HT-Mixed mode では 11n の通信では用いない L-STF, L-LTF, L-SIG が付加されており、ヘッダが非常に大きくなっている。そこで 11n では Data にできるだけ情報を詰め込めるように、次のような“フレームアグリゲーション”が用いられている。

上位層（TCP/IP など）からの転送データは 11n の MAC 層では MSDU として扱われる。この MSDU に必要なヘッダ・フッタをつけ、複数つなげて A-MSDU する。このように多重化されたものを MPDU の Body とする。さらに、MPDU 自体にも多重化用のヘッダ・フッタをつけて複数個つなげ、A-MPDU とする。11n ではこの A-MPDU が PSDU となり、適当な信号処理を行ってヘッダ・フッタをつけ、MAC フレームの Data とする。

このように 11n では MSDU と MPDU のそれぞれについて多重化（すなわち 2 度多重化）することで MAC フレームの Data 部分に挿入する情報量を増やし、MAC フレームのヘッダによるパフォーマンス低下の影響を小さくしている。従来の 11a/b/g では MSDU が二段

階の多重化なしにそのまま PSDU として送信されていたため、これに比べて 11n は送信の効率が良いといえる。

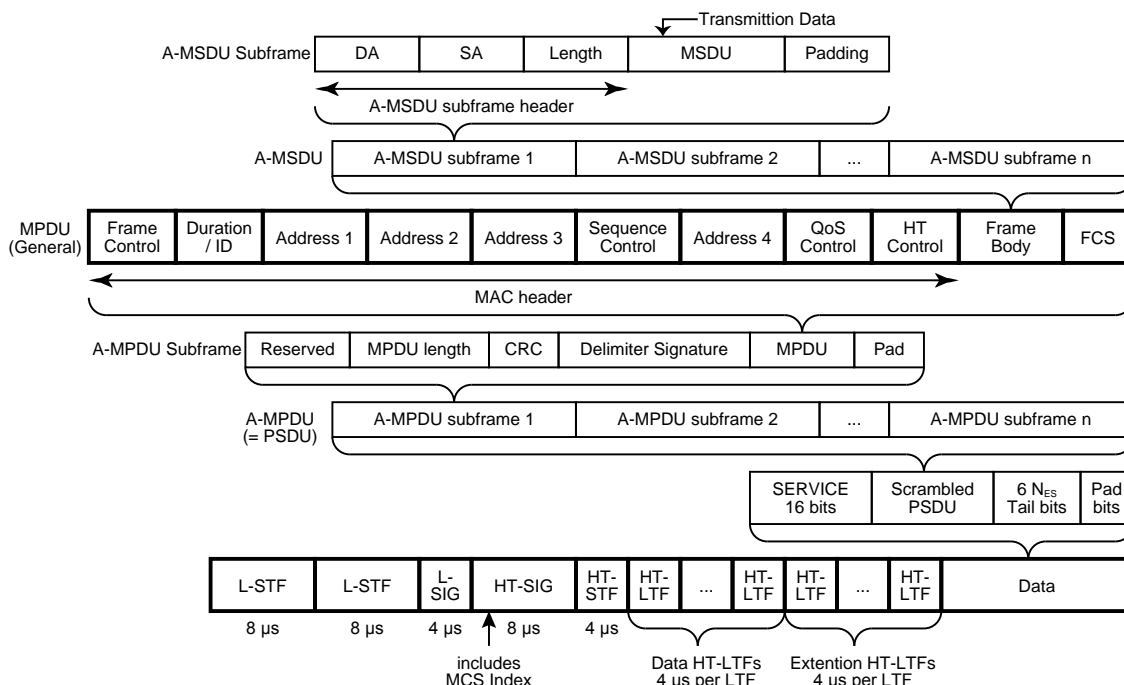


図 6 : IEEE 802.11n における MAC フレームの構造 (HT-Mixed Mode) [9]

TCP/IP など上位レイヤーからのデータが MSDU に入る。

11n では Data 部分が大きくなったために、1つの MAC フレーム自体も大きくなっている。よって電波帯域の占有時間が長くなり、CSMA/CA のフレーム衝突回避による（複数個端末があるときの全体としての）パフォーマンス低下が懸念される。しかし、11n では MIMO やチャンネルボンディングなどによって通信帯域自体が広がっているため、電波状況が良ければ結果的には占有時間はそれほど長くない。

## 2.2.2 MIMO

IEEE 802.11n では複数本のアンテナを用いて、同一周波数で同時に通信することで通信帯域を広くする MIMO (Multiple Input Multiple Output) 技術が採用されている。一般に同一周波数で複数の通信を同時に行うと混信してしまうため、無線 LAN では CSMA/CA と呼ばれる衝突回避機構が採用されている。しかし MIMO では、複数本のアンテナによって混ざってしまった信号を、計算により分離している。

たとえば図 7 のように、送受信共に 2 本ずつアンテナを用いる場合を考える。このとき、各送信 (input) 側アンテナ  $T_1, T_2$  から受信 (output) 側アンテナ  $R_1, R_2$  に向かう伝送路は  $h_{mn}$  で表せる。これを行列  $H$  で表すと、input 側を  $x$  , output 側を  $y$  , 伝送路で

の雑音を  $\mathbf{n}$  として次式で表せる。

$$\mathbf{y} = \mathbf{H}\mathbf{x} + \mathbf{n} \quad (1)$$

ただし,

$$\mathbf{H} = \begin{pmatrix} h_{11} & h_{12} \\ h_{21} & h_{22} \end{pmatrix}, \quad \mathbf{x} = \begin{pmatrix} T_1 \\ T_2 \end{pmatrix}, \quad \mathbf{y} = \begin{pmatrix} R_1 \\ R_2 \end{pmatrix}$$

すなわち  $\mathbf{H}$ ,  $\mathbf{n}$  が予め分かっているならば単なる二元一次の連立方程式である。よって  $\mathbf{H}^{-1}$  が計算できれば input 側の  $T_1$ ,  $T_2$  をただ一つに決定する (=信号を分離する) ことができる。実際にはプリアンプルでトレーニングを行うことで、データ伝送開始前に  $\mathbf{H}$ ,  $\mathbf{n}$  を決定している<sup>1</sup>。

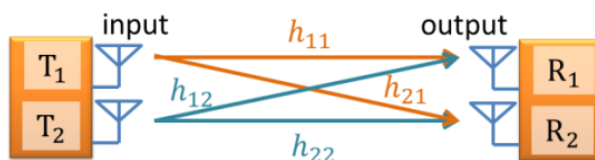


図 7: 2x2 MIMO 通信の概念図

### 2.2.3 省電力モード (SM Power Save)

2.1.4 節で紹介した手法では省電力モード中は完全に通信が途絶えてしまうが、IEEE 802.11n における省電力(Power Save)モードでは、2.2.2 節の MIMO における複数本のアンテナ通信を中止し、ただ 1 つのアンテナのみに通電するため通信を継続できる。MIMO による高速通信はできないが、通電するアンテナを減らすため電力を節約できる。

SM Power Save モードには Static と Dynamic の二種類ある。Static では、移動端末側が AP に向かって省電力モードへの移行を宣言すると、それ以降は明示的に省電力モードを解除しない限りアンテナ 1 本のみで通信を継続する。Dynamic では、無通信時にはアンテナ 1 本で通信を待ち受け、データを転送する際にはまず待ち受けている 1 本のアンテナでそれを検知し、通信中はすべてのアンテナに通電して高速通信を行う。

SM Power Save モードの有効・無効設定は、端末側 (AP ではない側) のみで行うことができる。また、AP は端末と SM Power Save モードで通信することをサポートしなければならない。端末側のモード移行の宣言は、AP とアソシエーションしようとする時に行う場合と、アソシエーション完了後に行う場合とで手法が異なる。

AP とアソシエーションしようとする時に SM Power Save モードを宣言するには、図 8

<sup>1</sup> たとえば  $T_1$  からのみ決まった信号を送出したときの受信側  $R_1$ ,  $R_2$  の値と、 $T_2$  からのみ同じ信号を送出したときの受信側  $R_1$ ,  $R_2$  の値を連立させれば、 $\mathbf{H}$  や  $\mathbf{n}$  が具体的に決定する。

に示すように Association Request Frame のヘッダの一つである HT Capabilities 中の HT Capabilities Info を構成する SM Power Save の値を 00 (Static) または 01 (Dynamic) に変更する。ネゴシエーション時に SM Power Save モードが有効になっていると、ネゴシエーションが終わってアソシエートした後の通信もずっと SM Power Save モードのままである。

通常モードで AP とアソシエートした後でも、図 9 に示す SM Power Save Frame を送信することで SM Power Save モードに移行することができる。SM Power Save Enabled ビットを 1 にすれば SM Power Save モードが有効になるし、0 にすれば中止することもできる。SM Mode ビットによって、Dynamic (1) か Static (0) を選択できる。

このように 11n では AP とネゴシエーションする時であっても、アソシエート後であっても、自由に SM Power Save モードに移行したり通常モードに復帰したりすることができる。

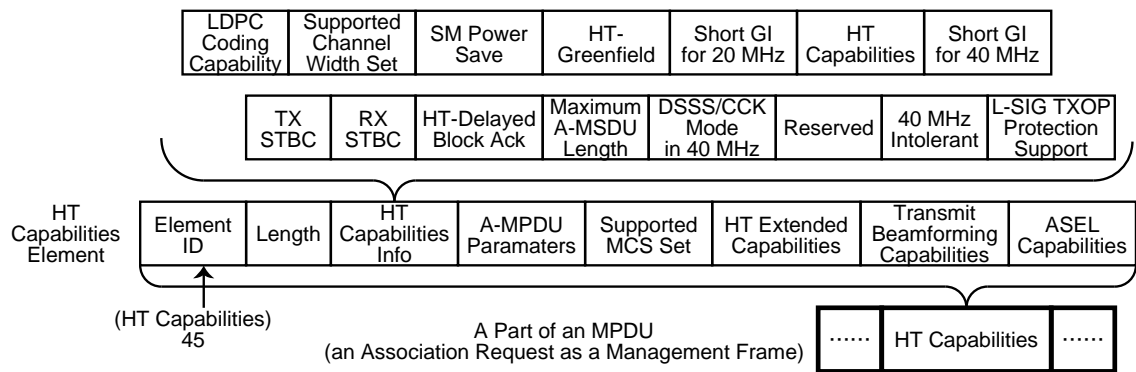


図 8 : Association Request フレームの構造 [9]

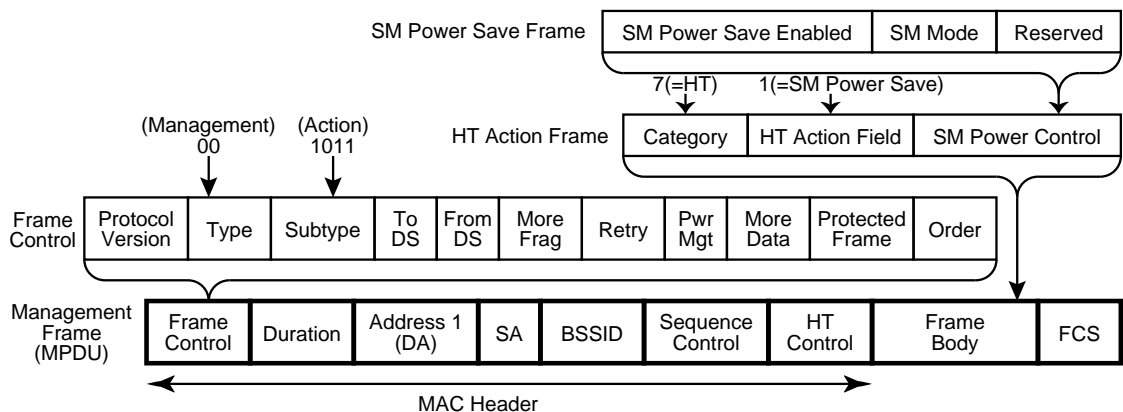


図 9 : SM Power Save フレームの構造 [9]

### 2.2.4 変調方式・アンテナ数・通信レートの設定

IEEE 802.11n では複数の変調方式，使用アンテナ数，通信レートが選択できるため，従来の 11a/b/g のように“送信レートだけ”を指定するのでは間に合わない。そこで，MCS Index という概念を導入している。

MCS Index は表 1 に示すように，0 から 76 までの数値と，変調方式，使用アンテナ数，通信レートなどを 1 対 1 で対応づけたものである。Management Frame の中でも Beacon, Association Request, Association Response, Reassociation Request, Reassociation Response, Probe Request, Probe Response を表す Frame では，図 10 に示すように Supported MCS Set という項目が設定できる。この項目では受信できる MCS Index を，この中の RX MCS Bitmask のビットをたてることで指定する。たとえば MIMO を使わずただ一つのアンテナのみの通信を受け入れる場合は，MCS Index のうち 0 から 7 の値のみをサポートすれば良いので，対応するビットを 1 に設定し，残りのビットを 0 にすれば良い。

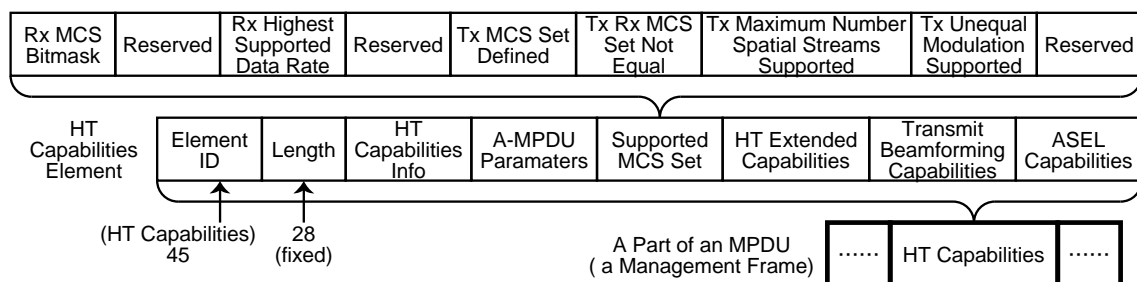


図 10： サポートするレート等を伝えるための Management フレームの構造 [9]

表 1: MCS Index (0~76) のうち, サポートしなければならないものの一覧 [9]

MCS Index	Spatial Stream	Modulation	Coding Rate	Data Rate (Mb/s) for 20 MHz	
				800 ns GI	400 ns GI
0	1	BPSK	1/2	6.5	7.2
1	1	QPSK	1/2	13.0	14.4
2	1	QPSK	3/4	19.2	21.7
3	1	16-QAM	1/2	26.0	28.9
4	1	16-QAM	3/4	39.0	43.3
5	1	64-QAM	2/3	52.0	57.8
6	1	64-QAM	3/4	58.5	65.0
7	1	64-QAM	5/6	65.0	72.2
8	2	BPSK	1/2	13.0	14.4
9	2	QPSK	1/2	26.0	28.9
10	2	QPSK	3/4	39.0	43.3
11	2	16-QAM	1/2	52.0	57.8
12	2	16-QAM	3/4	78.0	86.7
13	2	64-QAM	2/3	104.0	115.6
14	2	64-QAM	3/4	117.0	130.0
15	2	64-QAM	5/6	130.0	144.4

GI: Guard Interval

### 2.3 Mobile IP / Mobile IPv6 の概要

Mobile IP (MIP) および Mobile IPv6 (MIPv6) は移動端末が異なるネットワークに接続しても, 同一の IP アドレスを保持する仕組みである。

現状の IP プロトコルでは端末が異なるネットワークに移動してそこで通信が回復しても, 一部のパケットは失われてしまう可能性がある。IP プロトコルではパケットのヘッダに宛先 IP アドレスを書き込み, その宛先 IP アドレスを通信経路途中のルータが読み取り仕分けることでデータが通信先に届く [10]。しかし図 11 のようにパケットを送出した後に宛先の端末が移動して IP アドレスが変わってしまうと, パケットに書き込まれた宛先は存在しないことになり, そのパケットは宛先不明で失われてしまう。

UDP プロトコルのようなコネクションレス型通信の場合は, その通信を利用するアプリケーションが失われたパケットの再送を要求するよう独自に実装するかパケットをあきらめることになるが, いずれにしてもサービスの品質低下は避けられない。

またコネクション型である TCP プロトコルの場合は IP アドレスが変わるとコネクショ

ンを張り直す必要がある。失われたパケットは TCP の仕様で自動的に再送されるが、セッションを確立するまではパケットを送出できず遅延が生じ、やはりサービスの品質低下につながる。

よってシームレスなハンドオーバーを実現するためには同一の IP アドレスを保持する仕組みが必要であり、このために MIP および MIPv6 を用いることができる。

MIP および MIPv6 では MN の所有者の家などに HA を設置し、移動先でネットワークに接続すると IP アドレスを HA に通知する (Binding Update, BU)。BU を受け取った HA は移動先のネットワークと IP トンネルを構築する。以降は、HA に届いたパケットは IP トンネルを通じて MN に送られる (図 12)。

この手法ではパケットの送信元 (Correspondent Node, CN) が MIP および MIPv6 に対応していなくても MN と通信ができるが、すべての通信が一旦 HA を経由することになるため通信効率が落ちる。そこで特に MIPv6 では経路最適化機能も定義されている [10]。CN が MIPv6 に対応していれば自動的に経路の最適化が行われ、MN が移動した際に HA を介さず、CN から直接 MN にパケットが送られるようになる。つまり、CN が MIPv6 に対応していれば効率よく MN にパケットを送れるし、対応していなくても HA を介して MN にパケットを送れるということになる。

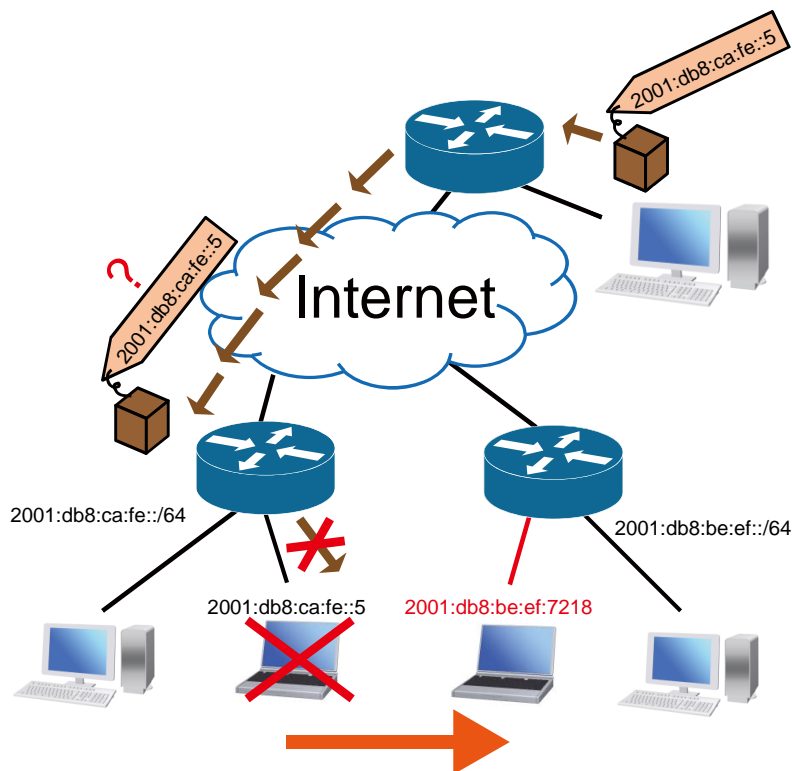


図 11 : Mobile IP や Mobile IPv6 を用いない場合  
端末の IP アドレスが変わってしまいパケットが到達しなくなる。

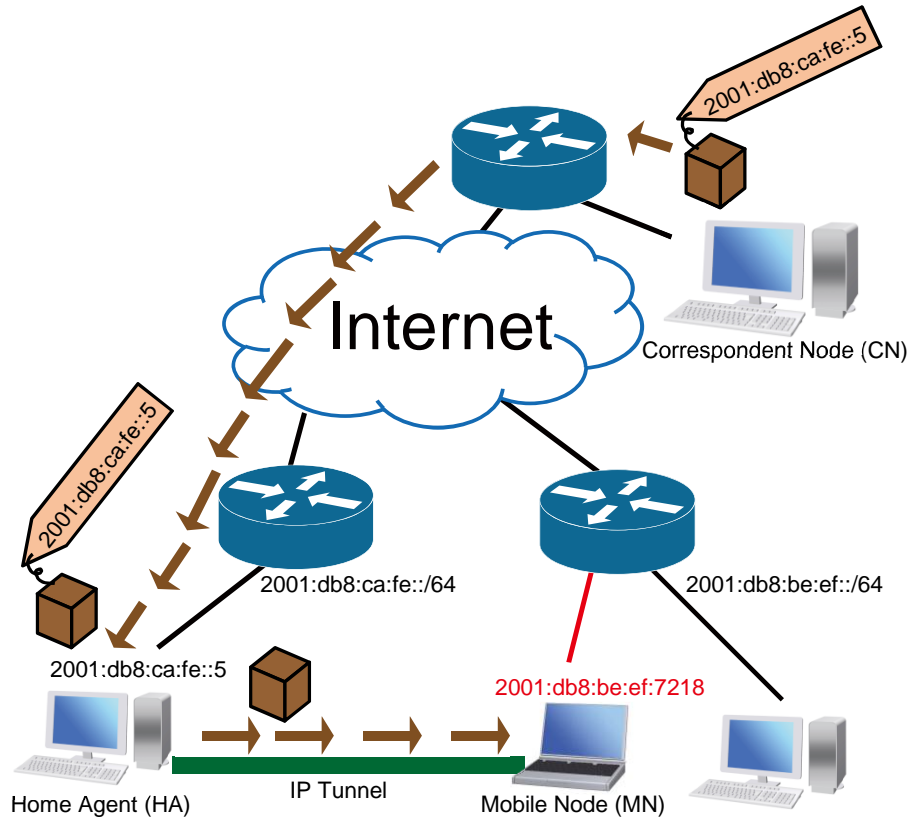


図 12 : Mobile IP や Mobile IPv6 におけるパケット転送の基本的な概念図  
(図の IP アドレスは IPv6 で表記したが IPv4 でも同様)



# 第3章 提案手法

---



3. 独立させたアンテナで他の接続可能な AP を探索する (図 1 に示すように複数あるチャンネルに次々と切り替えて AP を探す)
4. 接続可能な AP のうちのひとつとアソシエーションをする
5. 接続が確立されたら, 新しい IP アドレスを HA に通知する (Binding Update; BU)
6. IP トンネルを HA との間に構築する
7. 古い AP から切断する
8. ハンドオーバーに使ったアンテナを再び含めて新 AP と MIMO 高速通信を開始する

新 AP と接続した後のパケットの扱いは MIP および MIPv6 で解決できるので, “いかにして通信をハンドオーバー向けに切り替えるか” が焦点となる。ここで, 2.2 節で紹介したように, IEEE 802.11n では通信に用いるアンテナ数を制御する仕組みが備わっているため, これを利用することを考える。

## 3.2 送信用アンテナ数と受信用アンテナ数の違い

2.2.2 節で説明したとおり 11n では MIMO による通信が可能だが, 送信時に MCS Index を指定することができる。従って MCS Index を変更することで, 送信用アンテナ数を減らすことは容易である。また, 2.2.2 節の式(1)から分かるように送信側は自由にアンテナ数を減らしても良い。使用しなかったアンテナからのデータは 0 として受信されるため, 他のアンテナからのデータが失われることはないからである。しかし受信側が勝手にアンテナ数を減らしてしまうと, 行列  $(h_{mn})^{-1}$  の適用先である  $\mathbf{y} = \begin{pmatrix} R_1 \\ R_2 \end{pmatrix}$  が欠損するため, 送信データを復元することができなくなる。

従って重要なのは, 受信側が使用アンテナ数を減らそうとする際には, 前もってそのことを送信側に伝えることである。そのための方法を以下で説明する。

## 3.3 省電力モードの活用

2.2.3 節で紹介したように, 省電力モード (SM Power Save) を有効にすると受信に使用するアンテナを 1 本にすることができる。本来はこうすることで消費電力を抑えるものであるが, ここで空いたアンテナを使ってハンドオーバーすることを考える。IEEE 802.11n (HT MAC features, 項目番号 HTM17) によると, AP は移動端末側に対する SM Power Save モードによる通信をサポートしなければならない [9]。よって移動端末側が SM Power Save モードへの移行を通知すれば, 必ずそれ以降は, AP はアンテナ 1 本で受信可能な形でデータを送信することになる。

具体的には次のようにする。

1. ハンドオーバー開始時には SM Power Save Frame (図 9) を Static として送信し、MIMO をやめてアンテナ 1 本のみで通信を継続する
2. 残りのアンテナではチャンネルを次々に切り替えて AP を探索する
3. 接続先 AP を決めたら、Association Request Frame (図 8) で SM Power Save を Static としてアソシエーションをする
4. アソシエートが完了したら MIPv6 の手順を開始する
5. SM Power Save Frame (図 9) で省電力モードを解除し、MIMO 通信を開始する

なお、省電力モードでは必ず通信を続けるアンテナが 1 本（正確には空間ストリーム数が 1）になってしまう。よって例えば移動端末が 4 本アンテナを持っていたとしても、通信できるのは 1 本だけであるため、ハンドオーバー中は「3 本のアンテナによる MIMO が可能であっても」MIMO を用いた高速通信をあきらめなければならない。ただし、ハンドオーバーするときは電波状況が悪いと考えられるため、そもそも高速通信ができない可能性が高いと思われる。

#### 3.4 MCS Index 指定の変更の活用

2.2.4 節や表 1 で紹介したように、MCS Index を操作することで通信に用いるアンテナ数（正確には空間ストリーム数）を制御することができる。この仕組みを用いて、ハンドオーバー時に“通信用アンテナ数”と“ハンドオーバー用アンテナ数”を設定する方法を考える。

ハンドオーバー時には AP 探索に全チャンネルをスキャンする必要があるため、ハンドオーバーに用いるアンテナ数が多いほど効率が良い。また、通信にも複数のアンテナを用いた方が MIMO による高速通信が期待できる。よって、3 本以上のアンテナを移動端末が持っているときは、状況に応じてアンテナを振り分けることが可能である。

ただし、現状では“送信時に”レートを指定することはできる（図 10）が、受信時にはその仕組みがない。さらに、最適なレートを決定する仕組みは規定されていない [11]ため、一般に市販されている AP や移動端末では、フレーム伝送の失敗による再送<sup>2</sup>や信号強度から経験的に予め決められたレートを使用することにしている。また、信号強度とレートの関係についての研究もされている [12]。

ハンドオーバーが起きるときには電波状況が悪い、すなわち通信レートが落ちていると考えられる。このときは MCS Index が小さく、1 つの空間ストリーム数で通信しているこ

---

<sup>2</sup> 一定時間内に ACK フレームが返送されてこないときは通信が失敗したと判断して再送する。

とが推測される<sup>3</sup>。よって暗黙のうちに一部のアンテナをハンドオーバーに用いることが可能である。

## 3.5 ハンドオーバー手法のまとめ

3.3 節の省電力モードを使う手法では、通信を継続できるアンテナが1本に限られるものの、容易にハンドオーバー用のアンテナを確保することができる。一方で 3.4 節の MCS Index を操作する方法では、明示的に受信時のレートやアンテナ数を制御することはできないものの、ハンドオーバーが起こる状況を考えれば、3本以上の空間ストリームを扱える場合、より効率よく複数のアンテナを用いることができる。

まとめると、この両方の手法を組み合わせた次のような手順が良いと考えられる。

1. 信号劣化や強度不足を検知し、ハンドオーバーを検知する
2. このときにすでに MCS Index が 0 から 7 までの値で通信している場合は、そのままハンドオーバー用にアンテナを 1 本以上確保する
3. 空間ストリーム数と実装アンテナ数が同じ場合は省電力モードへの移行を通知するか信号強度をあえて落とすなどして、ハンドオーバー用のアンテナを確保する
4. ハンドオーバー用アンテナで他の AP を探索する
5. 新 AP と省電力モードでアソシエートする
6. 接続が確立したら MIP または MIPv6 の手順を踏む
7. IP トンネルが構築されたら旧 AP から切断し、全アンテナで通信を再開する

IEEE 802.11n では高速通信を実現するために 2.2.1 節で説明した“フレームアグリゲーション”が行われている。しかし、ハンドオーバーが発生するほど通信状況が悪化していれば、巨大なフレームを送受信するのに時間がかかり、ハンドオーバーに遅れが生じる恐れがある。研究の目的は異なるが、この問題の解決には [13]のように、動的にアグリゲート数を制御するのが良いと思われる。

---

<sup>3</sup> このときはビームフォーミングを行って、通信帯域よりも通信品質を確保していると推測される。どのアンテナを使うかは送信側次第である。

# 第4章 実験と評価

---

本章では提案手法の有効性を確認するため、まずはソフトウェア無線機を用いて物理層ハンドオーバー実験を行う。また、そこで得られた結果を評価する。次に一般的な無線 LAN モジュールを用いて物理層ハンドオーバー実験を行う。さらに、IP 層ハンドオーバーも合わせて行い、実環境下における提案手法の有効性を評価する。

## 4.1 ソフトウェア無線機を用いた実験

無線 LAN では通信に使用する周波数帯が高いため、そのままでは扱いが難しい。そこで通常は中間周波数変換によって周波数を落として処理することになる。

図 14 に示したように、中間周波数変換ではアンテナで受信した無線電波に局部発信器からの適切な周波数の正弦波を入力することで、より低い周波数に変換する。局部発信器の周波数を変更することで信号を任意の周波数に落とすことができるため、通信チャンネルを動的に変更することができる。

ここで、IEEE 802.11n の MIMO では全アンテナで同一周波数での通信を行うため、ハードウェアの実装によっては一つの局部発信器を全アンテナで共有している可能性がある。しかし提案手法ではハンドオーバー用のアンテナを独立して制御することになるため、独立した局部発信器が必要である。そこで、アンテナごとに異なる周波数で通信することを考えて、一般の無線 LAN モジュールに比べて応用範囲のひろいソフトウェア無線機で実験することにした。

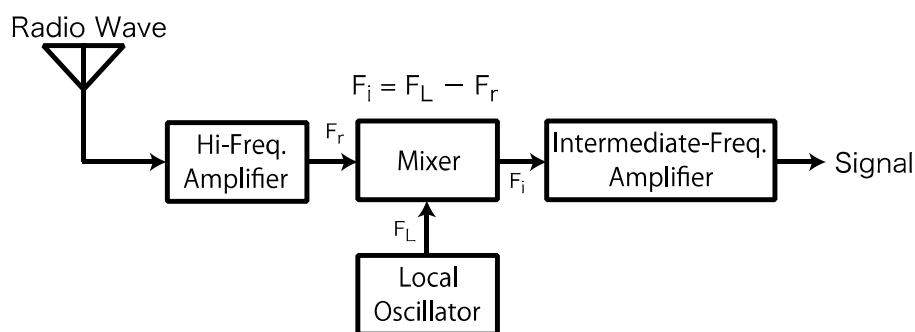


図 14： 中間周波数変換の模式図

### 4.1.1 実験機器

ソフトウェア無線 GNU Radio [14]に対応した製品として有名なものに Ettus Research 社 [15]の USRP シリーズがある。今回は USRP 1 を用いることにした。この機器は単体で MIMO に対応しており、最大で 2 組の RF ボードを搭載することができる。実験では、2.4

GHz 帯の電波を送受信できる RFX2400 ボードを 1 台の USRP につき 2 組搭載し、MIMO 通信できるようにした。

また、IEEE 802.11n に対応したテストベッド Hydra 0.4.1 [16] [17]を用いる。これは USRP 1 向けに GNU Radio 3.2.2 で記述されたもので、MIMO 通信にも対応している [18]。これらの諸元を表 2 に示す。

表 2 : IEEE 802.11n に向けた Hydra の実装 [19]

System Bandwidth	1 MHz *
Center Frequency	2.4 – 2.5 GHz
Maximum TX Power	10 mW
Modulation	BPSK, QPSK, 16-QAM, 64-QAM
Coding	Bit-Interleaved Binary Convolution Coding
SISO Data Rates	0.65, 1.30, 1.95, 2.60, 3.90, 5.20, 5.85, 6.50 Mbps *
MIMO Data Rates	2× SISO Data Rates
Diversity Schemes	Cyclic Delay Diversity, Space-time Coding, Spatial Mapping, Beamforming

\* indicates non-standard values

#### 4.1.2 実験準備

G.729 コーデックは圧縮率が高く、VoIP で広く用いられている。そこで、G.729 と同サイズの packets を送信する実験を試みることにした。

G.729 で用いる RTP では 20 ms 以上の間隔で送出するのが一般的である [20]。G.729 で定義される 8 kbps の通信を 20 ms 間隔で行うためには、1 packet に 20 Byte の情報をのせる必要がある。RTP ヘッダは 12 Byte であるため、あわせて 32 Byte のデータを UDP で送出することになる [21]。そこで、UDP ペイロード 32 Byte に「送出時刻」と「packet 順序番号」を載せた実験用 packet (図 16) を生成し UDP で送信するプログラム、およびその packet を受信するプログラムを準備した。

さらに本論文の提案手法では、通信に用いるアンテナを途中で増減する作業が必要になる。これは Hydra には実装されていないため、ソースコードに手を加えて、実験途中で使



用アンテナ数を増減できるようにした。しかし試行錯誤したものの、アンテナごとに異なる周波数で通信できるようにするには至らなかった。そこで、2台の USRP をまとめて1つのインターフェースだと見なし、使用アンテナ数の合計が2を超えないように制御することにした。Hydra と物理マシンの接続・制御構成を図 15 に示す。

実験は、図 17、図 18 に示すように USRP を 4 台電波暗室内に並べて行った。

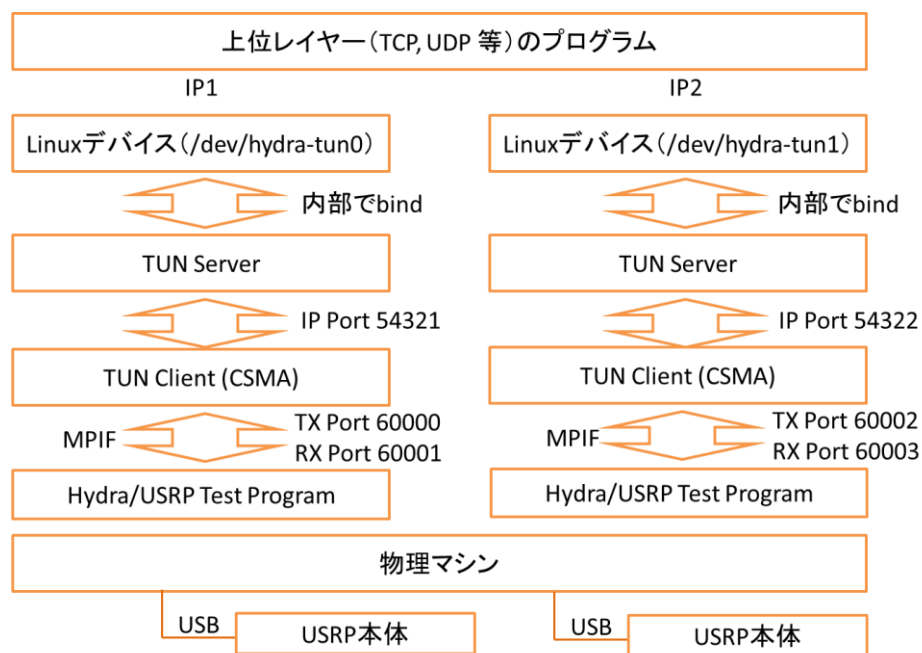


図 15： 2 台の USRP を 1 つの無線 LAN モジュールに見立てるための機器構成

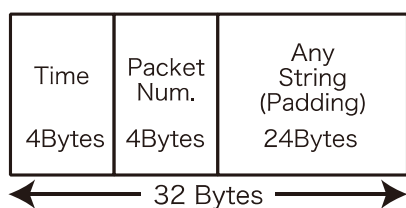


図 16： G.729 とパケットサイズをあわせた実験用パケット

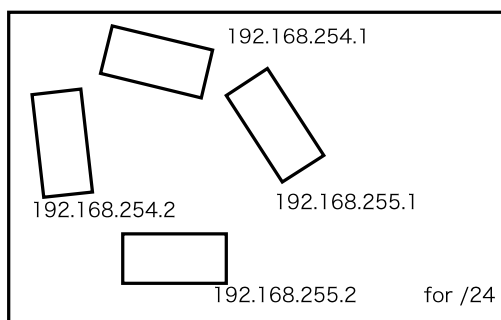


図 17： 電波暗室内での USRP の配置

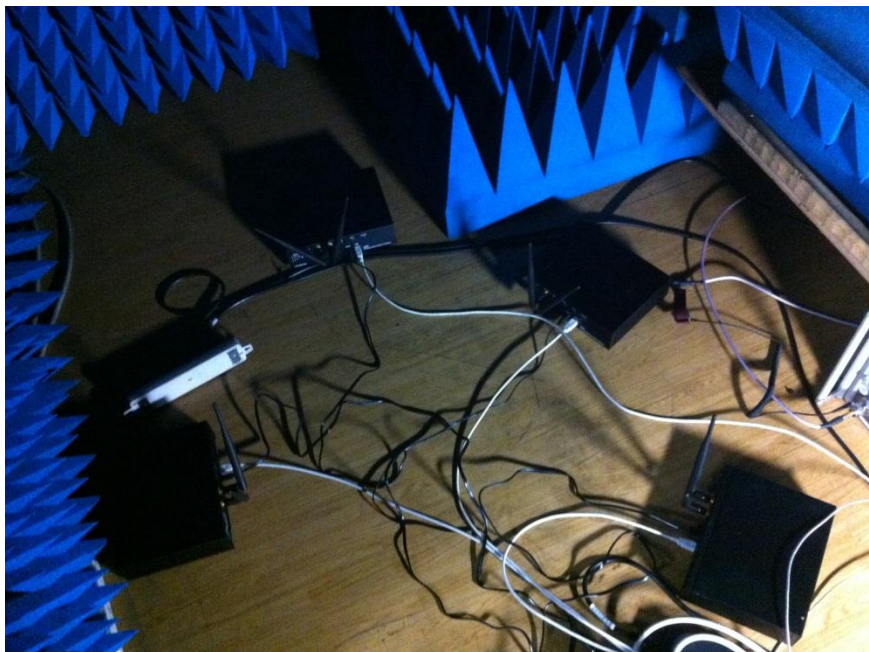


図 18： 電波暗室内でのソフトウェア無線機による実験の様子

#### 4.1.3 フレーム処理速度の問題

USRP 1 および Hydra を用いて 20 ms 間隔で図 16 のパケットを送受信する実験を行ったところ、およそ 10 パケットに 1 つしか受信されておらず、ハンドオーバーをする以前にパケットロス率が非常に高いことが分かった。そこで ping で疎通および RTT を確認したところ、表 3 に示す結果が得られた。

至近距離での RTT がおよそ 250 ms であることから、「パケット生成→変調→送信→受信→復調→パケット解析」におよそ 130ms 程度かかっていることが推測される。これは USRP や Hydra の制限であると考えられるため、パケット送出間隔を 200 ms に改めて実験を行うことにした。

表 3： Ping による USRP 間での RTT (10 回平均)

	192.168.255.1→255.2	192.168.254.1→254.2
RTT	245 ms	264 ms

### 4.1.4 物理層ハンドオーバーに向けた実験

表 4 に示すような条件で，次の手順で実験を行った。

1. 図 19 のように Mobile Node から Node A に向かって MIMO 通信 (MCS Index は 15) で先述のパケットを送出する。
2. MCS Index を 7 にすることで空間ストリーム数を 1 に減らす。
3. 使用アンテナを 1 つに減らす。
4. UDP パケットの送出先を Node B に切り替える。これは図 20 に相当する。

最後の手順 4 においてパケットロスが発生しなければ，物理層ハンドオーバーが成功したと言えることになる。

表 4: USRP による実験に用いた機器の構成

#1&2 Ubuntu 9.10 (VMware) on hp ProBook 6550b		
IP Address	192.168.255.1	192.168.254.1
Center Freq.	Rx: 2.416 GHz Tx: 2.424 GHz	Rx: 2.396 GHz Tx: 2.404 GHz
CPU	Intel Core i5-560M 2.67 GHz, 2 cores, 4 threads	
RAM (Virtual)	2 GB	2 GB
#3 Ubuntu 9.10 on DELL vostro 1200		
IP Address	192.168.255.2	
Center Freq.	Rx: 2.416 GHz Tx: 2.424 GHz	
CPU	Intel Core2 Duo T7250 2.00 GHz, 2 cores, 2 threads	
RAM	2 GB	
#4 Ubuntu 9.10 on DELL vostro 1200		
IP Address	192.168.254.2	
Center Freq.	Rx: 2.396 GHz Tx: 2.404 GHz	
CPU	Intel Core2 Duo T7250 2.00 GHz, 2 cores, 2 threads	
RAM	2 GB	

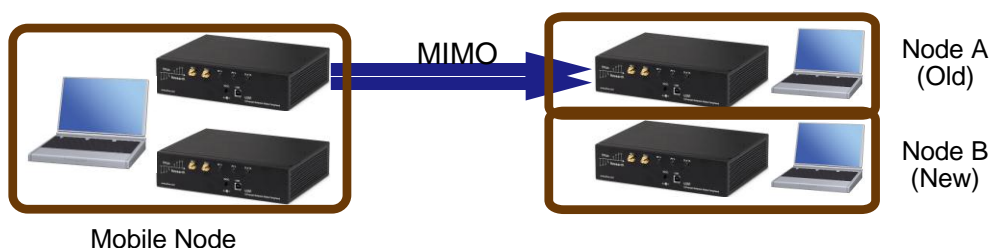


図 19： ハンドオーバー前の MIMO 通信中の状態

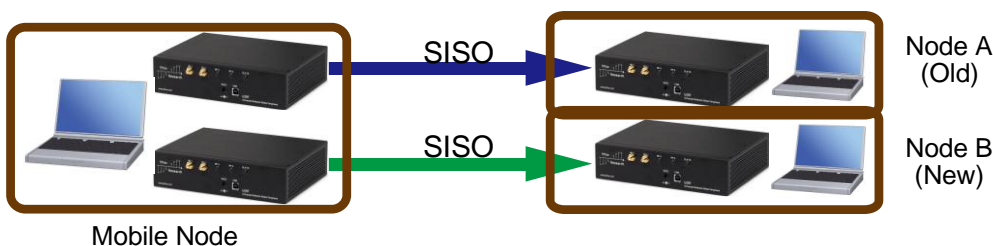


図 20： SISO に切り替えてハンドオーバー中の状態

#### 4.1.5 物理層ハンドオーバーの実験結果と評価

図 21 に示すように通信全体を見ればパケットロスやデュプリケートが発生しているが、ハンドオーバーのための操作（赤線）を行った直後には発生していない。従って、物理層におけるハンドオーバーそのものは成功したと言える。

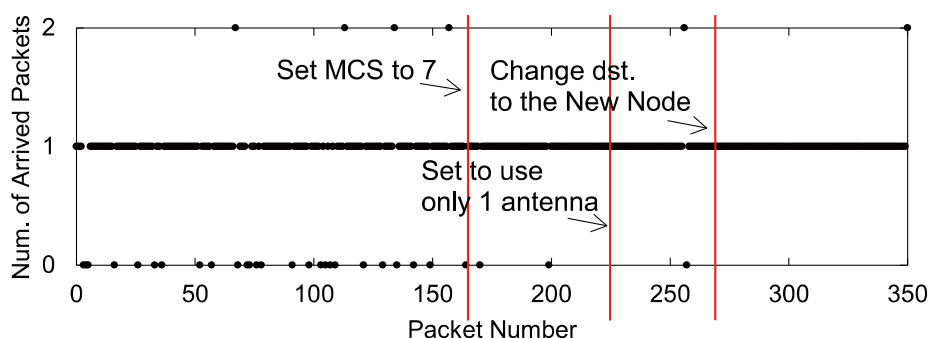


図 21： USRP によるハンドオーバー実験の結果

縦軸において、0 はパケット損失、1 はパケット到達、2 は duplicate 発生を示す。

ここで、ハンドオーバー時以外のパケットが正常に受信されていない。この実験は電波暗室内で行ったものであり、さらに各 USRP の組み合わせで中心周波数を変えているので、電波状況が悪いということは考えられない。従って、実験機器やソフトウェアに起因するものだと考えられる。

また、USRP の構造上パケットの処理をソフトウェアで行うため、高速動作させることが困難であることが分かった。特に、VoIP 通信の場合本来なら 20ms 間隔であるが、表 3 の結果から 200ms 間隔にしても快適な通信は難しいと思われる。

以上の評価に加え、結果的には USRP 1 台で複数の周波数で通信するには至らなかったことも踏まえて、より現実的な環境下でも実験が行えるよう、以降は一般的な無線 LAN モジュールを用いて実験することにした。

### 4.2 市販の無線 LAN モジュールを用いた実験

提案手法の有効性は 4.1 節で確認できたが、実験装置および構成に起因する処理速度に課題が残った。そこで、ノート PC を移動端末に見立てて、内蔵の無線 LAN モジュールを用いてハンドオーバー実験を行うことにした。通常の無線 LAN モジュールなので、一般に使用されている無線 LAN 端末と同程度の処理速度で動作することが期待されるためである。

なお、今回の実験ではモジュールのアンテナ数が最大 2 であるため、第 3 章で提案した手法のうち“SM Power Save”によってアンテナ数を制御する方式を採用することにした。

#### 4.2.1 移動端末用ノート PC への市販の無線 LAN モジュールの搭載

通常の無線 LAN モジュールは複数の周波数で同時に通信することはできないため、移動端末とするノート PC に無線 LAN モジュールを 2 つ内蔵することにした。ここで、本来であれば 1 つのモジュール内の 2 本以上の RF 回路を独立して制御するべきところであるため、2 つのモジュールを同じ Intel 製の「Centrino Advanced-N 6200」で統一することで本来の構成との差異を小さくとどめた。このモジュールは 2x2 MIMO 通信をサポートするため、アンテナを 2 本接続する必要がある。

また、今回用いたノート PC には無線 LAN 用のアンテナ（逆 F 型板金アンテナ、図 22）が 3 本しか搭載されていなかったが、実験ではモジュールが 2 つあるため最低 4 本のアンテナが必要である。そこで Tyco 製無線 LAN アンテナ（フィルムアンテナ、図 23）を別途内蔵することにした。ここでフィルムアンテナを採用したのは、ノート PC 内に内蔵する際のスペースの都合である（図 24）。

図 19、図 20 と同様に、2 つの無線 LAN モジュールをまとめて 1 つのモジュールとして見なすことにした。



図 22： 本体内蔵の逆 F 型板金アンテナ



図 23： 追加した逆F型フィルムアンテナ

左図のようにディスプレイの裏に配置し、右図のようにアンテナ部分が隠れないようにした

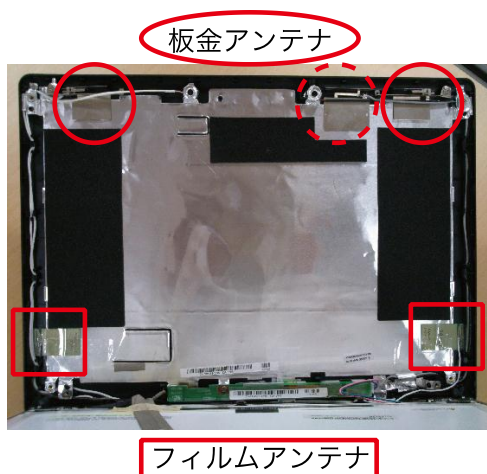


図 24： 各アンテナの配置  
破線は今回未使用のアンテナ

## 4.2.2 全体の構成

4.1 節で物理層ハンドオーバーの適用可能性が認められたので、図 25 に示す構成で IP 層ハンドオーバーも含めた実験を行った。ここで、実環境下におけるハンドオーバーを意識して、Mobile Node の IP アドレスは Router から DHCP を用いて取得する構成にした。

実験では G.729 に模したパケット (図 16) を実際の場合と同じ 20ms 間隔で送受信することにした。すなわち、Mobile Node と Correspondent Node との間で通信を継続しつつ、Mobile Node の接続先を AP1 から AP2 (もしくは AP2 から AP1) に切り替えた際にパケットが欠落しないことを確かめることにした。

各機器の機種等を表 5 にまとめた。ここで、Home Agent は Mobile Node の拠点となるネットワークに置く中継器であるが、実運用の際には各 Mobile Node の所属するネットワークごとに多数展開することを想定して、環境をコピーしやすい仮想マシン上に準備することにした。

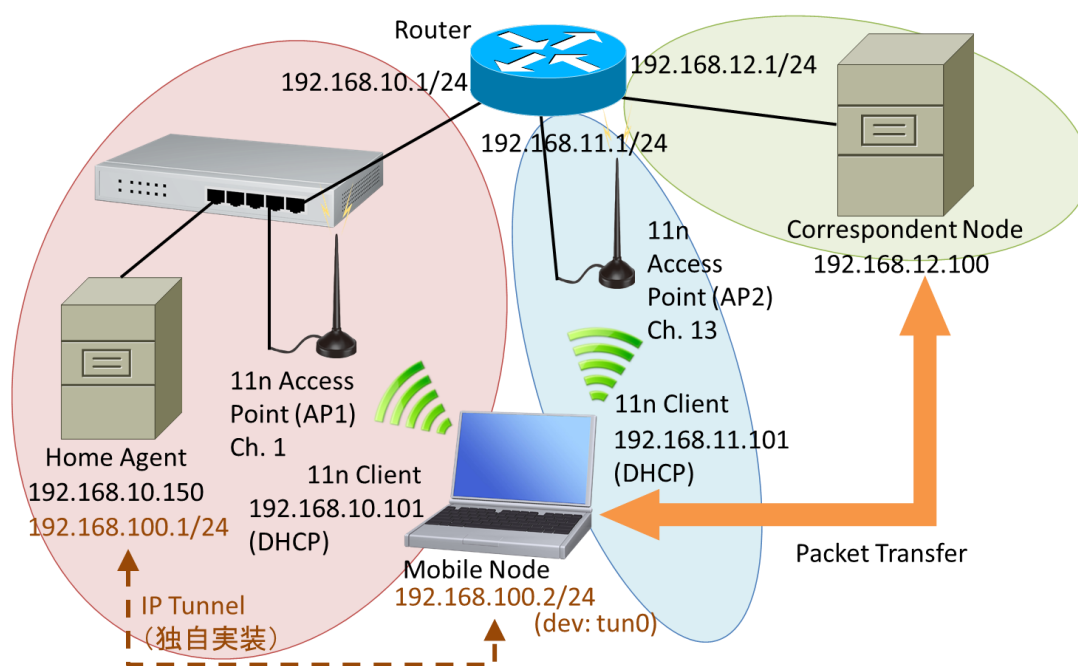


図 25： IP 層ハンドオーバーも含めた提案手法の動作実験のネットワーク構成

## 4.2.3 ハンドオーバーを制御するプログラムの開発

### 4.2.3.1 制御プログラムの仕組み

今回の実験では物理層ハンドオーバーと IP 層ハンドオーバーの両方を制御する必要がある。そこで、IP 層ハンドオーバー用プログラムを外部から制御できる仕組みにし、物理層ハンドオーバーのプログラムから制御できるようにした。ハンドオーバーの手順を図 26、



図 27 に示す。

手順において、IP 層ハンドオーバー等はミリ秒オーダーでの動作が予想されるものの、物理層ハンドオーバーのための AP 検索 (4.2.4.3 節で述べる `wpa_supplicant` を使用) には秒オーダーの時間がかかることが予想された。よって、制御プログラム自体の動作速度はそれほど問題にならないと言えるため、制御プログラムは Python (version 2.7.3) で開発することにした。

アンテナ数を減らすには、送信用と受信用のそれぞれについて設定しなければならない。送信用アンテナを 1 本にするには MCS Index 0~7 のいずれかに設定すれば良く、受信用アンテナを 1 本にするには SM Power Save を Static に設定すれば良い。

表 5： 市販の無線 LAN モジュールによる実験に用いた機器の構成

Home Agent		AP1 & L2 Switch	
Model	VMware on hp ProBook 6550b	Model	Planex MZK-WG300NX
OS	Ubuntu 12.04 (32bit) Linux Kernel 3.2.30	Wired LAN	1000 Base-T
CPU	Intel Core i5-560M 2.67 GHz, 2 cores, 4 threads	Wireless LAN	IEEE 802.11b/g 11n 300 Mbps (2T2R)
RAM	2 GB (Virtual)	Encrypt	WPA2/PSK(AES)
Mobile Node		Channel	1 (w/ Channel Bonding)
Model	DELL vostro 1200	AP2	
OS.	Ubuntu 12.04 (32bit) Linux Kernel 3.6.4 (Modified)	Model	NEC Aterm WR8160N(ST)
CPU	Intel Core2 Duo T7250 2.00 GHz, 2 cores, 2 threads	Wired LAN	100 Base-TX
RAM	2 GB	Wireless LAN	IEEE 802.11b/g 11n 300 Mbps (2T2R)
Correspondent Node		Encrypt	WPA2/PSK(AES)
Model	DELL vostro 1200	Channel	13 (w/ Channel Bonding)
OS.	Ubuntu 12.04 (32bit) Linux Kernel 3.2.0	Router	
CPU	Intel Core2 Duo T7250 2.00 GHz, 2 cores, 2 threads	Model	NEC IX2015
RAM	2 GB	Wired LAN	100 Base-TX

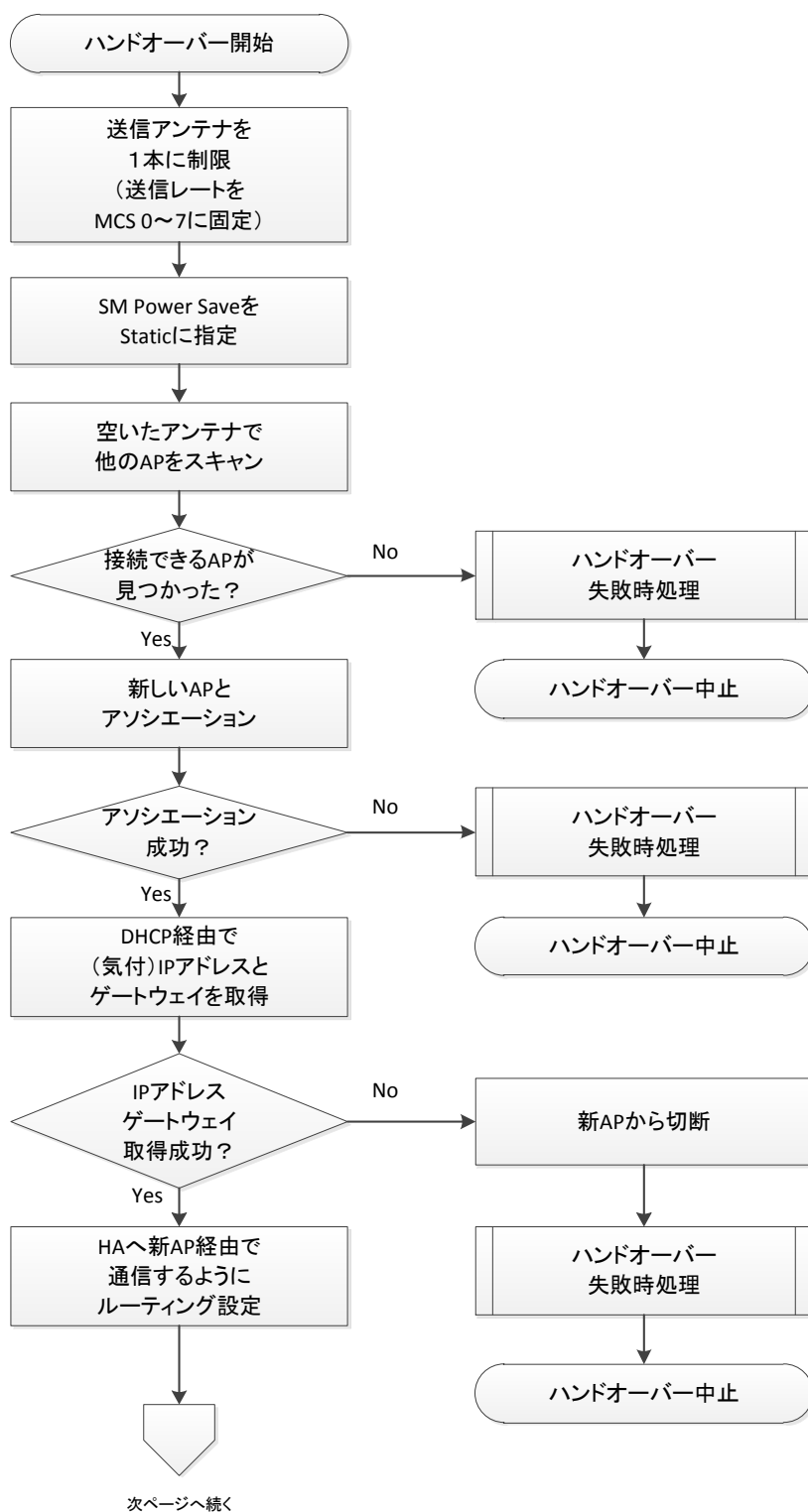


図 26： 物理層・IP 層両方のハンドオーバーの流れ図（1）

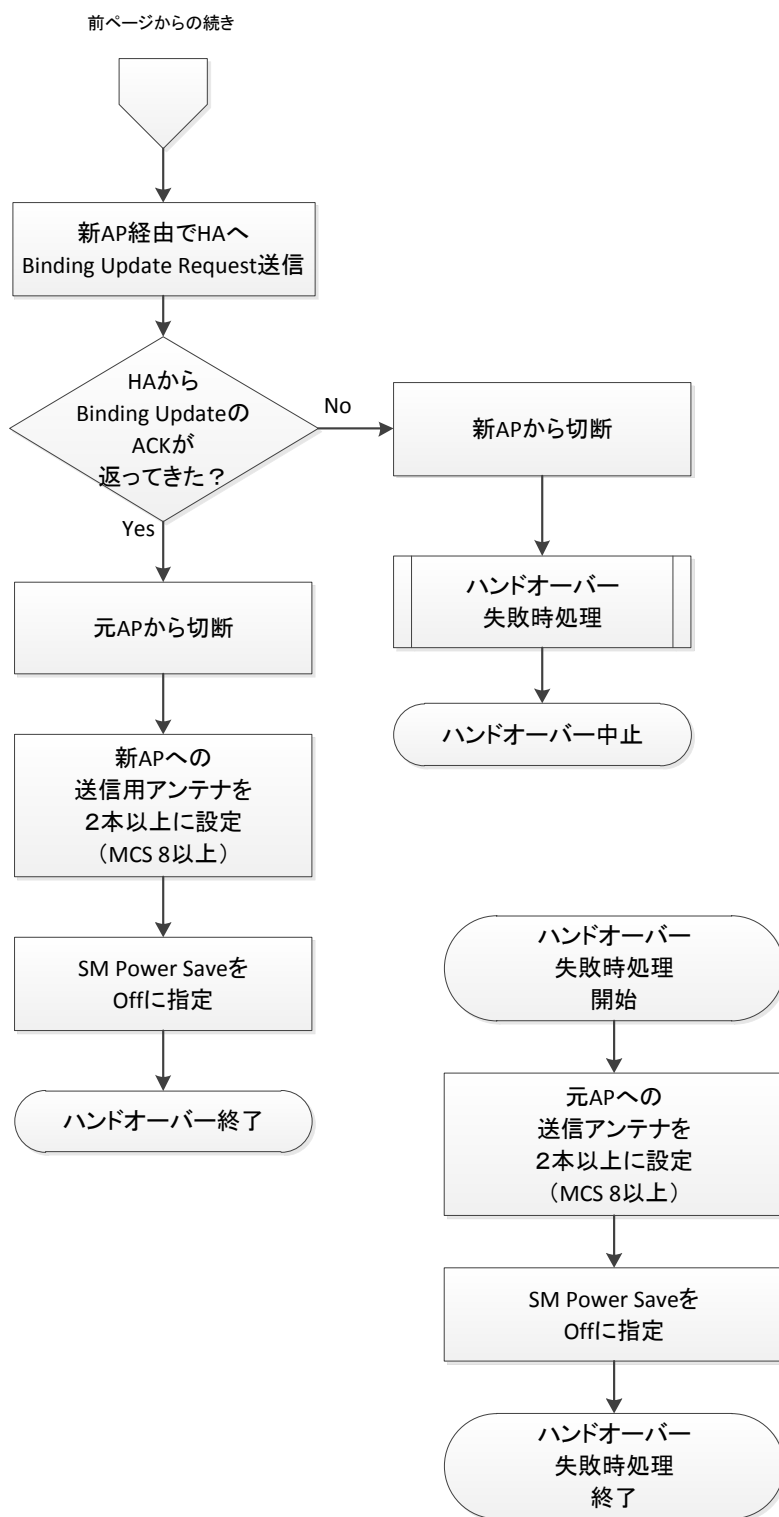


図 27： 物理層・IP 層両方のハンドオーバーの流れ図（2）

### 4.2.3.2 DHCP による IP アドレスの取得

今回の実験では 4.2.2 節で述べたように Mobile Node は AP とアソシエーションしたあとに DHCP で IP アドレスを取得することになっているが、このために DHCP クライアントの `dhclient` (version 4.1) を用いることにした。次のコマンドで実行できる。

```
# dhclient [dev]
```

ここで、`[dev]` は IP アドレスを設定するデバイス名 (`wlan1` など) である。

`dhclient` はあくまでも IP アドレスを取得するだけであり、デフォルトでは実際にその IP アドレスを検査したりデバイスに設定したりするために、シェルスクリプトである `dhclient-script` が実行される。さらにこのスクリプト内から、所定のディレクトリ内に置かれた他のシェルスクリプトが実行されるようになっている。

ハンドオーバー手順 (図 26) において、無線 LAN デバイスに付与する IP アドレスと Home Agent へのゲートウェイを DHCP 経由で取得する必要があるが、`dhclient` 自体は標準出力などに取得した情報を書き出す仕組みにはなっていない。よってハンドオーバー制御プログラムで IP アドレスやゲートウェイを把握することができない。そこで `dhclient-script` から実行できる情報取得スクリプトを準備した。今回は標準出力に DHCP で得られた情報を書き出す仕組みにし、それをハンドオーバー制御プログラムから取得するようにした。

## 4.2.4 物理層ハンドオーバーの実装

### 4.2.4.1 ドライバの動作の制御方法

物理層ハンドオーバーを行うためには、3.5 節で述べたように Mobile Node において送受信に用いるアンテナ数を制御する必要がある。しかし今回の実験では、4.1 節のソフトウェア無線機と異なり市販の一般的な無線 LAN モジュールを用いているため、通常操作では細かい制御はできない。例えば、通常無線 LAN 設定に用いられる `iw` コマンド (今回用いたのは version 3.7) には「SM Power Save のオン・オフ」機能は実装されていないし、「MCS Index による送信レート設定」機能も (実装はされているが対応が不完全なため) 働かない。

従って無線 LAN モジュール用のドライバに外部から直接働きかける必要があるが、このためにデバッグ用の `debugfs` を用いることにした。`debugfs` は Linux Kernel でよく用いられるデバッグ用のインターフェースであり、特定のファイルに対して値を読み書きすることでユーザスペースから内部プログラムの動作を確認・制御するためのものである。

`debugfs` を有効にするには、ドライバの含まれる Linux Kernel をコンパイルする際に指定する必要がある。そこで Linux Kernel version 3.6.4 について、今回用いる Intel 製無線

LAN モジュールのためのドライバ類である `iwlwifi` の `debugfs` を有効にして、さらにその他制御に必要な部分を設定し、コンパイルしなおして Ubuntu 12.04 上の Linux Kernel と差し替えて用いることにした。なお、送信レートを `debugfs` 経由で手動設定するとレートの取得時に誤った値が返されることがあったので、ドライバの該当箇所を修正した上で用いた。なお、この物理層ハンドオーバープログラムは、4.2.3 節で述べた Python 製ハンドオーバー制御プログラムに組み込む形で実装した。

### 4.2.4.2 SM Power Save および送信レート設定

今回用いる Intel 製無線 LAN モジュールは `iwlwifi` というドライバで制御されるが、このドライバの場合は `debugfs` 経由で SM Power Save の有効・無効の切り替えと、送信時レートを設定できる。

具体的には、AP 側に受信アンテナの減少を通知するための Static SM Power Save を有効にするためには “static”，解除するためには “off” を次のようにしてファイルに書き込む。なお、改行文字も一緒に書き込むとエラーになるので、`-n` オプションをつける必要がある。

このファイルに値を書き込むとバッファ内のデータの転送が終わり次第すぐに SM Power Save を AP にリクエストする (図 9)。

```
# echo -n [static/off] > /sys/kernel/debug/ieee80211/[phy]/netdev:[dev]/smps
```

ここで、`[phy]` は `iw` コマンドで用いられるインターフェース名 (`phy0` など) であり、`[dev]` はデバイス名 (`wlan1` など) である。

また、ハンドオーバーは本来1つの無線 LAN モジュールで行うことを想定しているため、受信のみならず送信に用いるアンテナも減らす必要がある。

送信時レートは次のようにして設定できる。

```
# echo -n [flag] >
  /sys/kernel/debug/ieee80211/[phy]/netdev:[dev]/stations/[bssid]/rate_scale_table
```

ここで、`[bssid]` は無線 LAN AP の BSSID (MAC アドレス) を指す。このファイルは AP と正しく接続した後に出現する。また、`[flag]` は使用アンテナや符号化方式を示す 16 進数値であり、MCS Index に対応する値を設定する。例えば、MCS Index 15 で表される符号化方式は表 1 より 64-QAM (符号化率 5/6)、空間ストリーム数が 2 [9] であり、これに対

応する値は 0x1810F である<sup>4</sup>。

従って、プログラムの実装に際しては次のことに注意した。

- SM Power Save は AP に関係なく設定できる
- 送信レートは AP 接続後に AP ごとに設定する必要がある
- 送信レートの設定には、iwlwifi のソースコードから得られたフラグ値の計算が必要である

### 4.2.4.3 AP とのアソシエーション制御

11n では WPA2 などの暗号化通信を用いることが標準的になっている。今回の実験では、Linux で無線 LAN の (WEP を除く) 暗号化通信をする際に標準的に用いられる wpa\_supplicant (version 0.7.3) を用いることにした。

wpa\_supplicant では、複数のデバイスを同時に扱うことができる。また、動作をプログラム外から制御するために、デバイスごとに Unix ドメインソケット等のインターフェースが実装されている。例えば、AP とアソシエーションを行う場合、次の手順を踏めば良い。

1. /var/run/wpa\_supplicant/[dev] を Unix ドメインソケット (SOCK\_DGRAM) として開く
2. “SELECT\_NETWORK [Network ID]” を送信する

ここで、[dev]はデバイス名、[Network ID]は設定ファイルで定義した AP を表す ID である。

同様に、AP を探索するには “SCAN”，まわりの AP 一覧を取得するには “SCAN\_RESULTS” などのようにコマンドを送信すれば良い。

接続先 AP を決めるためには様々な指標があり、例えば筆者の所属する研究室における研究として [22]が挙げられる。ただし、今回は VoIP に限らず IEEE 802.11n の性質・仕様に基づいたハンドオーバー手法がテーマであるため、簡易的に RSSI の大きな AP を選択する

---

<sup>4</sup> MCS 15 に対応する値は「HT フラグ 0x100 + 空間ストリーム数 2 の MIMO フラグ 0x8 + 64-QAM (符号化率 5/6) のフラグ 0x7」に使用アンテナのフラグを加えたものである。Intel Centrino Advanced-N 6200 ではアンテナ B とアンテナ C の 2 つを搭載しているため、「アンテナ B のフラグ 0x08000 + アンテナ C のフラグ 0x10000」を加えたものが設定すべき値 (0x1810F) である。

ことにした。

### 4.2.5 IP 層ハンドオーバーの実装

IP 層ハンドオーバーは Mobile IP または Mobile IPv6 を用いる。ただし、いずれも実装例がほとんど見つからず、Mobile IP については正常に動作させられる実装を見つけることができなかった。また、Mobile IPv6 については WIDE の USAGI プロジェクトの実装を見つけることができたが、“切断を検知してからハンドオーバーを開始する”という実装になっていたため、今回の提案手法の動作実験で用いることはできなかった。

そこで、Mobile IP の概念に基づいて簡易的な IP 層ハンドオーバーのプログラムを開発することにした。

#### 4.2.5.1 tun/tap の利用

tun/tap は Linux における仮想のイーサネットデバイスである。ユーザスペースからは普通の物理イーサネットデバイスに見えるようになっており、IP アドレスなどをもち、通信デバイスとしてそのまま指定することができる。

また、制御プログラムを組めば、そこからキャラクタデバイスとして読み書きができる。tun/tap に対して read すると tun/tap に向けて送信されたフレームやパケットが得られ、write すると tun/tap からそれを bind しているプログラムに対してフレームやパケットが送出される。すなわち Mobile Node と Home Agent の両方で tun/tap に到達したフレームやパケットを制御プログラムで読み出し、物理イーサネットデバイス経由で相手方に転送することで、Mobile Node と Home Agent 間でトンネルを構築することができる。

tun と tap の 2 種類あり、tap ではイーサネットフレーム全体、tun ではそのうちの IP パケット部分を操作することができる。今回は IP 層ハンドオーバーのためのプログラムを開発するので、tun を用いることにした。

#### 4.2.5.2 要件定義と動作の仕組み

プログラムの開発にあたり次のように要件を定義した。

- 高速動作が可能
  - 一般に VoIP 通信においては 50ms 程度までの遅延が許容されると言われているため、IP 層のトンネリングによる遅延をせいぜい 20ms 程度にまでおさえる必要がある。
- プログラム外部から制御できる仕組みの実装
  - 物理層ハンドオーバーのタイミングで IP 層のハンドオーバーを行う必要があるため。
- 複数インターフェースの同時使用



- ハンドオーバー前と後では Mobile Node からパケットを送受信する時に用いるインターフェースが異なるため。

また、プログラムの仕組みは次の通りである。

- Mobile Node 側

- ハンドオーバーのプログラムは/dev/tun を用意する
- 通信を発生させるプログラムは/dev/tun を bind する
- Home Agent への転送手順
  - ◇ ハンドオーバーのプログラムは tun 経由で IP パケットを受けとる
  - ◇ 受け取った IP パケットを UDP パケットの Payload にそのまま書き込む(tun に届いた IP パケットを UDP でカプセル化)
  - ◇ UDP パケットを物理デバイス (wlan1 等) 経由で Home Agent に送信する
- Home Agent からの受信手順
  - ◇ ハンドオーバーのプログラムは物理デバイス経由で UDP パケットを受けとる
  - ◇ 受け取った UDP パケットの Payload (転送されてきた IP パケットが含まれる) を取り出す
  - ◇ 取り出した IP パケットを tun に書き込む

- Home Agent 側

- ハンドオーバーのプログラムは/dev/tun を用意する (つまり通信を発生させるプログラムが/dev/tun を bind する)
- \*適切にルーティングテーブルを設定することで, Mobile Node へ届けるべきパケットが Linux のルーティング機能により tun に到達する
- Mobile Node への転送手順
  - ◇ ハンドオーバーのプログラムは tun 経由で IP パケットを受けとる
  - ◇ 受け取った IP パケットを UDP パケットの Payload にそのまま書き込む(tun に届いた IP パケットを UDP でカプセル化)
  - ◇ UDP パケットを物理デバイス (eth1 等) 経由で Mobile Node に送信する
- Mobile Node からの受信手順
  - ◇ ハンドオーバーのプログラムは物理デバイス経由で UDP パケットを受けとる
  - ◇ 受け取った UDP パケットの Payload (転送されてきた IP パケットが含まれる) を取り出す
  - ◇ 取り出した IP パケットを tun に書き込む

ここで、\*下線を引いた点について、Correspondent Node から届いたパケットは Mobile Node へ転送するために Home Agent に到達しなければならない。図 25 に示したように Home Agent の IP アドレスは 192.168.100.1/24 であるから、ルータのルーティングテーブルにおいて 192.168.10.0/24 宛てのパケットを 192.168.100.1 に転送するように設定する必要がある。このルータは Mobile Node のホームとなるネットワークに配置されているものであるから、実際の環境下においてもネットワーク管理者の権限が及ぶものであり、問題はない。

また、Home Agent において Mobile Node に届けるべきパケットは tun に到達する必要があるが、tun は仮想デバイスであるため、実際の経路としては Home Agent にパケットが到達するために物理デバイス (eth1 等) を経由しなければならない。このためには Home Agent 内で物理デバイスと tun の間でパケット転送が行われる必要がある。このパケット転送は適切に Linux 側で設定しておくことで自動的に行われる。具体的には次のように設定しておく。

- tun の IP アドレスとして 192.168.100.1 を設定する
- ルーティングテーブルにおいて、192.168.100.0/24 宛てのパケットがデバイス tun に到達するように設定する
- IP パケット転送機能を有効にする
- (パケットが正常に転送されない場合は) IP アドレスによるパケットフィルタを無効にする

なお、IPv4 パケット転送機能は次のコマンドで有効にできる。

```
# sysctl -w net.ipv4.ip_forward=1
```

また、ネットワークの構成によっては、tun の IP アドレスに向けたパケットが物理デバイス (異なる IP アドレスをもつ) 経由で届く際に、IP アドレスの差異を理由にフィルタでパケットが破棄されることがある。この IPv4 アドレスによるパケットフィルタは次のコマンドで無効にできる。

```
# sysctl -w net.ipv4.conf.all.rp_filter=0
# sysctl -w net.ipv4.conf.default.rp_filter=0
# sysctl -w net.ipv4.conf.[dev].rp_filter=0
```

ここで、[dev]は物理デバイス名 (eth1 など) である。

### 4.2.5.3 複数インターフェースの同時制御

複数のインターフェースを扱うプログラムは大きく分けて表 6 に示す 3 通りが考えられるが、一般に複数のインターフェースを用いた簡単なプログラムは `select` 方式で開発される。ここで `select` 方式とは、インターフェースにパケットが到達するまで待機し、到達したらそのインターフェースからパケットを取り出す仕組みのことである。

IP 層ハンドオーバーは 4.2.3 節や 4.2.4 節の物理層ハンドオーバーの制御と異なり、IP パケットの転送機能を含める必要があるため、4.2.5.2 節で要件定義したように高速動作が求められる。そこでプログラムを C 言語で開発することにした。なお、コンパイルは Home Agent の環境下で `gcc 4.6.3` を用いた。C 言語を用いることで高速動作が期待できるだけでなく、Linux 以外の環境にも移植しやすいという利点を享受できる。

今回必要な IP 層ハンドオーバーはパケットを転送するだけの簡単なものであるが、将来的に UNIX 系列以外の環境下でも動作させることを考えて、まずは `select` 方式で開発した。

ところが、`select` 方式で開発したプログラムのパケット処理時間を測定した結果、1 パケットあたりおよそ 40 ms かかっていることが分かった。これは要件の「1 パケットあたりの遅延が 20 ms まで」を満たさないため、今回の実験で用いることはできなかった。

そこで、`thread` 方式でプログラムを開発しなおした。`thread` 方式とは 1 つのプログラム（プロセス）内で複数の関数を同時に実行するための仕組みであり、プログラム内の変数設定とスレッドのタイミング制御を綿密に行う必要がある。

`thread` 方式でプログラムを開発し、Ping を 10 回 Mobile Node から Home Agent を介して Correspondent Node に向けて実行したときの RTT を計測した。あわせて、Mobile Node から直接 Correspondent Node に向けたときの RTT も計測した。これを表 7、図 28 に示す。この時間の差異が、IP 層ハンドオーバーを行うための仕組みによる遅延を表す。`thread` 方式の場合遅延時間がわずか 0.7 ms に収まっており、今回の実験で十分用いることができると言える。

表 6： 複数インターフェースを用いるプログラムの方式

	長所	短所
fork 方式	●開発が容易	●UNIX 系列でしか動作しない ●並列処理用のメモリ確保に時間がかかる
select 方式	●開発が容易 ●並列処理の同期を取りやすい ●UNIX 以外でも動作	●インターフェースの切り替えに時間がかかる
thread 方式	●UNIX 以外でも動作	●並列処理の同期を取るのが難しい ●デバッグが難しい

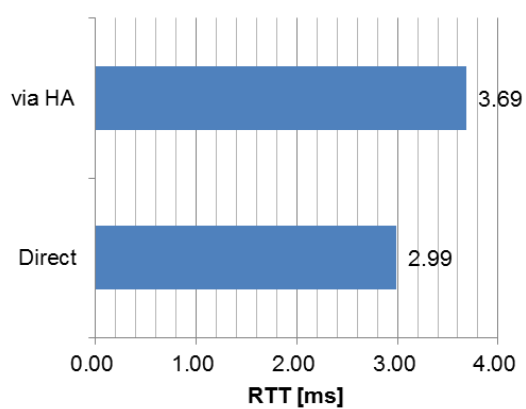


図 28： thread 方式による Home Agent を経由したときと直接通信した時の Ping による RTT の比較

表 7： thread 方式による Home Agent を経由したときと直接通信した時の Ping による RTT の比較

	Direct	via HA
RTT (ms)	2.90	3.07
	2.89	5.50
	2.89	3.10
	2.88	4.02
	2.87	3.07
	2.87	3.04
	2.88	3.05
	2.87	3.05
	2.86	5.92
	3.98	3.03
Average (ms)	2.99	3.69

なお表 7 において Home Agent を介した場合、一度だけ RTT が 5.92ms となっており、他と比べて大きな値になっている。これは Home Agent 側で画面の再描画などの割り込み処理が発生したことによるものと考えられ、このことから以降の実験においてパケットごとに処理時間の揺らぎが発生することが推測される。

また、要件で定義したように「プログラム外部から制御できる仕組み」が必要であるが、4.2.4.3 節で紹介した wpa\_supplicant と同様に、Unix ドメインソケット経由で制御できるように開発した。

### 4.2.5.4 下り方向の通信への対応

Mobile Node から Correspondent Node への上り方向の通信においては、単純に Home Agent は Mobile Node から受信したパケットを Correspondent Node へ転送すれば良い。しかし、Correspondent Node から Mobile Node への下り方向の通信においては、Home Agent は適切に Mobile Node の所属するネットワークへパケットを転送しなければならない。そこでネットワークを移動した際には、Mobile IP では Registration Request [23]、Mobile IPv6 では Binding Update [24]によって Home Agent に Mobile Node の IP アドレス（これを Care-of Address（気付アドレス）と呼ぶ）を通知することになっている。

今回実装する簡易 IP 層ハンドオーバープログラムにおいても下り方向の通信をサポートするため、これにあたる仕組みを実装した。Mobile IP では UDP パケットで送られ、Mobile IPv6 では専用のヘッダに格納されて送られることになっているが、今回は簡単のため UDP パケットとして送ることにした。ただし、この場合 Mobile Node または Correspondent Node から到達する転送すべき IP パケットも UDP でカプセル化しているためにこれらと混ざってしまう。そこで IP パケットは先頭の 4bit が 0x4（IPv4）または 0x6（IPv6）であることに着目し、ここで区別が付くように制御パケットを図 29 のように定義した。

この定義はプログラム開発にあたって IP パケットの転送と同じ機構で処理できるように IP パケットに模した形にしてある。また、将来的に気付アドレスの通知以外の制御メッセージにも利用できるように柔軟性を持たせた構造にしてある。なお、図 29 自体をペイロードとして UDP で送るため、中継経路上では単なる UDP パケットとして扱われる。

気付けアドレスの通知の際には、Mobile Node から Home Agent へ向けて図 29 について次のような制御パケットを送ることにした。

- COMMAND は 0x43, 0x54, 0x52, 0x4c
- Payload は 0x42, 0x55, 0x00, 0x00

また、それに対する応答は、Home Agent から Mobile Node へ向けて次のような制御パケットを送ることにした。

- COMMAND は 0x43, 0x54, 0x52, 0x4c
- Payload は 0x42, 0x55, 0x00, 0x06

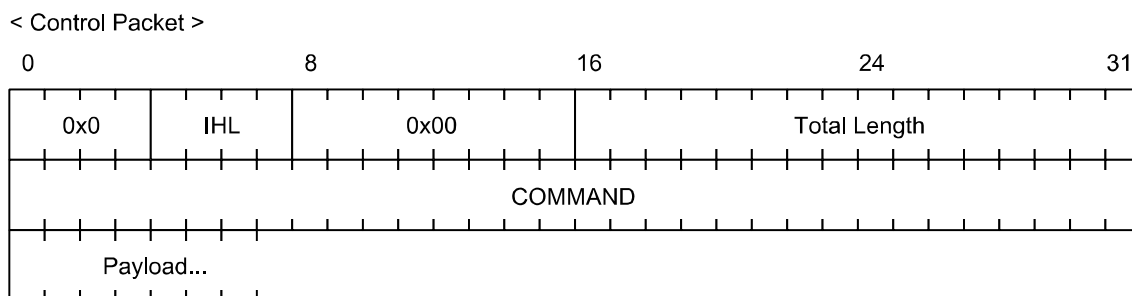


図 29： IP 層ハンドオーバープログラムの制御パケット  
IHL は IP ヘッダのそれと同じ形式。

#### 4.2.6 物理層・IP 層両方のハンドオーバーの実験と評価

4.2.4 節, 4.2.5 節で説明したように, 物理層・IP 層両方においてハンドオーバーするためのプログラムを開発した。電波暗室内に図 30 のように機器を配置し, 図 16 のパケットを 20ms 間隔で Home Agent, Correspondent Node 間で送受信する実験を行った。

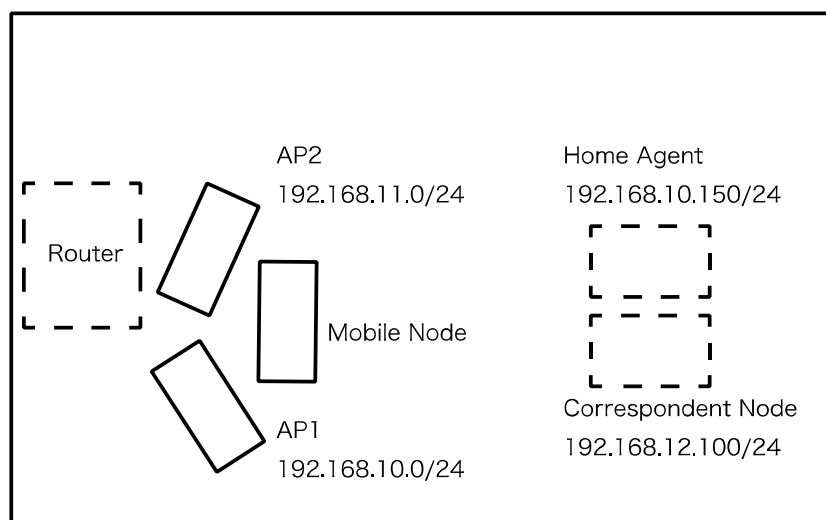


図 30： 電波暗室内での AP 類の配置

##### 4.2.6.1 Mobile Node から Correspondent Node へ向けた上りの UDP パケット送信実験

まずは, Mobile Node から Correspondent Node へ向けて, 図 16 のパケットを 20ms 間隔で送信した。それと同時に, 開発したプログラムで物理層・IP 層ハンドオーバーを行い, パケットの欠落およびパケット到達時間の揺らぎの計測を行った。ここで, MIMO 時の MCS Index は最大 15, SISO 時は最大 7 であるが, 最大値に設定すると非常に理想的な環境下でないとい雑音などの影響により受信側で復号できずフレーム欠損につながるため,

実運用時には Mobile Node と AP が離れていることを踏まえて、MIMO 通信時 10 (2 ストリーム, QPSK, 符号化率 3/4), SISO 時は 5 (1 ストリーム, 16-QAM, 符号化率 2/3) に設定した。この結果を図 31, 図 32 に示す。なお,  $n$  番目のパケット処理時間の揺らぎ  $(T_{\text{delay}})_n$  は次の式で定義する。

$$(T_{\text{delay}})_n = T_n - T_0 \quad (2)$$

$$T_n = (T_{\text{sent}})_n - (T_{\text{received}})_n$$

ただし,  $T_{\text{sent}}$  はパケットを送信した時の送信側の時刻であり,  $T_{\text{received}}$  はパケットを受信したときの受信側の時刻である。

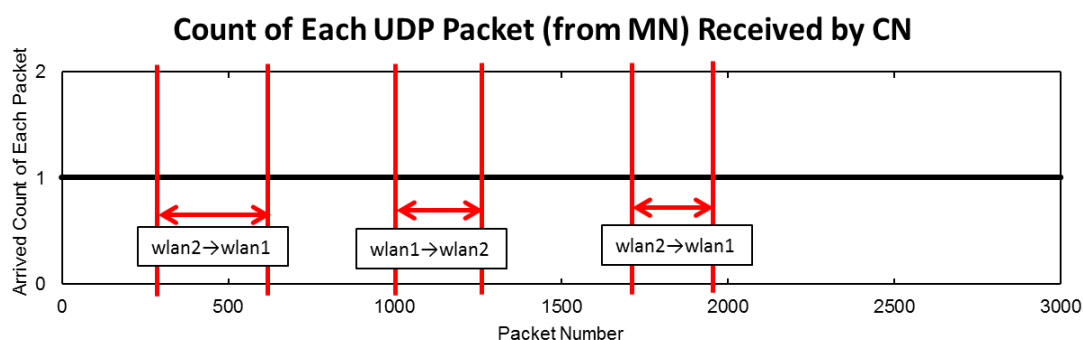


図 31 : Correspondent Node で受信された Mobile Node からのテストパケットの数  
赤の縦線の間でハンドオーバーを行った。縦軸は、横軸に示す番号を振った各パケットの到着個数を示し、0 はパケット損失、1 はパケット到達、2 は duplicate 発生を示す、

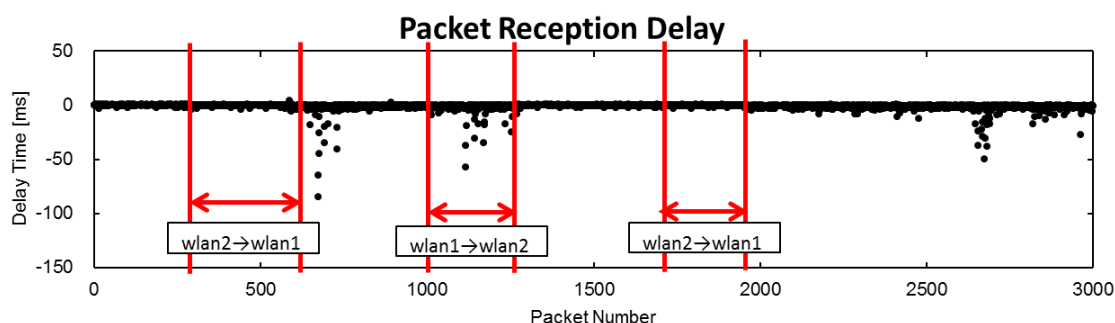


図 32 : Correspondent Node で受信された Mobile Node からのテストパケット処理時間の揺らぎ  
赤の縦線の間でハンドオーバーを行った。

この結果から分かるように、20 ms 間隔でパケットを送信している最中にハンドオーバーを行っても、パケットの欠落は一切起きていない。また、パケット処理時間の揺らぎは最大で 100 ms 程度であるが、これはハンドオーバーとは関係のないところで発生している。さらに、2 回目のバンドオーバーのところでは若干のパケット処理時間の揺らぎが見られるが、最大でもおよそ 50 ms であり、4.2.5.3 節で説明したプログラムによる遅延とあわせても十分実用に耐えうる結果が得られたといえる。

なお、グラフに示すことはできなかったが、パケットの到達順序の入れ替わりは発生しなかった。

また、ハンドオーバーにかかった時間とその内訳を表 8、図 35 に示す。

ハンドオーバーにかかる時間は AP や MN の性能・電波状況などに応じて異なるものの、今回の実験環境では 6 秒程度であった。従って、「ハンドオーバーを開始」してから「移動元 AP からの電波が到達しなくなる」までの時間を 6 秒以上稼げれば良いことになる。

また、時間がかかるのは AP の探索とアソシエーションである。従って、ハンドオーバーを開始するまでに積極的に接続可能な AP を探索することが求められ、このためには使用帯域の小さい時に SISO 通信に切り替えて AP 探索を背後で行う等の工夫があればなお良い。

### 4.2.6.2 Correspondent Node から Mobile Node へ向けた下りの UDP パケット送信実験

次に、Correspondent Node から Mobile Node へ向けて、図 16 のパケットを 20ms 間隔で送信した。それと同時に、開発したプログラムで物理層・IP 層ハンドオーバーを行い、パケットの欠落およびパケット到達時間の揺らぎの計測を行った。ここで 4.2.6.1 節と同様に、MIMO 通信時 10 (2 ストリーム, QPSK, 符号化率 3/4), SISO 時は 5 (1 ストリーム, 16-QAM, 符号化率 2/3) に設定した。この結果を図 33, 図 34 に示す。なお、n 番目のパケット処理時間の揺らぎは 4.2.6.1 節と同様に(2)式で定義した。



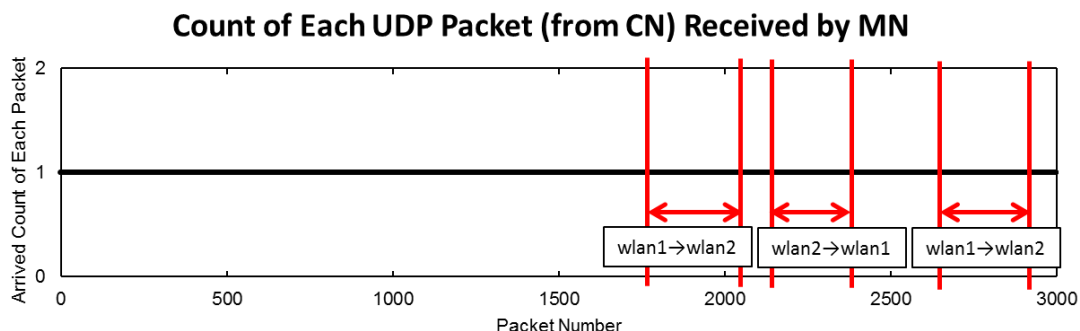


図 33： Mobile Node で受信された Correspondent Node からのテストパケットの数  
赤の縦線の間でハンドオーバーを行った。縦軸は、横軸に示す番号を振った各パケットの  
到着個数を示し、0 はパケット損失、1 はパケット到達、2 は duplicate 発生を示す、

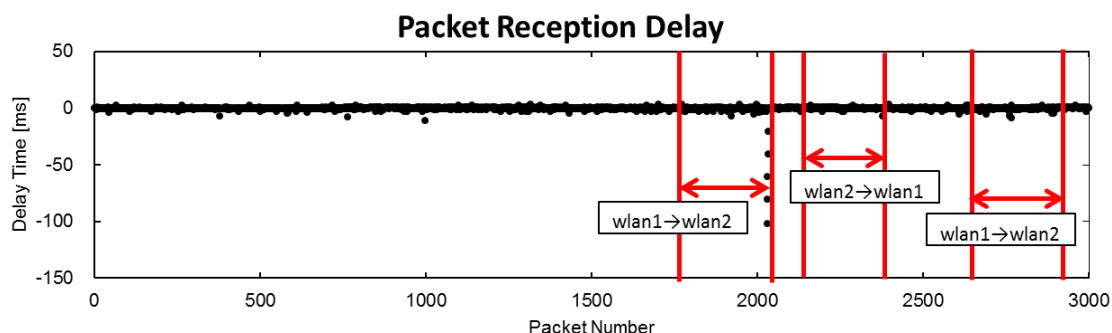


図 34： Mobile Node で受信された Correspondent Node からのテストパケット処理時間  
の揺らぎ  
赤の縦線の間でハンドオーバーを行った。

この結果から分かるように、20 ms 間隔でパケットを送信している最中にハンドオーバーを行っても、パケットの欠落は一切起きていない。また、パケット処理時間の揺らぎは最大で 100 ms 程度であるが、1 回目のハンドオーバーのところのみで発生しており、ハンドオーバーに直接関係するものではないと思われる。従って、4.2.5.3 節で説明したプログラムによる遅延とあわせても十分実用に耐えうる結果が得られたといえる。

なお、グラフに示すことはできなかったが、パケットの到達順序の入れ替わりは発生しなかった。

また、ハンドオーバーにかかった時間とその内訳を表 8、図 35 に示す。

この実験でハンドオーバーにかかった時間は、やはり 4.2.6.1 節と同様に 6 秒程度である。従って、「ハンドオーバーを開始」してから「移動元 AP からの電波が到達しなくなる」までの時間を 6 秒以上稼げれば良いことになる。

また、4.2.6.1 節と同様に時間がかかるのは AP の探索とアソシエーションである。従って、ハンドオーバーを開始するまでに積極的に接続可能な AP を探索することが求められ、このためには使用帯域の小さい時に SISO 通信に切り替えて AP 探索を背後で行う等の工夫があればなお良い。

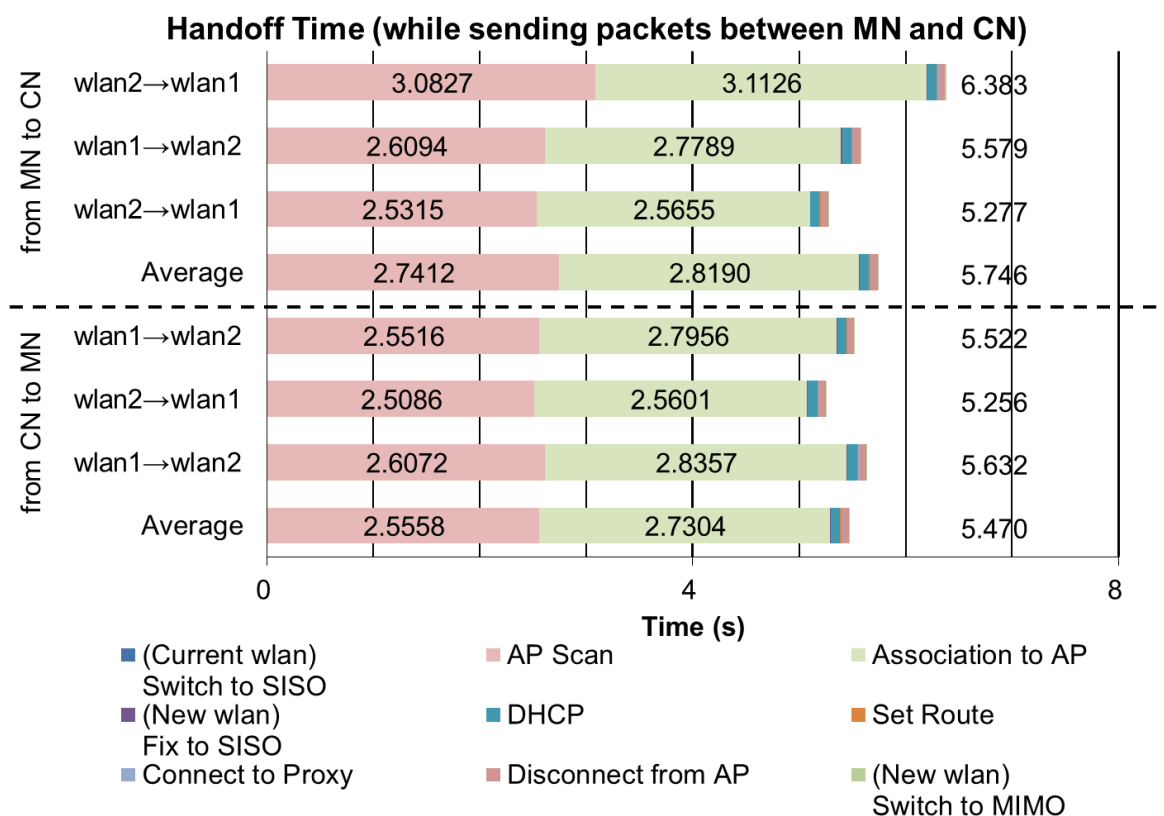


図 35 : Correspondent Node と Mobile Node 間で UDP 通信をしながらハンドオーバーを行った際の経過時間とその内訳  
具体的な数値は表 8 を参照。

表 8 : Correspondent Node と Mobile Node 間で UDP 通信をしながらハンドオーバーを行った際の経過時間とその内訳

(s)	(Current wlan) Switch to SISO	AP Scan	Association to AP	(New wlan) Fix to SISO	DHCP	Set Route	Connect to Proxy	Disconnect from AP	(New wlan) Switch to MIMO	Total Time
wlan2 →wlan1	0.0045	3.0827	3.1126	0.0049	0.0862	0.0086	0.0180	0.0547	0.0109	6.383
wlan1 →wlan2	0.0046	2.6094	2.7789	0.0094	0.0898	0.0090	0.0100	0.0628	0.0047	5.579
wlan2 →wlan1	0.0045	2.5315	2.5655	0.0049	0.0845	0.0108	0.0143	0.0564	0.0047	5.277
Average	0.0045	2.7412	2.8190	0.0064	0.0868	0.0094	0.0141	0.0580	0.0067	5.746
wlan1 →wlan2	0.0044	2.5516	2.7956	0.0051	0.0844	0.0078	0.0123	0.0564	0.0049	5.522
wlan2 →wlan1	0.0045	2.5086	2.5601	0.0052	0.0961	0.0084	0.0167	0.0513	0.0050	5.256
wlan1 →wlan2	0.0045	2.6072	2.8357	0.0051	0.0913	0.0085	0.0166	0.0583	0.0050	5.632
Average	0.0045	2.5558	2.7304	0.0051	0.0906	0.0082	0.0152	0.0553	0.0050	5.470

# 第5章 考察と議論

---

本章では、実験から得られた結果を基に提案手法が実環境下においても有効であることを述べる。また、今回の実験では 2x2 MIMO の場合のみを考えたが、3x3 MIMO の場合に 11n の仕様に加えるべき修正内容を述べる。さらに、提案手法のために Mobile IP や Mobile IPv6 の実装について考慮すべき点を整理し、経路の暗号化について既存手法の活用を提案する。

### 5.1 実験結果に基づく考察

4.2.6 節で述べた実験結果より、実環境下においても IEEE 802.11n の複数アンテナを独立に制御することによって、上り・下り両方の通信ともシームレスなハンドオーバーが可能になることが確かめられた。また、ハンドオーバーにかかる時間は環境や機器性能によって異なるものの、今回の構成におけるプログラムや AP、PC 等の場合およそ 6 秒程度ですむことが分かった。歩いて移動する場合を考えれば 6 秒という値は現実的であるように思われる。ただし [25]などの他の文献を参照すればアソシエーションは数百 ms 程度で完了するように報告されているため、この時間をさらに短縮することが期待される。

実験ではハンドオーバーを開始する際のトリガーについて触れなかったが、実際には RSSI や通信速度をトリガーとして用いることができる。また、IEEE 802.11n の場合は電波品質に応じて MCS Index を随時変化させ、MIMO 通信や SISO 通信（ハンドオーバーのためではなく通信品質を確保するためである）を動的に切り替えながら電波を送受信している。MCS Index は大まかに見れば数字が小さいほど雑音耐性があがるため、MCS Index の値の変化をトリガーにしても良い。

ビル内などに AP を配置する際には、一般的にはサイトサーベイを行って電波がつかめるように配慮する [2]。ここで、提案手法によるハンドオーバーを意識する場合、ある AP と他の AP のそれぞれの通信可能範囲について重複する距離が重要である。これは、重複している範囲にいる間にハンドオーバーをすませる必要があるからである。そこで、4.2.6.1 節の結果と 4.2.6.2 節の結果から、それらの平均である 5.61 秒をハンドオーバーにかかる時間として、重複距離と端末の移動可能な最大速度の関係を図 36 に示す。

提案手法は無線 LAN の AP が設置されているビルなど屋内での使用を想定しているため、人の歩行速度程度で端末が移動できれば十分である。人の歩行速度をおよそ 1 m/s とすると約 5.7 m 程度の重複距離があれば良い。一般的な無線 LAN の AP の電波が届く範囲を、(障害物のある)一般家庭において 1 台の AP で 1 戸をまかなえていることを踏まえて半径 50m 程度と考えれば、実現に際して十分問題ない値であると言える。

さて、提案手法が屋内においては十分適用可能であることが分かったが、屋外での使用についても考察する。

高速で移動するには重複距離を大きく設ければよいが、もちろんそれだけたくさんの AP が必要になる。例えば 50 m ずつ重複範囲を設ける場合には、移動可能速度は約 8.91 m/s (32 km/h) であり、原付程度の速度でしか移動できない。

また、電波の輻輳などを無視したとして AP を細かく並べた場合であっても、そもそも AP から 100 m しか通信できないので、重複距離は最大 100 m 程度になる。すなわちこのときが理論上の最大移動可能速度であり、約 17.8 m/s (84 km/h) である。よって車の速度程度が限界である。

従って今回得られた結果を基にすれば、提案手法は人が端末を持って建物内を移動する際には十分威力を発揮するが、例えば鉄道の脇に AP を設置して公衆無線 LAN サービスを行うといった応用は難しいといえる。ただし、今回の結果は AP とのアソシエーションにかかる時間が他の文献の報告に比べて長いので、時間を短縮することで高速移動体についても応用の余地はあると考えられる。

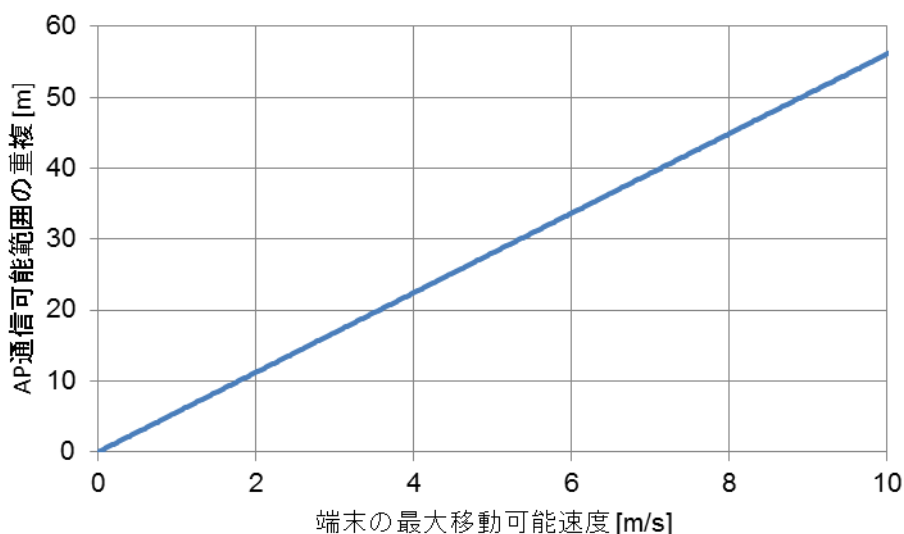


図 36： 端末の最大移動可能速度と AP 通信可能範囲の重複距離の関係

## 5.2 アンテナが 3 本以上ある場合の通信用アンテナ増減方法

本論文の実験では無線 LAN モジュールにアンテナが 2 本しか搭載されていない場合を考えたが、実際には IEEE 802.11n では 4 ストリーム (アンテナ 4 本) まで定義されている [9]。しかし SM Power Save はストリーム数を 1 に減らすことしかできないため、ハンドオーバー中の通信はアンテナ 1 本のときの速度に制限されてしまう。そこで、SM Power Save の



ンネルを次々に切り替えることによる時間」だと考えれば、複数の RF 回路で複数のチャンネルを同時にスキャンすることで、大幅な時間短縮が期待される。また、比較的大きな通信帯域が必要な環境下では、通信に用いるアンテナを最大限確保して、時間がかかっても 1 本のアンテナで移動先 AP をスキャンすることも考えられる。いずれにしても、3, 4 ストリーム MIMO 通信が可能な 11n 端末を最大限活用するためには、複数アンテナ・RF 回路の分離を自由に制御できることも求められる。

### 5.3 提案手法に適切な Mobile IP / Mobile IPv6 の実装と経路暗号化

本論文の実験では、提案手法に用いることができる Mobile IP / Mobile IPv6 の実装を見つけることができなかつたため独自に開発した。しかしこれは Mobile IP / Mobile IPv6 の RFC に抗ったものではなく、単に実装の問題である。そこで次のように提案する。

まず、Mobile IP の RFC [23]ではハンドオーバーのトリガーについて触れられていない。そこで、「Mobile Node が 2 つ以上のインターフェースを持っている場合には、移動先ネットワークから Registration Request が送信されてくる直前までは移動元ネットワークに Mobile Node がいるものとする」という記述を追加する。また、Mobile IPv6 [24]では 1 つのデバイスが 2 つ以上の IPv6 アドレスを持っている場合について触れられているので、その近辺に同様の記述を追加する。

さらに、本論文の実験では Home Agent において Mobile Node の認証は一切行っていないが、もし第三者が Home Agent に対して接続すると通信が傍受される恐れがある。また、経路上では UDP で単純にカプセル化しただけであるため、盗聴の恐れがある。

この 2 つの問題を解決するために、一般的な VPN による Home Agent と Mobile Node 間への経路の設定を提案する。例えば OpenVPN [26]では公開鍵暗号方式で認証しトンネルを張るが、VPN 上のパケットは UDP パケットとしてトンネリングしているため、本論文で用意した IP 層ハンドオーバープログラム上で動作させることができる。ただし、認証に時間がかかるため、端末の移動可能速度が小さくなる恐れがある。もしくは、本論文では行わなかったが、OpenVPN のソースコードに手を加えて、本論文で準備した IP 層ハンドオーバープログラムを組み込むことも考えられる。



# 第6章 結論

---

### 6.1 結論

本論文では無線 LAN の AP が設置されているビルなど屋内環境において、AP を切り替える際にパケットが欠落せず、極端な遅延も発生しないシームレスなハンドオーバーについて新手法を提案した。この新手法はすでに設置されたネットワークの構成にほとんど手を加えることなくハンドオーバーを実現するため、既存手法に比べて導入が容易であると言える。

まず、ソフトウェア無線機を用いて物理層におけるシームレスなハンドオーバーの実験を行った。これにより、提案手法の性能の可能性を確かめた。

さらに、実環境下での応用を踏まえた実験のために、物理層ハンドオーバーのみならず、IP 層のハンドオーバーについてもプログラムを開発した。このうち、物理層ハンドオーバーは提案手法の実装であり、IP 層ハンドオーバーは Mobile IP / Mobile IPv6 の概念をもとにした独自実装である。

それを用いて“物理層・IP 層の両方においてハンドオーバーを実現する”という動作実験を行った。この結果、提案手法は屋内で移動端末をもって人が移動した場合でも、シームレスなハンドオーバーが実現できることを確かめた。

最後に、実験結果並びに本研究を進めるにあたって発見した問題に対し、解決手法を提案した。

### 6.2 今後の研究の指針

本論文で報告した研究について、さらなる研究の発展のために、今後の課題・研究の指針を示す。

本論文の実験ではハンドオーバーのトリガーについて触れなかったが、適切なタイミングでハンドオーバーを開始しなければ移動端末の移動速度に影響を及ぼす恐れがある。そこで、まずはこのトリガーについて検討することが研究の指針の一つである。あわせて、移動先 AP の選択方法について、今回は RSSI を頼りにしたが、AP の混雑具合によっては RSSI と通信可能帯域は必ずしも対応しない。従って、最適な AP を選択する手法を検討すること、あるいは筆者の所属する研究室でも研究されている [22] の適用についても研究のテーマとなり得る。

次に、実験では AP の検索やアソシエーションに時間がかかっているが、例えば 5.2 節で提案した手法を用いたり、プログラム構成を見直してアソシエーション時間を報告されているような時間にまで短縮したりすれば、応用範囲が屋内だけでなく高速移動体通信にも広がる。この場合ネットワーク側に加える工夫について研究するというのが指針の一つである。

また、2.2.1 節で述べたように、IEEE 802.11n では通信の効率を上げるために複数のフ

フレームをまとめて送るフレームアグリゲーションが定義されている。これによりフレームサイズが大きくなっているとハンドオーバーに非常に時間がかかってしまう恐れがある。そこで、例えば [13] を適用することなどを考えて、ハンドオーバーを行う前にフレームサイズを小さくすることも考慮できればより望ましい。

さらに、実験ではハードウェア・ソフトウェアの制約から1つの無線 LAN モジュールのアンテナを独立制御するには至らなかったが、図 14 で示したモジュール内の RF 回路、中間周波数変換の局部発信器を複数搭載した回路にすることで、ハードウェアの実現は可能であると推測される。実際にこの機器を作製し実験することができればなお良い。

## 謝辞

---

本研究を進めるにあたり、多くの方々よりご協力、ご指導をいただきました。

指導教員の相田仁教授には、2年間研究を行うにあたり最高の研究環境を与えてくださいました。研究テーマの選定にあたっては、震災による混乱の中わがままを聞いてくださいましたし、研究に行き詰まった際には適切なアドバイスをいただきました。また、実験機器の調達や実験環境で困ったときにも親身になってご面倒をみていただきました。大変お世話になり、深く御礼申し上げます。

秘書の中山早百合女史には、研究室での生活において日頃より細やかなサポートをしていただいたおかげで、この2年間を本当に楽しく過ごすことができました。また、物品調達や学会発表の際にも多大なるお時間を割いていただき、大変お世話になりました。厚く御礼申し上げます。

技術専門職員の千葉新吾氏には、研究生活を送るにあたって支障が起きないように色々ご面倒をみていただきました。また、専門分野外の最新の動向についても日頃からご教示いただき、ここに感謝申し上げます。

また、実験で用いたソフトウェア無線機について、森川博之教授にアドバイスをいただき、深く感謝いたします。さらに、電波暗室内での実験においては、廣瀬明教授や廣瀬(明)研究室の皆様にお世話になりました。心より感謝申し上げます。

同期として頑張ってきた田中貴大氏は、慣れない東京へやってきた筆者に当初より友好的に接していただき、また文化の違いに刺激を受けました。お互い助け合い、切磋琢磨しあうことで充実した研究生活を過ごせたことをうれしく思います。

また、2011年4月から2年間、研究にイベントにと様々な面で研究生活を支えてくださった研究室の先輩方、後輩の皆様にも深く感謝しております。

最後に、この場に至るまで多大なる可能性とチャンスを与えてくれた家族に心より感謝いたします。

2013年2月6日

## 参考文献

---

- [1] Kiran Ahuja, Rajesh Khanna, and Brahmjit Singh, "Available bandwidth estimation: A criterion for network selection in 3G and WLAN heterogeneous environment," *International Journal of Research and Reviews in Computer Science*, vol. 2, no. 3, pp. 889-894, June 2011.
- [2] Frank Ohrtman and Konrad Roeder, *Wi-Fi handbook: building 802.11b wireless networks*.: McGraw-Hill Professional, 2003.
- [3] 平原 正樹 et al., "無線 LAN での Make-Before-Break ハンドオーバーの他方式比較とフィールド実証," *信学技報 MoMuC*, vol. 105, no. 624, pp. 13-17, Feb. 2006.
- [4] "日経パソコン 2006/09/11 号," p. 126.
- [5] R.Chandra and P.Bahl, "MultiNet: connecting to multiple IEEE 802.11 networks using a single wireless card," *INFOCOM 2004. 23rd Annual Joint Conference of the IEEE Computer and Communications Societies*, Mar. 2004.
- [6] "IEEE Std 802.11r-2008 Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications Amendment 2: Fast Basic Service Set (BSS)," July 2008.
- [7] H. Ahmed and H.Hassanein, "A performance study of roaming in wireless local area networks based on IEEE 802.11r," *2008 24th Biennial Symposium on Communications*, pp. 253-257, June 2008.
- [8] M. Finsterbusch, T. Möbert, and P. Schmidt, "Improved Scanning Methods to speed-up IEEE 802.11 Handoff," *Journal of Computing*, vol. 2, no. 12, Dec. 2010.
- [9] "IEEE 802.11n-2009 — Amendment 5: Enhancements for Higher Throughput," Oct. 2009.
- [10] IRI, ユビキタス研究所, マスタリング *TCP/IP IPv6 編*.: オーム社, 2005, pp. 214-226.
- [11] 守倉正博, 久保田周治, *802.11 高速無線 LAN 教科書*.: インプレス R&D, 2004, p. 99.
- [12] 岩木 紗恵子, 村瀬 勉, 小口 正人, "屋外での実機を用いたマルチレート無線LANのスループット測定と解析," *電子情報通信学会 CQ 研究会 CQ2011-55*, pp. 73-78, Nov. 2011.
- [13] 杉本武之, 森隆幸, 森本陽介, 山口実靖, 浅谷耕一, "IEEE802.11n 無線 LAN における

- パケット誤り率を用いた動的レート制御に関する検討," *電子情報通信学会総合大会 B-5-177*, p. 591, Mar. 2007.
- [14] GNU Radio. [Online]. <http://gnuradio.org/>
- [15] Ettus Research, LLC. [Online]. <http://www.ettus.com/>
- [16] Prof. Robert W. Heath Jr. Hydra Multihop Testbed. [Online].  
<http://www.profheath.org/prototyping/hydra-multihop-testbed/>
- [17] K. Mandke et al., "Early Results on Hydra: A Flexible MAC/PHY Multihop Testbed," *Proc. of IEEE Vehicular Tech. Conf.*, pp. 1896-1900, Apr. 2007.
- [18] K. Mandke, R. C. Daniels, R. W. Heath Jr., and S. Nettles, "On the Challenges of Building a Multi-Antenna Software Defined Packet Radio," *Proc. of the SDR Technical Conference and Product Exposition*, Oct. 2008.
- [19] Ketan Mandke, Robert C. Daniels, Soon-Hyeok Choi, Scott M. Nettles, and Robert W. Heath Jr., "A MIMO Demonstration of Hydra," *WinTECH '07 Proceedings of the second ACM international workshop on Wireless network testbeds, experimental evaluation and characterization*, pp. 101-102, Sep. 2007.
- [20] H. Schulzrinne and S. Casner, "RTP Profile for Audio and Video Conferences with Minimal Control," RFC 3551 Standard, July 2003.
- [21] G. Schroder and M. Hashem Sherif, "The road to G.729: ITU 8-kb/s speech coding algorithm with wireline quality," *IEEE Communications Magazine*, vol. 35, no. 9, pp. 48-54, Sep. 1997.
- [22] 藤原直弘, 相田仁, "無線 LAN VoIP ハンドオーバー時における最適 AP 選択手法の検討," *2008年電子情報通信学会大会講演論文集*, p. 233, Mar. 2008.
- [23] C. Perkins Ed., "RFC5944: IP Mobility Support for IPv4, Revised," *IETF*, Nov. 2010.
- [24] C. Perkins Ed., D. Johnson, and J. Arkko, "RFC6275: Mobility Support in IPv6," *IETF*, July 2011.
- [25] K. Yamada et al., "A Communication System with a Fast Handover under a High Speed Mobile Environment," *IEEE 72nd Vehicular Technology Conference Fall (VTC 2010-Fall)*, pp. 1-5, Sep. 2010.
- [26] OpenVPN. [Online]. <http://openvpn.net/>

## 発表一覧

---

1. 出水達也, 相田仁, "IEEE 802.11n に基づくシームレスなハンドオーバー手法の提案," 2012年電子情報通信学会通信ソサイエティ大会, 通信講演論文集2, no.B-6-30, p.30, Sep. 2012.
2. 出水達也, 相田仁, "IEEE 802.11n に基づくシームレスなハンドオーバーの実験的実装と評価," 電子情報通信学会技術研究報告, NS IN, Mar. 2013. (予定)