

## 端末移動時におけるパケットロスレス ハンドオーバーの提案

金本綾子<sup>†</sup> 鈴木秀和<sup>†</sup> 渡邊晃<sup>†</sup>

<sup>†</sup>名城大学理工学研究科

モバイルコンピューティング環境では、端末が移動しても接続を切断することなく通信を継続することが要求されている。そこで、我々はエンド端末だけで移動透過性を実現できる Mobile PPC を提案している。しかしながら Mobile PPC だけでは、ハンドオーバー時に通信切断時間が生じ、パケットロスが大きく発生してしまうという課題がある。これは、次に接続するためのアクセスポイントの探索や、移動先ネットワークで IP アドレスを取得する際に通信ができない時間帯が発生するためである。本稿では、端末に 2 枚の無線 LAN インタフェースを搭載し、Mobile PPC をターゲットとして上記課題を解決する方法について述べる。提案方式を実装し評価を行った結果、パケットロスがほとんど発生しないこと、通信に与える負荷は十分に小さく、消費電力もほとんど増加せずに実現できることが確認できた。

## A proposal of a packet loss-less handover when a terminal moves

Ayako Kanemoto<sup>†</sup> Hidekazu Suzuki<sup>†</sup> Akira Watanabe<sup>†</sup>

<sup>†</sup>Graduate School of Science and Technology, Meijo University

In the mobile computing environment, it is strongly demanded that the connection is maintained even when mobile terminals change their locations during the communication. However, in TCP/IP, IP addresses change along with the movement of terminals, and communication is inevitably broken. To solve this problem, we have been studying a new technology called "Mobile PPC" that can achieve "mobility" only with end terminals. Although Mobile PPC has resolved the problem, a lot of packet loss must occur at the time of movement due to channel scanning and IP address acquiring process. In this paper, we propose a new method that can achieve packet-lossless handover in Mobile PPC.

### 1. はじめに

無線 LAN やインターネットの急速な普及により、いつでも誰でもどこからでもネットワークへのアクセスが可能なユビキタス社会を実現するために、移動しながら通信を行える環境が要求されている。しかし、インターネットでは端末が移動すると IP アドレスが変化するため、通信が継続できないという問題がある。そこで、端末の移動による IP アドレスの変化を隠蔽し、通信を継続できるようにする移動透過性の研究が盛んに行われている[1]。

IP 層で移動透過性を保証するプロトコルとして、IPv4 対応の Mobile IP[2] や、IPv6 対応の MobileIPv6[3]が IETF(Internet Engineering Task Force)で提案されている。Mobile IP は、移動ノード MN(Mobile Node)の位置を管理する HA(Home Agent)を導入し、通信相手ノード CN(Correspondent Node)が自分の通信相手が HA であるように見せかける移動透過性技術であ

る。しかし、HA という特殊なネットワーク機器の配備が必要であるほか、通信経路に冗長が発生したり、トンネル転送時に余分なヘッダが必要となったりするなどの課題があり、普及が滞っている。現状のネットワークは IPv4 が主体であることから移動透過性を早期に普及させるには IPv4 での実装が可能であることが望ましいが、IPv4 に対応した実装が存在するのは現在のところ、Mobile IP と後述する Mobile PPC (Mobile Peer to Peer Communication)のみであり、移動透過性の研究は IPv6 を前提としたものがほとんどである。しかし、IPv6 はまだ普及しておらず、仮に IPv6 が普及したとしても当分の間は IPv4 と IPv6 の共存環境になると考えられる。

そこで、我々は IPv4 対応に、エンド端末だけで移動透過性を実現できる Mobile PPC[4]の研究を行っている。Mobile PPC では DDNS(Dynamic DNS)[5]を利用して通信を開始する。通信開始後に MN が移動すると、MN から CN に対して変化情報を直接通知し、両端末

の IP 層の中に保持するアドレス変換テーブルを更新する。以後の通信は上記テーブルに基づきアドレス変換する。この方式により、IP アドレスの変化を上位ソフトウェアから隠蔽し、通信を継続することができる。Mobile PPC は既存端末と上位互換性があり、段階的な普及を望める方式である。

しかし、IPv4 の世界で移動透過性を実際に運用しようとするに極めて難しい面があることがわかった。その理由は、アクセスポイントの切り替え手順や新規に IP アドレスを取得するまでの手順が明確に定められていないことである。また、無線 LAN で通常利用されるインフラストラクチャモードは、端末が通信中に移動するような状況はほとんど考慮されておらず、迅速な AP(Access Point)の切り替えができない。Windows では、端末が通信中に異なる AP に接続すると DHCP シーケンスを開始する仕組みが標準で備わっている。しかし、Windows の仕組みと DHCP サーバの処理はうまく連携しておらず、新 IP アドレスの取得に数秒から数十秒の時間を要することがある。また、この機能が動作するには、DHCP サーバが移動先で整備されていることが前提となる。さらに、Windows 以外の OS ではこのようなしくみは備わっておらず、AP の切り替えを検出する手段を別途準備する必要がある。

以上のことから、移動透過性の機能が IP 層レベルで仮に実現できたとしても、移動時の通信断絶時間やパケットロスが大きく、そのままでは実用的ではないという課題がある。そこで、本研究では Mobile PPC をターゲットとして、ハンドオーバー時にもパケットロスを極力減らす方法を検討した。

上記のような課題を解決するため、本研究では端末に無線 LAN インタフェースカードを 2 枚搭載し、一方で通常の通信とネットワークの監視を、もう一方で IP アドレスの取得を行うことによりパケットロスがほとんど発生しないシームレスなハンドオーバーの実現を検討した。

提案方式を FreeBSD5.3 上に実装し、動作確認と性能測定を実施した結果、パケットロスがほとんどないハンドオーバーが可能であること、一般通信に与える負荷は十分に小さく、無線 LAN インタフェースが 1 枚の場合と比較しても消費電力においてほとんど影響がなく実現可能であることが分かった。本提案方式は Mobile PPC への適用時に有効な手段となりうるが、他の移動透過性技術にたいしても有効な方式であると考えられる。

以下 2 章で既存のパケットロス対策について、3 章で Mobile PPC の概要について説明し、4 章でハンドオーバーの現状について説明する。5 章では提案方式について、6 章では実装について、7 章では実装に対する評価を示し、さらに 8 章にてまとめる。

## 2. 既存のパケットロス対策

通信中の端末が移動したときのパケットロス対策を分類すると、L2L3 連携方式、L3 プロトコル拡張方式、L2 ドライバ改造方式、デュアルインタフェース方式に分類できる。以下それぞれの方式の概要と課題について述べる。

### (1)L2L3 連携方式[6],[7],[8]

L2L3 連携方式は、無線レイヤ（以下 L2）と IP レイヤ（以下 L3）の連携を取ることににより L3 のハンドオーバー時間をできるだけ効率よく実行しようとする方式である。一般に L2 と L3 の機能は独立しており、ハンドオーバーの連携もほとんど取られていない。その結果、L2 で AP が切り替わったあとの L3 の処理が効率的に実現できない。L2L3 連携方式では L2 と L3 の連携が目的で、L2 や L3 のハンドオーバーに係る処理は必要なため、これらの処理に係わる通信断絶は避けられない。

### (2)L3 プロトコル拡張方式[9],[10],[11],[12],[13]

L3 のプロトコルを拡張することにより、AP と端末が連携して端末の移動を予測する。端末が AP の切り替え前に IP アドレスを取得しておいたり、ルータでパケットをバッファリングするなどによりパケットロスを防止する。この方式はパケットロス対策としては最も研究が進んでいる。しかし、ネットワーク機器に変更が必要であるため、一般の環境では利用できない。また、端末が予測どおり動けばよいが様々な移動ケースを想定すると制御が複雑になるという課題がある。

### (3)ドライバ改造方式[14],[15],[16]

無線 LAN インフラストラクチャモードでは AP の切り替えが make after break(旧 AP と切断した後には新 AP と接続する方式)であるため、ネットワークでバッファリングしない限りパケットロスが避けられない。そこで、無線レイヤのプロトコル自体を新たな方式に切り替え、make before break(新 AP と接続した後には旧 AP を切断する方式)を可能とする方式が提案されている。しかし、この方式は端末と AP の両者が機能を実装している必要があり、一般の環境では利用できない。また、この方式においても、隣接する AP のチャンネルが異なるような場合は、隣接 AP を探すためにチャンネルスキ

ランを行う必要があるが、この動作のためにパケットロスが発生することは避けられない。

#### (4)デュアルインタフェース方式[17],[18],[19]

端末に無線インタフェースを複数保持させ、一方でパケットの送受信、もう一方でネットワーク監視（チャネルスキャンやAPの切り替え）とIPアドレスの取得を行う。IPアドレス取得後、カードの役割を交換し、逆のカードでそれぞれ通信の継続とネットワーク監視を行う。

この方式は、端末だけに処置をすればよくネットワークには変更が不要である。パケットロスも原理的になくことが可能である。ただし、複数インタフェースを利用するため余分なカードが必要で、消費電流が増加するという課題がある。

### 3. Mobile PPC の概要

本研究で実装の対象とした Mobile PPC について、その概要を記述する。Mobile PPC は第三の装置の助けを借りることなく、エンド端末のみで移動透過性を実現するプロトコルである。Mobile PPC では、通信開始時の IP アドレスの解決（初期 IP アドレスの解決）にはダイナミック DNS(DDNS)を適用する。通信開始後、一方の端末が移動したとき、アドレスがどのように変化したかを知る方法（継続 IP アドレスの解決）として、Mobile PPC による移動情報通知処理を用いる。図 1 に Mobile PPC における移動情報の通知を示す。

Mobile PPC ではエンド端末の IP 層に CIT (Connection ID Table) と呼ぶアドレス変換テーブルを保持する。通信中に一方の端末の IP アドレスが変化すると、エンドエンドでその変化情報を交換し、CIT の内容を更新する。このために使用する packets を CU (CIT Update) および CU Response と呼ぶ。以後の通信は CIT の内容に従って IP 層でアドレス変換を行う。Mobile PPC は、エンド端末のみに機能を実装すればよく、導入の敷居が低い。また、通信経路の冗長やトンネル転送によるパケット長の変化がないため、高スループットを実現できる。既存端末との上位互換性があり、段階的な普及が期待できるなどの特徴がある。

### 4. ハンドオーバーの現状

ハンドオーバー処理を検討するにあたり、次の 2 つの場合を考える必要がある。一つは同一ネットワーク

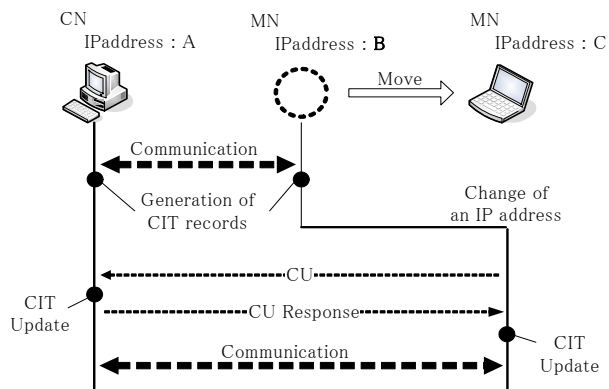


図 1 Mobile PPC における移動情報の通知

内を移動する場合に生じるハンドオーバー(エリア内ハンドオーバー)で、もう一つは異なるネットワーク間を移動する場合に生じるハンドオーバー(エリア間ハンドオーバー)である。前者の場合は L2 のハンドオーバーのみ、後者の場合は L2 ハンドオーバー、IP アドレスの取得、L3 ハンドオーバーが発生する。L2 ハンドオーバーでは、チャネルスキャンによる通信断絶時間が最も大きな時間を占める。

一般に同一ネットワーク内に存在する AP には同一のネットワーク識別子 ESS-ID を割り当てる。移動端末 MN はこの ESS-ID によりネットワークの違いを認識することができる。ESS-ID は AP から定期的送信されているビーコンの情報内に含まれている。MN が AP を切り替えた際に、ESS-ID が変化しない場合は同一ネットワーク内の移動と見なせる。一方 ESS-ID が変化する場合異なるネットワーク間を移動したと判断し、移動先のネットワークに存在する DHCP サーバから新しい IP アドレスを取得する。Windows にはこのような仕組みが標準で実装されている。

エリア間ハンドオーバーの現状を図 2 に示す。図 2 は MN が通信しながらハンドオーバーを実行し、ハンドオーバー終了後に移動先の AP を介して通信を再開するまでの流れを示している。MN は AP からの信号強度が一定レベルより低くなると当該 AP とのアソシエーションを切断する。次に、MN はチャネルスキャンを行い、利用可能な AP を探す。MN は最適な AP を選択し、認証要求メッセージを New AP へ送信する。認証要求メッセージを受け取った New AP は、MN の認証を行い、認証応答メッセージを MN へ返す。次に MN は、再アソシエーション要求メッセージを New AP に送信する。New AP は再アソシエーション応答メッセージ

を MN へ返し、MN と New AP とのアソシエーションが確立される。その後、New AP は IAPP (Inter Access Point Protocol) などのプロトコルを用いて全 AP に対してアソシエーション情報を伝える。IAPP とは、AP 間でアソシエーションデータを伝送するためのアクセスポイント間プロトコルであるが、ベンダ固有のプロトコルが使われる場合もある。以降、MN と AP 間の認証要求、認証応答、再アソシエーション要求、再アソシエーション応答、IAPP に係わる処理をまとめて、再接続処理と呼ぶ。上記チャネルスキャンと再接続処理時間は AP と MN の組み合わせにより大きく異なり、40ms~600ms の時間を要する。いずれの場合においても、チャネルスキャンが上記時間中の 80~99% を占め、続く再接続処理は直ちに終了する。チャネルスキャンと、再接続処理は、AP を切り替える際に常に発生する動作である。再接続処理が完了すると、MN は ESS-ID を確認し、以前と値が異なる場合はネットワークが変わったと判断し、DHCP サーバより新 IP アドレスを取得する。ここでは、DHCP シーケンス(DHCP DISCOVER, DHCP OFFER, DHCP REQUEST, DHCP ACK)と、IP アドレス取得後に行われる Gratuitous ARP による重複アドレスチェックが含まれる。IP アドレス取得処理には最低でも約 2~5 秒の時間を要する。OS と DHCP サーバの相性によっては数十秒を要することもある。この間は IP アドレスが定まらないので通信を行うことができない。IP アドレスの取得を完了すると、Mobile PPC の移動情報通知処理により両端末の CIT の更新を行う。移動情報通知処理時間には、MN と通信相手の CIT 更新時間、CU パケットおよび CU Response パケットの伝送時間が含まれるが、全てを含めても 514  $\mu$  秒程度で終了し、ほとんど無視できる。

## 5. 提案方式

### 5.1. 提案方式の概要

本研究では、ハンドオーバー時間を削減する手段として、デュアルインタフェース方式を選択した。L2L3 連携方式、L3 プロトコル拡張方式、L2 ドライバ改造方式は、いずれもその機能を発揮するには異なる装置や機能の連携が必要で、標準化を行うなど長期の対策が必要である。L2L3 連携方式、L2 ドライバ改造方式は、L2 のチャネルスキャンに係るパケットロスを解決

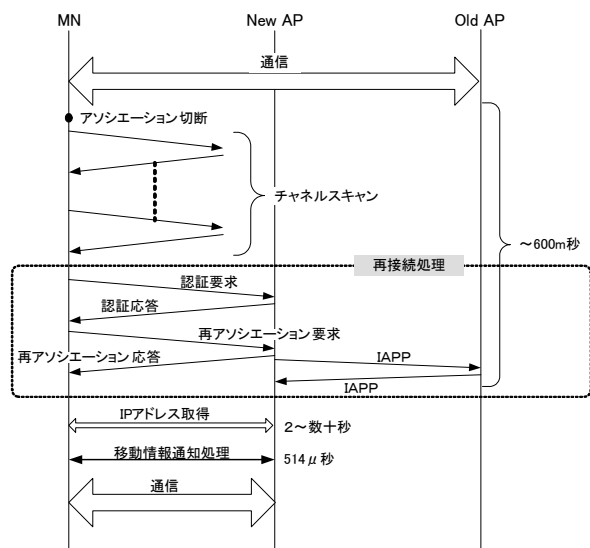


図2 エリア間ハンドオーバーの現状

できない上、MN と DHCP サーバとの相性問題は検討の範疇外である。L3 プロトコル拡張方式はルータがパケットをバッファリングするなどの処理によってパケットロスを無くすることができる可能性があるが、ルータなどのネットワーク機器がその機能をサポートする必要があり、一般のネットワークに適用していくのは難しい。これに対し、デュアルインタフェース方式は端末だけの対策によって、これらの課題を解決できる可能性を秘めている。Mobile PPC を用いて移動透過性の実使用を試みるには最も適した方式であると判断した。

### 5.2. ハンドオーバーシーケンス

図3に提案方式のエリア間ハンドオーバーを示す。MN はカード1で Old AP に接続し、通信を行っているものとする。ここではもう一枚のインタフェースカードはスリープ状態としている。スリープ状態とは省電力状態で、パケットやフレームの送信、受信を一切遮断している。MN はカード1で通信中に、カード1を用いて接続中の Old AP の信号クオリティを測定する。信号クオリティは、信号強度から雑音強度を引いたものである。信号強度は Old AP から送信されるビーコン及び、Old AP を経由し送信されたパケットを受信するとき測定される。ドライバは、AP から受信するパケットに対して、信号強度とノイズレベルに関する情報を維持している。この情報を基に Old AP の信号クオリテ

ィを判断する。信号クオリティが低下し通信の状態が不安定になる前にハンドオーバーを実行できるように、通信に適する信号クオリティの閾値  $\alpha$  を設けておく。Old AP の信号クオリティが一定時間、閾値  $\alpha$  より低くなると、MN はカード 1 による通信を維持しながらカード 2 のスリープを解除する。次にチャネルスキャンを実行し接続可能な AP を探索し、信号クオリティが最も強い AP を次に接続する New AP と定める。さらに、MN は New AP の ESS-ID を調べることによって、ネットワークが Old AP と同一か否か判断する。

New AP と Old AP が同一のネットワークの場合、エリア内ハンドオーバーと判断し、カード 2 をスリープの状態に戻す。次に通信中のカード 1 と Old AP との接続をいったん切断した後、New AP に対して再接続処理を実行しカード 1 での通信を継続する。

New AP と Old AP が異なるネットワークの場合は図 3 に示す動作を実行する。MN はカード 1 による通信を継続しながら、カード 2 により New AP と接続し、DHCP サーバから新 IP アドレスを取得する。次にカード 2 を用いて Mobile PPC の移動情報通知処理を実行して両エンド端末の CIT を更新する。移動情報通知処理後は、新 IP アドレスを持つカード 2 で通信を行う。カード 1 は一定時間後に Old AP とのアソシエーションを切断する。カード 1 でのアソシエーションをしばらく残す理由は、旧 IP アドレス宛のパケットも MN が受信できるようにするためである。MN はカード 1 と Old AP とのアソシエーションを切断した後はカード 1 をスリープ状態にする。

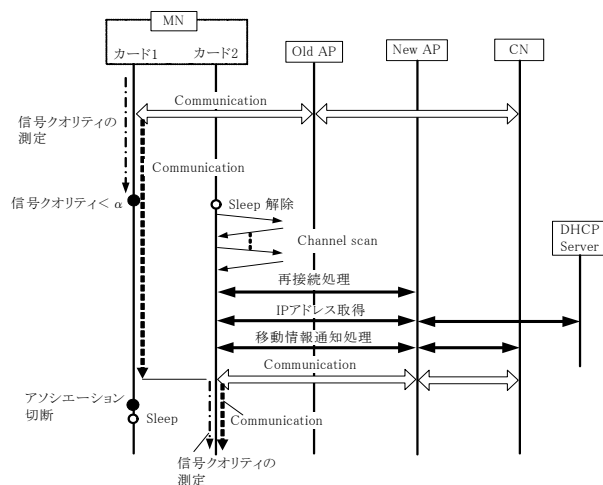


図 3 提案方式のエリア間ハンドオーバー

## 6. 実装

### 6.1. モジュール構成

提案方式によるハンドオーバーアルゴリズムを FreeBSD5.3-Releases 上に実装した。FreeBSD を採用した理由は、Mobile PPC の実装が完了し検証済みのためである。ハンドオーバー処理は全てアプリケーション層にて実装した。

Mobile PPC とハンドオーバー処理のモジュール構成を図 4 に示す。Mobile PPC を実現するモジュールは大きく分けて、アドレス変換モジュールと移動管理モジュールがある。アドレス変換モジュールは、パケット送受信時には IP 入力関数である `ip_input` から、パケット送信時には IP 出力関数である `ip_output` から呼び出され、アドレス変換処理を終えたら差し戻す形をとっている。移動管理モジュールは IP アドレス取得後にカーネルから呼び出され、CU, CU Response の交換と CIT の生成を行う。

ハンドオーバー処理モジュールを Device Change demon(DC デーモン)と呼びデーモンとして動作させる。DC デーモンには、インタフェースカードのスリープとその解除のタイミングを判断する機能、信号クオリティ判定機能、AP の選択とネットワークを判定し、IP アドレス取得を指示する機能がある。信号クオリティ判定機能では、接続中の AP の信号クオリティと閾値  $\alpha$  を比較し、 $\alpha$  より低下した場合は AP を切り替えるための指示へ移行する。AP の選択、ネットワーク判定機能とは、チャネルスキャンで取得した情報から最も信号クオリティが良い AP を選択する機能と、その AP の ESS-ID を確認しネットワークが同一か否かを判断する機能である。

### 6.2. Device Change demon の実行内容

DC デーモンは FreeBSD に標準にインストールされている System Control コマンド(`wicontrol`, `ifconfig`, `dhclient`, `route`)を使用する。使用方法は以下の通りである。通信と並行して `wicontrol` によりドライバ内部で管理されている信号クオリティ情報と New AP の ESS-ID 情報を取得する。チャネルスキャンも `wicontrol` で実行を指示する。New AP への接続処理は、New AP の ESS-ID を引数として `ifconfig` を使用する。これにより、MN と AP 間では自動的に再接続に必要なシ

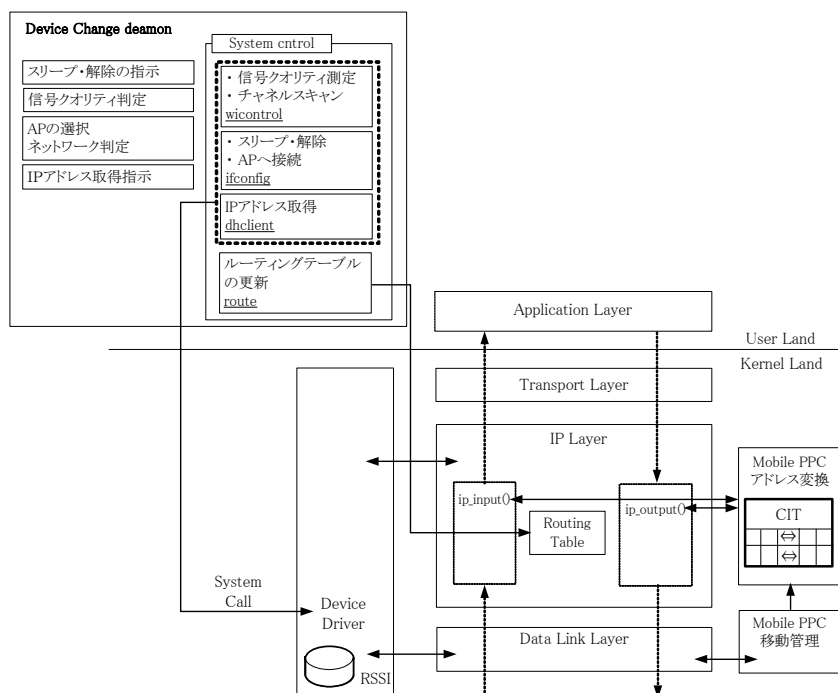


図4 Mobile PPC とハンドオーバー処理のモジュール構成

一ケンスが実行される。インタフェースカードのスリープとその解除にも `ifconfig` を用いる。

IPアドレスの取得には `dhclient` コマンドを実行して、DHCPサーバからアドレスを取得する。インタフェースカードが1枚の場合、`dhclient` はIPアドレスの設定とルーティングテーブル内のデフォルトゲートウェイの設定を自動的に行う。しかしながらインタフェースカードが複数の場合、IPアドレスは自動的に設定されるが、これまで使用していたデフォルトゲートウェイのIPアドレスがルーティングテーブルにそのまま残ってしまうという問題がある。これは `dhclient` が最初にコマンドを実行したインタフェースのみを監視する仕様になっていることが原因である。当初の実装ではこの問題を解決するために、`dhclient` を実行する前にルーティングテーブルからデフォルトゲートウェイを削除していた。こうすることで、`dhclient` により得られたデフォルトゲートウェイの情報が自動的に設定されるようになる。しかしながら、この方法だと、デフォルトゲートウェイを削除してから `dhclient` 処理(IPアドレスの取得、デフォルトゲートウェイの設定)が終了するまでの間、全てのインタフェースにおいてパケットの出力が出来なくなる。パケットを出力するためには、端末が保持するルーティングテーブルにデフォルトゲ

ートウェイの情報が必要となるからである。この間の通信断絶時間は、図3で示したように2秒から数十秒であり、パケットロスが多く発生してしまう。この問題を解決するために、MNのルーティングテーブルの修正を `dhclient` に任せることなくDCデーモンで更新することとした。更新処理では、`route` コマンドを使用して移動前のデフォルトゲートウェイを削除し、移動後のデフォルトゲートウェイの設定を行う。この処理をIPアドレス取得後に実行することで、IPアドレス取得中の通信断絶時間を無くすることが可能となった。更新処理に使用するデフォルトゲートウェイの情報はDHCPより割り当てられたリースリスト `dhclient.leases` の内容を参照することにより知ることができる。

## 7. 評価

上記機能を実装したMNを移動させてハンドオーバー処理を行わせ、所定の動作が可能であることを確認した。以下に試作の評価結果を示す。

### 7.1. 実験環境

提案方式の性能を測定するために図5に示す環境でハンドオーバー実験を行った。2つのルータ R1, R2に

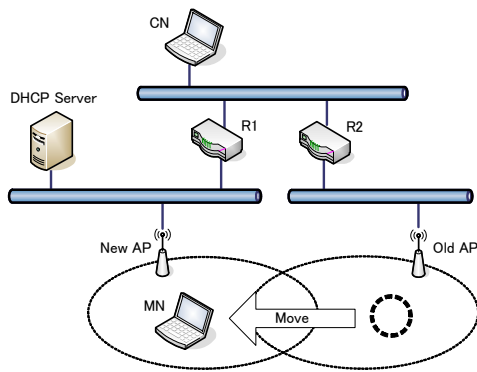


図5 実験環境

表1 装置仕様

	MN	CN / R1 / R2
CPU	Pentium3 597.08MHz	Pentium4 3.0GHz
Memory	191MB	512MB
NIC	corega Wireless LAN PCCL-11	100BASE-TX
OS	FreeBSD5.3-Release	FreeBSD5.3-Release

よりサブネットが異なる3つのネットワークを用意し、MNの移動先となるネットワークにはDHCPサーバを設置した。表1に装置仕様を示す。CN, MNにはMobile PPCを実装している。また、MNには提案方式を実装し、Old APの無線セルからNew APの無線セルへと移動させた。

## 7.2. 通信断絶時間とパケットロスの測定

測定にはIperf[20]を用い、MNとCN間で742バイトのUDPパケットを約56ms間隔で連続的に送信させながらMNの移動を10回試行した。測定した結果、CNからMNへの送信、MNからCNへの送金のいずれの場合においても、パケットのロスは全く発生せず、所定の動作が実現できることがわかった。ただし、MNからCNへの送信においては移動情報通知処理の間に送信する通信パケットは、理論的にはロスが発生することがわかっている。今回の測定では、送信間隔が比較的大きな値を用いたためパケットロスは結果的に発生しなかったといえる。移動情報通知処理に約500 $\mu$ 秒を要しているが、この時間は冗長な処理を省くことにより1 $\mu$ 秒以下にまで短縮できることが分かっており[4]、理論的にもパケットロスはほとんど発生しないようにすることが可能である。

表2 パケットロス数

	パケットロス数
CN $\Rightarrow$ MN	0
MN $\Rightarrow$ CN	0

## 7.3. スループット測定

提案方式ではDCデーモンが常に動作しているため、本処理が通信に影響を与える可能性がある。そこでDCデーモンを実装した場合と、実装していない場合のスループットを比較した。DCデーモンによる電波クオリティの定期監視の間隔は10ms、および100msとした。測定にはIperfを用い、MNとCN間で30秒間のTCP通信を10回試行してパケットの転送量を計測し、その平均をとった。表3にスループット測定結果を示す。CNからMNへの送信と、MNからCNへの送金のどちらの場合も、スループットにほとんど変化がないことが分かる。このことからDCデーモンが通信に与える影響はほとんどないといえる。

表3 スループット

DCデーモン	スループット[Mbps]	
	CN $\Rightarrow$ MN	MN $\Rightarrow$ CN
実装なし	4.923	4.895
実装あり(間隔:100ms)	4.925	4.89
実装あり(間隔:10ms)	4.928	4.89

## 7.4. 消費電力

デュアルインタフェース方式は、これまで消費電力が増加するという課題が指摘されていた。しかしながら、本提案方式では通信中のカードを用いて信号クオリティの測定を行い、もう一枚のカードはスリープ状態にしておくため、両カードが同時に動作するのはハンドオーバー時のみである。表4にスリープ機能付きの無線LANインタフェースの消費電力を示す[21]。パケットを送信するために必要な電力は543mW、受信するために必要な電力は384mW、受信待ち受けに必要な電力は263mWである。それに対し、スリープ状態ではインタフェースカードへの漏れ電流のみとなるため、57 $\mu$ Wと極わずかとなる。このようにスリープ状態ではほとんど電力を消費しないため、本提案方式ではカード1枚の場合と比較しても消費電力はほとんど同じで実現できる。

表 4 消費電力

送信中	543mW
受信中	384mW
受信待ち受け	263mW
スリープ状態	57 $\mu$ W

## 8. まとめ

本研究では、端末側だけの処置で実現が可能なことに着目し、デュアルインタフェース方式を選択してパケットロスレスハンドオーバーの提案を行った。提案方式を Mobile PPC に適用し評価を行った結果、パケットロスのほとんどないハンドオーバーを実現できること、一般通信に与える負荷は十分に小さいことが確認できた。

本提案方式は Mobile PPC の適応時に有効な手段となりうるが、他の移動透過性技術に対しても有効な方式であると考えられる。また、Mobile PPC のようなエンドエンドの移動透過技術では MN と CN が全く同時に移動した場合に通信が継続できないという課題が指摘されている。本提案方式を用いればこのような状況においても移動透過性を実現できる可能性があり、今後検討を進めていきたい。

## 参考文献

[1] 寺岡文男, "インターネットにおけるモバイル通信プロトコルの標準化動向," 電子情報通信学会論文誌, Vol.J84-B, No.10, pp.1746-1754, Nov.2000

[2] C. E. Perkins."IP Mobility Support for IPv4,"RFC 3344. Aug.2002.

[3] D.Johnson, C. Perkins, J. Arkko, "MobilitySupport in IPv6," RFC3775. June 2004.

[4] 竹内元規, 鈴木秀和, 渡邊晃, "モバイル端末の移動透過性を実現する Mobile PPC の提案," 情報処理学会論文誌, Vol.47, No.12, pp.3244-3257, Dec.2006.

[5] Vixie. P., Thomson. S., Rekhter. Y., and Bound. J.: Dynamic Updates in the Domain Name System (DNS UPDATE), RFC 2136, IETF (1997).

[6] 後郷和孝, 神谷弘樹, 渋井理恵, 金子晋丈, 玉載旭, 小森田賢史, 藤巻聡美, 寺岡文男, "リンク層情報を利用したネットワーク層主導高速ハンドオーバー機構の設計と実装" 情報処理学会研究報告. MBL, Voll.2005, No.47 (20050525), pp. 13-18

[7] 井島亮一, 塚本和也, 檜原茂, 尾家祐二, "WLAN ハンドオーバーにおける新たな決定指標の調査," 電子情報通

信学会研究報告, IN2005-40, pp.67-72, (2005,7)

[8] Shakkottai, S. Rappaport, T.S. Karlsson, P.C. , "Cross-Layer Design for Wireless Networks," IEEE Communication Magazine, Vol. 41, No. 10, pp. 74-80, October 2003.

[9] 小川猛志, 伊東匡, "DHCP をベースとしたシームレスハンドオーバー方法の研究", 電子情報通信学会論文誌 B Vol.J88-B No.11 pp.2228-2238, 2005/11/01

[10] 高橋秀明, 小林亮一, 岡島一郎, 梅田成視, "Hierarchical Mobile IPv6 with Buffering Extension の通信品質評価," 情報処理学会論文誌 Vol.46 No.2, Feb.2005

[11] 萬代雅希, 笹瀬巖, "Mobile IP における位置情報を用いた低レイテンシなハンドオフ方式", 情報処理学会論文誌, Vol. 45, No. 4, pp. 1121-1133, Apr. 2004.

[12] 本山智祥, 首藤晃一, 奥村康行, "スヌーピングルータ (SR)適用によるスムーズハンドオーバーモバイルネットワーク" 電子情報通信学会論文誌 B, Vol.J88-B, No.3 (20050301) pp. 622-633

[13] H. Soliman, C. Castelluccia, K. El-Malki, and L.Bellier, "Hierarchical MIPv6 mobility management (HMIPv6)," RFC4140, Aug. 2005.

[14] モバイルブロードバンド協会, "MBA 標準 0201 号「MIS プロトコル仕様書 Ver.1.02」," <http://www.mbassoc.org/j-services/mbas0201r060606.pdf>

[15] モバイルブロードバンド協会, "MBA 標準 0202 号「MIS モバイル IP 仕様書 Ver.1.02」," <http://www.mbassoc.org/j-services/mbas0202.txt>, 2004.

[16] 森岡仁, 真野浩, 太田昌孝, 寺岡文男, MIS プロトコルと PDMA による高速ハンドオーバー," 電子情報通信学会技術研究報告. WBS, IEICE technical report, Vol.104, No.672(20050223) pp. 243-248

[17] 松岡保静, 吉村健, 大矢智之, "エンドツーエンド型 IP ソフトハンドオーバー", 電子情報通信学会論文誌, B, Vol. J86-B, No.8, pp.1369-1378, 2003 年 8 月.

[18] 森岡仁志, 大森幹之, 太田昌孝, 真野浩, "2 台の無線 LAN 送受信機を用いたシームレスハンドオーバーの実現," 第 10 回マルチメディア通信と分散処理(DPS)ワークショップ論文集, pp.263-268, Oct.2002.

[19] 相原玲二, 藤田貴大, 岸場清吾, 田島浩一, 西村浩二, 前田香織, "常に最適経路で通信を行う移動透過アーキテクチャ MAT の性能評価"インターネットコンファレンス 2006 論文集, pp.13-20, Oct.2006.

[20] Iperf : <http://dast.nlanr.net/Projects/Iperf>

[21] <http://www.keystream.co.jp/tech/>  
<http://www.keystream.co.jp/tech/save01.php>