

Title	モバイルネットワークにおけるネットワークアーキテクチャに関する研究
Author(s)	森川, 大樹
Citation	
Issue Date	1999-03
Type	Thesis or Dissertation
Text version	author
URL	http://hdl.handle.net/10119/1254
Rights	
Description	Supervisor:中島 達夫, 情報科学研究科, 修士

修士論文

モバイルネットワークにおける ネットワークアーキテクチャに関する研究

指導教官 中島 達夫 助教授

北陸先端科学技術大学院大学
情報科学研究科

森川 大樹

1999年2月15日

目次

1	序論	1
2	関連研究	5
2.1	IETF Mobile IP	5
2.1.1	IETF Mobile IP の概要	5
2.1.2	問題点	7
2.2	VIP	8
2.2.1	VIP の概念	8
2.2.2	VIP の問題点	11
2.3	まとめ	12
3	移動計算機環境の問題点と提案	13
3.1	現在の移動計算機環境	13
3.2	問題点	14
3.2.1	IP アドレスの動的な変更	14
3.2.2	多様な通信メディア	14
3.2.3	メディア切替えのタイミング	15
3.2.4	通信メディアの切替えによる通信特性の変化	15
3.3	まとめ	16
4	JAIST Mobile IP	17
4.1	改善方法	17
4.2	JAIST Mobile IP	19
4.2.1	システムアーキテクチャ	19
4.2.2	設計	21
4.3	まとめ	27

5	実装	28
5.1	実装	28
5.1.1	全体構成	28
5.1.2	RT-Mach への実装	29
6	評価と考察	38
6.1	3種類の手法の比較	38
6.2	IETF Mobile IP との性能比較	40
6.3	まとめ	44
7	まとめと今後の課題	46

目次

1.1	移動計算機環境	2
2.1	IETF Mobile IP	6
2.2	ネットワーク層の分離	9
2.3	VIP	11
4.1	JAIST Mobile IP	20
4.2	プロトコルスタックと通信メディアの分離	21
4.3	通信メディア切替え機構	23
4.4	環境サーバとハンドオフコントローラ	25
4.5	パケットスヌープ機構	26
5.1	全体構成	29
5.2	RT-Mach への実装	30
5.3	メディア切替え機構	32
5.4	移動計算機内のスヌーパ	33
5.5	プロキシエージェント	34
5.6	パケットスヌープ機構の構造	35
6.1	実験環境	40

表 目 次

6.1	3 種類の手法の比較	39
6.2	計算機のスペック	41
6.3	測定結果	42
6.4	メディア切替え機構でのパケット処理の時間	42
6.5	プロキシエージェントでのパケット処理の時間	43
6.6	デバイスとのパケット送受信の時間	43

第 1 章

序論

無線や有線などのネットワーク技術の進歩によりインターネットや LAN などのネットワークへ接続の手段が大幅に拡大された。また、PDA やノート型計算機など小型計算機の発達により計算機の携帯が可能となり屋内だけではなく屋外や移動先で計算機を利用することが多くなった。これらの小型計算機には動的に抜き差しが可能な PC カードを利用することで様々な通信メディアを切替えて使用することが可能である。そして、こういったネットワーク技術や様々な通信メディア用の PC カード、小型計算機を利用することにより、いつでもどこにおいてもネットワークへ接続し他の計算機を含む計算機資源を利用することができるようになった。

例えば、研究室では Ethernet や ATM、FDDI などの高速な有線メディアを利用し、会議室では低速ではあるが配線などを考慮しなくてもすむ無線メディアを、屋外に出たときには携帯電話や PHS を使用してネットワークに接続するということが可能である。

このような環境 (以下、移動計算機環境と呼ぶ) において、移動先やユーザの要求によって通信メディアを動的に切替えなければならないときや、複数の通信メディアから一つを選択しなければならない場合が生じる。また、これらの操作を行なった時に通信性能が極端に変化する場合がある。これは通信メディアごとに様々な特性があり、それらの相違によって起こる。

一方、現在のネットワークアプリケーションはクライアント/サーバ型の形態をとっているものが多い。例えば、Web サーバや nntp サーバ、ftp サーバなどがある。こうしたアプリケーションのほとんどはクライアントとサーバ間でコネクションを必要とする。このようなアプリケーションを使用中に計算機を移動したり通信メディアを切替えるとコネクションを持続できなくなるため作業を続けようと思うと一度アプリケーションを立ち上げ直したり、最悪計算機の設定を変更して再起動する必要がでてくる。これらに対処する

方法として、アプリケーション側で計算機の移動や通信メディアの切替えをサポートするという方法もあるがそれぞれのアプリケーションで行なうのは柔軟性に乏しく非効率的である。

こうした状況の中、システムレベルでの計算機の移動透過性に対する期待が高まってきている(図 1.1)。このようなシステムの実現により計算機の移動の際の作業の中断やネットワーク設定などから解放される。さらに、アプリケーションレベルで計算機の移動を考慮せず従来の IP と同様に扱えるため開発・運用の面での有用性も期待できる。

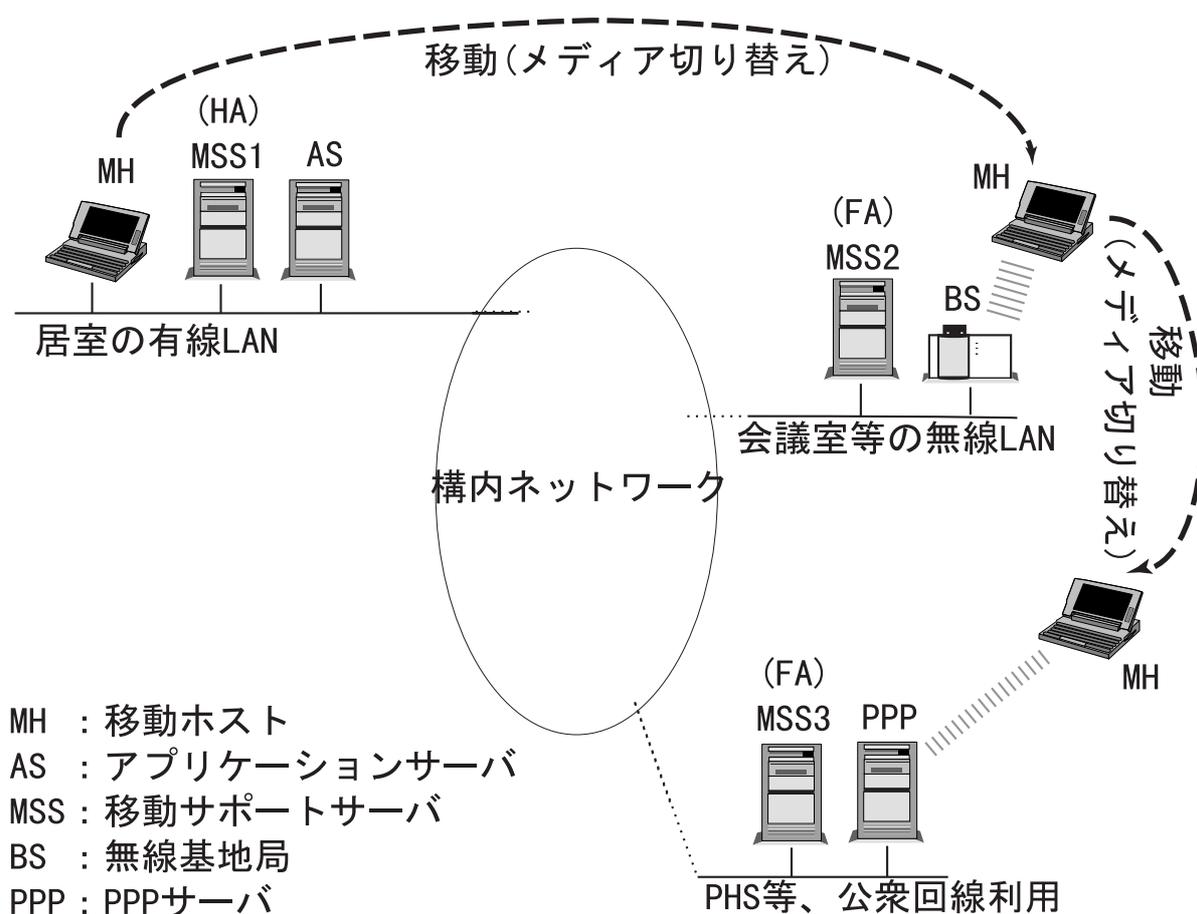


図 1.1: 移動計算機環境

このような移動計算機環境を実現させるために様々な研究が行なわれている。スタンフォード大学の MosquitoNet [4] による状況に応じて複数の通信メディアを使い分けを行なうための研究や、IETF Mobile IP(以下、IETF MIP) [5] や Sony CSL の VIP [7] による計算機の移動透過性に関する研究などがある。しかし、これらの研究にはいくつかの問題点が残されている。例えば、IETF MIP ではネットワーク上にホームエージェントやフォーリンエージェントと呼ばれるサーバを配置することで計算機の移動を可能としている。移動する計算機の IP アドレスはネットワークへの接続点ではなく計算機の識別子として扱っている。これによりネットワークへの接続場所を意識することなく計算機の識別を可能としている。しかし、現在のシステムソフトウェアでは計算機を識別するための識別子とネットワーク接続点とが明確に分離されていない。

こうした多種多様な問題がある中、本研究では以下の4点の問題に焦点を合わせ、これらの問題を解決することで上述したような移動計算機環境を実現する。

明確な計算機識別子

現在の計算機のネットワーク環境で与えられている IP アドレスは計算機の識別及び、ネットワークの接続点の両方の役割を担っている。計算機が移動しない場合はこれで問題はないが我々が想定する移動計算機環境では計算機の移動などにより IP アドレスが動的に変化してしまい、IP アドレスの扱いがネットワークの接続点として使えるが計算機の識別としては使えなくなる。我々の想定する移動計算機環境においては計算機を識別することは非常に重要である。

多様な通信メディア

図 1.1 のような環境を想定した場合、移動計算機は複数の通信メディアを常時利用することが可能である。この環境を活用するためには計算機の動作中にそれらを切替える仕組みが必要不可欠である。

メディア切替えのタイミング

無線ネットワークにおけるハンドオフ処理に関しては様々な最適化が考案されているが、異なる通信メディアの切替えを考慮しているシステムはない。また、円滑な通信メディアの切替えを可能にするには使用できるメディアの情報を統一して管理し、その情報を元に切替えのタイミングを適切に制御する必要がある。

メディア特性の変化

上記の問題を解決したとしても、単に通信メディアを動的に切替えネットワークに接続するだけでは十分な通信性能が得られないと考えられる。これは、通信メディアの切替えなどにより先程までの通信メディアのエラー率やバンド幅、遅延などの特性に大きな違いが生じるからである。したがって、各メディアの特性に応じて通信性能を向上するための機構が必要となる。

本研究では、これら 4 つの問題に対処し柔軟な移動計算機環境のためのフレームワークとして JAIST Mobile IP(以下、JAIST MIP) を提供する。以下 2 章では関連研究について述べ、3 章では現在の計算機環境の問題点、関連研究での問題点を述べ、4 章ではそれらに対処する JAIST MIP を提案する。5 章では JAIST MIP の設計と実装について述べ、6 章では JAIST MIP の評価について論ずる。

第 2 章

関連研究

2.1 IETF Mobile IP

計算機の移動に伴う問題点に対処するためのプロトコルとして IETF Mobile IP [5] や VIP [7] などがある。本節では IETF Mobile IP について述べ、次節では VIP について述べる。

2.1.1 IETF Mobile IP の概要

Mobile IP は、ネットワークの物理的な接続点が変わっても、同じアドレスを使用し続けることを可能とする IP 層での移動透過性を提供することを目的としている。IETF MIP では移動する計算機 (移動ホスト) は拠点とするネットワーク (以下ホームネットワーク) とホームネットワーク上の唯一の IP アドレス (以下、ホームアドレス) を持つ。ホームネットワークには移動ホストのホームアドレスを管理するホームエージェントがある。また、このホームエージェントは移動ホストが移動した時の移動先でのアドレス (以下、気付けアドレス) も管理している。移動ホストの移動先のネットワークにはフォーリンエージェントが配置されている場合がある。このフォーリンエージェントは移動先で移動ホストへのパケットの転送や登録の補助を行なう。これらの概念図を図 2.1 に示す。

この図において R1、R2、R3、R4 はルータであり MH は移動計算機、S は移動計算機と通信を行なっている計算機である。特に R2 のあるサブネットは移動計算機のホームネットワークであり、そのネットワーク上に存在する R2 がこのサブネットでのホームエージェントとなっている。また、R4 はフォーリンエージェントである。

IETF MIP の基本的な動作は移動ホストが別のネットワークへ移動している時にホー

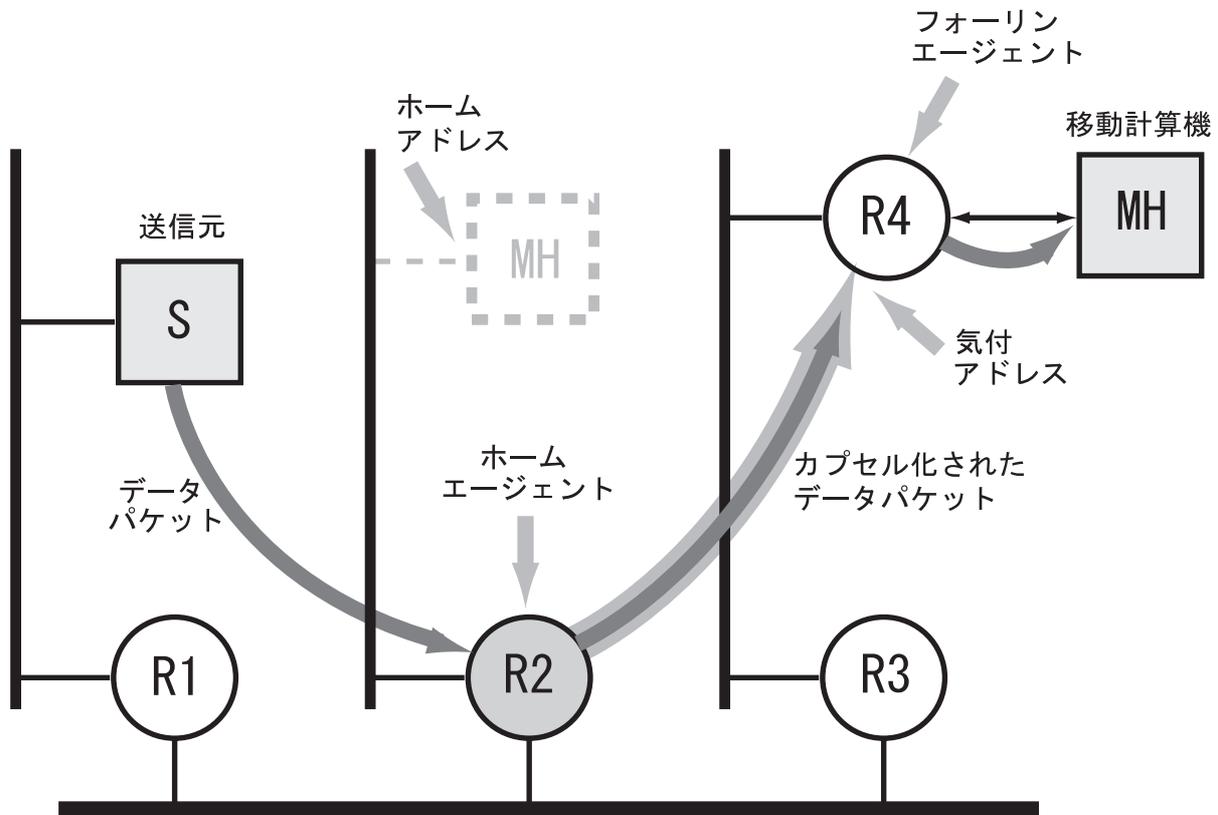


図 2.1: IETF Mobile IP

ムエージェントが移動ホスト宛てのパケットを転送することである。移動ホストはホームネットワークから他のネットワークへ移動すると、移動先のネットワーク上で気付けアドレスを取得する。この気付けアドレスの取得方法は移動先のネットワーク上のフォーリンエージェントから割り当ててもら場合と移動ホスト自身が DHCP [9] などを使用して取得する場合がある。

気付けアドレスを取得した移動ホストはその気付けアドレスをホームエージェントに登録する。これによりホームエージェントは移動ホストが現在どこにいるかを把握する。ホームエージェントはホームネットワークに送られてきた移動ホストのホームアドレス宛ての IP パケットを移動ホストの代わりに受け取り IPinIP [6] のカプセル化を施し移動ホストが登録した気付けアドレス宛てに転送する。気付けアドレスがフォーリンエージェントの場合はフォーリンエージェントによって IPinIP のパケットから元の IP パケットを取り出し、そのパケットを移動ホストへ転送する。気付けアドレスが移動ホスト自身で確保したアドレスの場合は移動ホスト自身が IPinIP のパケットから元の IP パケットを取り出

す。このようにして移動ホストがどこに存在しても移動ホスト宛てのパケットが移動ホストへ届くことを可能としている。

移動ホストが送信する IP パケットのソースアドレスは移動先でもホームアドレスを用いている。これにより移動ホストの通信相手は常にホームアドレス宛てに IP パケットを送るので、移動ホストはどのネットワークに移動しても通信を続けることが可能となる。また、通信相手は移動ホストが移動しているかどうかを気にせずにする。

このように IETF MIP ではホームアドレスはネットワークインターフェースに割り付けられたアドレスではなく移動ホストの識別子として働く。ネットワークへの接続点としての IP アドレスは、気付けアドレスとして移動したネットワーク上でその都度確保される。そして、通信相手はホームアドレスを知っているだけで移動ホストがどこにしようともネットワークに接続している限り移動ホストとの通信は保証される。

2.1.2 問題点

上記のようにして IETF MIP では計算機の移動透過性を実現している。しかし、以下に示すような問題点が残っている。

計算機識別子とネットワーク接続点の混合

気付けアドレスを移動ホスト自身が確保する場合、ホームエージェントから転送されてくるカプセル化 IP パケットを受け取るために、移動ホストの通信メディアは気付けアドレスで動作しなければならない。一方移動計算機から通信相手のホストへ送信する IP パケットのソースアドレスには気付けアドレスではなくホームアドレスを使用しなければならない。そのため、ネットワークに接続された通信メディアはネットワークからの ARP などに対しては気付けアドレスを持ったホストとして動作し、送信される IP パケットにはホームアドレスを持ったホストとして動作するといった変則的な動作が必要となる。

また、気付けアドレスをフォーリンエージェントによって確保する場合、移動先のネットワークで、フォーリンエージェントから移動ホストへ転送される IP パケットの宛先アドレスはホームアドレスとなっている。この場合ホームアドレスは別のネットワーク上のアドレスなので通常の IP パケットのルーティングを用いることができない。もし、通常の IP パケットのルーティングを行なうとこのパケットはまたホームネットワークへと送信されたことになり無限にルーティングされ続けてしまう。このためフォーリンエージェントはリンク層を直接制御して移動ホストにパケットを転送しなければならない。リンク層がイーサネットの場合、フォーリンエージェントは移動ホストの MAC アドレスを指定してパケットを転送する。

移動ホストが PPP で接続する場合、移動ホストは PPP サーバを介してフォーリンエージェントの配置されたネットワークに接続される。そのためフォーリンエージェントがホームアドレス宛ての packets を転送する先は PPP サーバになる。したがって PPP サーバも IETF MIP を認識しなければならない。このようにフォーリンエージェントから移動ホストへの IP packets の転送はリンク層の制御が必要になり、移動ホストが使用する通信メディアごとに IETF MIP に対応する必要がある。

これらの問題の根本は計算機の識別子としてのホームアドレスが移動先のネットワーク上で時にはネットワーク接続点として扱われているということである。

通信メディアの特性への対応

通信メディアにはイーサネットや無線 LAN、PPP など様々な種類がある。移動計算機環境ではこれらをユーザの要求やその場の環境に応じて切替えて使用することが考えられる。しかし、IETF MIP ではこれらを動的に切替えることを考慮していないという問題がある。

また、移動計算機が移動先でイーサネットなど有線メディアと比較してバンド幅やエラー率などの通信性能の劣る無線 LAN や PPP を使用すると移動計算機と移動計算機とフォーリンエージェント間の通信が他の基盤ネットワークより性能が劣るためこの部分が end 間通信のボトルネックとなってしまう。移動計算機環境では通信メディアを切替えて使用されると思われる。end 間通信全体のパフォーマンスを向上するための研究がなされているが、これらには静的なネットワークに関するものが多い。通信メディアにはそれぞれ様々な特性があるため通信性能を改善する方法も動的に変更する必要がある。IETF MIP では、これらの機構の欠如のために end 間通信の通信性能が低下するといった問題がある。

2.2 VIP

移動計算機環境における計算機の移動透過性を実現するためのシステムとしてソニー CSL では VNP (Virtual Network Protocol) を提案している。以下では VNP を IP で使用する VIP (Virtual Internet Protocol) [7] について述べる。

2.2.1 VIP の概念

VIP は IETF MIP と同様にネットワークの物理的な接続点に変化しても通信を持続できる移動透過性を提供することを目的としている。VIP では従来のネットワーク (物理

ネットワーク) 上に仮想ネットワークという論理的なネットワークを導入している。仮想ネットワークは相互接続された論理的なネットワークを物理ネットワーク上に形成する。そして VIP を利用する各ホストは仮想ネットワークと物理ネットワークでのアドレスを持つ。これらはそれぞれ仮想ネットワークアドレス (VN アドレス)、物理ネットワークアドレス (PN アドレス) と呼ばれる。各ホストは物理ネットワーク上を移動しても仮想ネットワークの世界では移動しない。つまり物理ネットワーク上を移動し PN アドレスが変化しても VN アドレスは常に同じとなっている。このように VIP では計算機の識別子を提供することで移動透過性を保証している。

これらの仮想ネットワークと物理ネットワークは OSI の 7 層モデルでのネットワークレイヤを仮想ネットワークサブレイヤと物理ネットワークサブレイヤに分離することにより実現している。仮想ネットワークサブレイヤは物理ネットワーク上でのホストの位置を隠蔽する。ホストには VN アドレスが割り当てられておれ、送信する場合は VN アドレスは対応する物理アドレスに変換される。物理ネットワークサブレイヤは従来のネットワークレイヤである。このサブレイヤはパケットの中継、経路制御情報の交換等を行なう。ホストには PN アドレスが割り当てられる。(図 2.2)

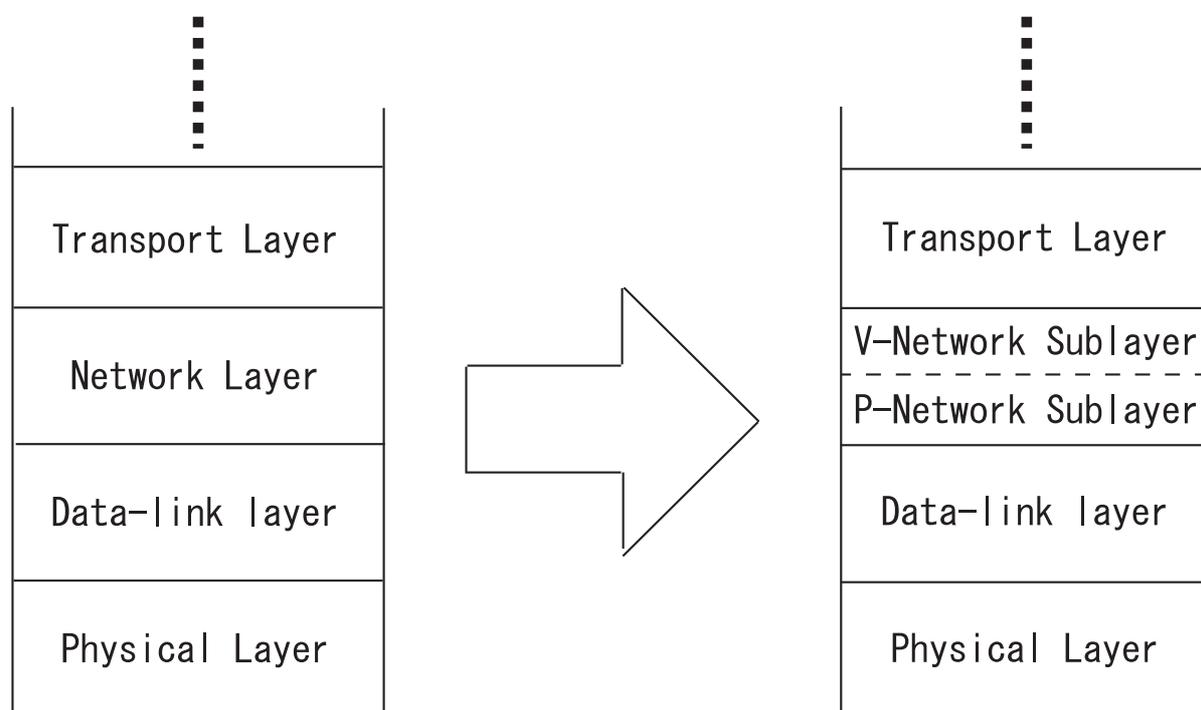


図 2.2: ネットワーク層の分離

VIP の仮想サブレイヤでは VN アドレスと PN アドレスの変換を行なっている。VIP をサポートするホストやルータは AMT(Address Mapping Table) を持っている。これは VN アドレスと対応する PN アドレスのテーブルであり、仮想サブレイヤでアドレス変換を行なう際に参照される。AMT は拡散キャッシュ法と呼ばれる方法でホストやルータに拡散される。

拡散キャッシュ法

拡散キャッシュ法では各ルータやホストは AMT を保持している。移動ホストが移動すると拠点とするネットワーク上のルータにその位置を登録するためコントロールパケットを送信する。この時にこのコントロールパケットが途中で通ってきたルータは AMT エントリに移動ホストの VN アドレスと PN アドレスを加える。こうすることにより途中にあった計算機は移動ホストへ移動ホスト宛てのパケットを送信できるようになる。

また移動計算機がある計算機と通信を行なっている際に通るルータやホストも AMT エントリを更新する。こうすることにより通信が行なわれるにつれて移動ホストの情報が他のルータやホストにも拡散していくため、多くのパケットは経路上のルータでフォワードされることになる。すなわちあるホストが移動ホストからパケットを受信すれば、そのホストは移動ホストの PN アドレスを知ることになり、それ以降は移動ホストに直接パケットを送信できるようになる。

AMT エントリの拡散は VN サブレイヤのコントロールパケットの発信と通常のデータパケットの発信で起こる。コントロールパケットは移動ホストがネットワークに接続されたり切断されたりする時に送信される。

図 2.3 にホスト移動時のパケットの流れを示す。図において R1、R2、R3 はルータであり、MH が移動ホスト、S が MH と通信をしているホストである。左のネットワークを拠点とするネットワークと仮定する。図では移動ホストは拠点ネットワークから右側のサブネットに移動している。拠点ネットワークにいる時はホスト S からのパケットは従来通り R1 を経て移動ホスト MH へ送られる。移動ホスト MH が右のネットワークへ移動すると、まずそのネットワークでの PN アドレスを取得する。この時 VN アドレスは不変である。次に拠点としているネットワークへコントロールパケットを送信する。このパケットは R1、R2、R3 を通るためこれらのルータは移動ホストの VN アドレスと PN アドレスを AMT へ作成/更新する。この状態でホスト S が移動ホスト MH へパケットを送信とする。ホスト S は移動ホスト MH が拠点としているネットワーク上にいると思ってい

るので、R1へパケットを送信する。R1がこのパケットを受け取るとAMTエントリにある移動ホストMHの情報により移動ホストが移動していることを感知し宛先のPNアドレスを現在の移動ホストのPNアドレスに変換しR2→R3→移動ホストという風にパケットを転送する。そして移動ホストはホストSへパケットを送信し、そのパケットによりホストSはAMTエントリを作成/更新する。こうすることにより次回移動ホストSへパケットを送信する時は直接現在の移動ホストMHのPNアドレス宛てに送るようになる。

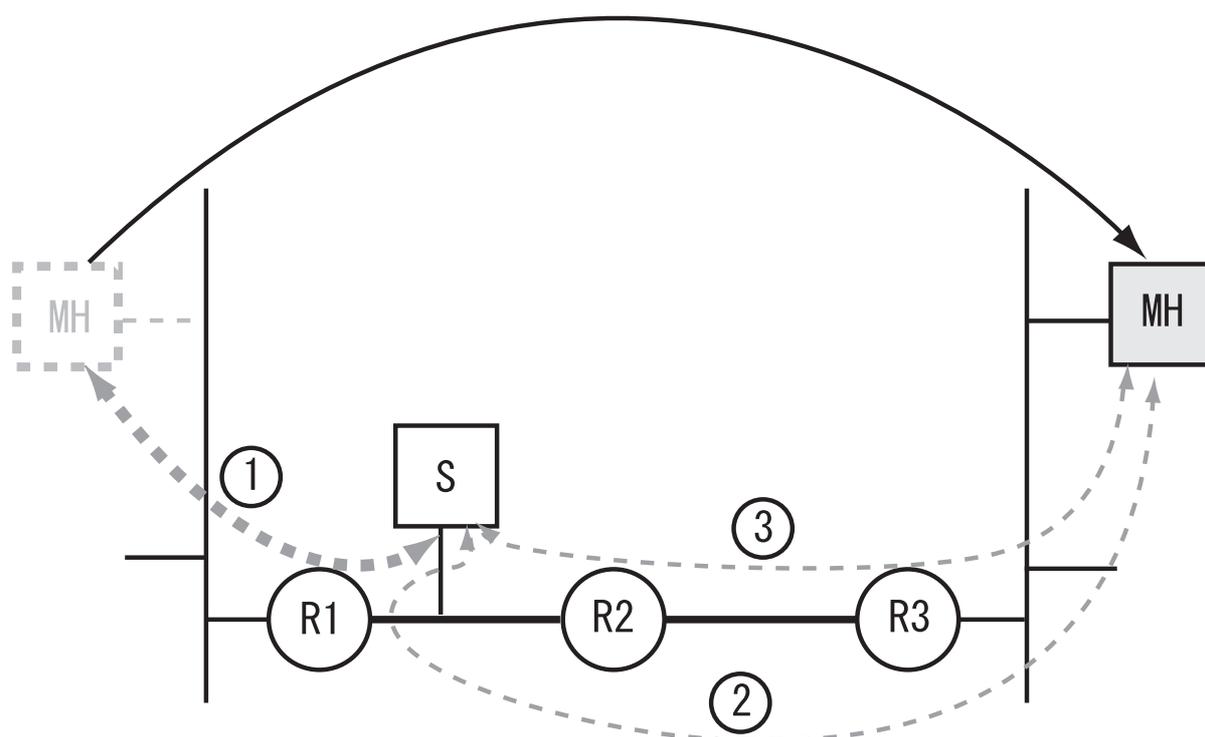


図 2.3: VIP

2.2.2 VIP の問題点

VIP では上記で述べた方法で計算機の移動透過性を実現している。しかし、以下のような問題点も考えられる。

ネットワーク全体への対応

VIP での重大な問題点としては、VIP を使用する際にネットワーク全体がVIPに対応する必要があるということである。ネットワーク全体がVIPに対応していなくても一応

は動作するが、拡散キャッシュ法などでの利点が全く得られなくなる。現在のシステムの全てを VIP 対応とするのは非常に難しい。

通信メディアの特性

移動計算機環境において移動計算機が拠点とするネットワーク外に移動した時に使用する通信メディアとしては無線 LAN などイーサネットと比較すると通信性能の劣るメディアを使用することが考えられる。そのような比較的通信性能の低い通信メディアを使用してネットワークに接続していると、その通信メディアの特性のために通信性能が極端に低下してしまうことが起こり得る。これは IETF Mobile IP でも考えられる問題であるが、IETF MIP ではフォーリンエージェントを拡張することでこの問題を解決する方法がある。しかし、VIP ではフォーリンエージェントというものがなく通信メディアの特性に適応するのは現状では無理である。

2.3 まとめ

移動計算機環境で必要となるであろう計算機の移動透過性を実現するプロトコルの研究が行なわれている。しかし、これらの研究にもいくつかの問題点や課題が残っている。移動透過性を実現しても通信性能が極端に低下するのであれば非常に問題となる。移動透過性を実現し、かつ、通信メディアの特性に適応するようなシステムが必要である。

第 3 章

移動計算機環境の問題点と提案

3.1 現在の移動計算機環境

様々なネットワーク技術が発展し、計算機環境の基盤が整備されてきたため、いつでもネットワークに接続可能な環境となった。そのような中で重要となるのは計算機の移動透過性である。この移動透過性を実現させるために 2 章で述べたような様々な研究がなされている。

現在の移動計算機環境で利用可能な通信メディアは多様である。イーサネットや FDDI のような有線メディアは高速だが屋内の特定の場所でしか利用できない。無線 LAN は有線メディアに比べれば低速であるが移動が可能であるといった利点がある。しかし、接続可能な範囲は基地局から数 100m の距離内に限られている。構内での PHS システムを利用したり、携帯電話を利用すると使用範囲はかなり広範囲となるが通信速度は大幅に低下してしまう。このようにいつでもどこでも利用可能な通信メディアは存在するが、一つのメディアで全ての要求に応じられることはない。したがってネットワークに接続する場所やその時の条件によって通信メディアを選択する必要がある。

通信メディアの動的な切替えは現在では PC カードを利用することにより可能である。イーサネットなどの有線メディアや無線 LAN、PIAFS などの無線メディアはこの PC カードにより実現されている。PC カードを差し替えることにより計算機の動作中にでも必要に応じて動的に通信メディアを切替えることが可能となっている。例えば、研究室ではイーサネットのような有線メディアを使用し、会議室へ移動すると PC カードをイーサネット用から無線 LAN 用に差し替えて会議室でネットワークに接続できる。さらに無線 LAN の有効範囲の外にでる時には PHS や携帯電話用の PC カードに差し替えればネットワークへの接続は可能である。

3.2 問題点

移動計算機環境では、小型計算機を携帯し、様々な通信メディア用の PC カードを差し替えることでハードウェア的にはいつでもどこでもネットワークに接続することが可能となっている。しかし、同時に移動計算機環境において従来では考えられていなかった問題が発生している。

3.2.1 IP アドレスの動的な変更

従来の計算機環境では計算機は固定されており IP アドレスは静的なものであった。そのため IP アドレスを計算機の識別子として使用してもさほど問題は無かった。しかし、移動計算機環境では IP アドレスが動的に変化することがある。これにより IP アドレスを計算機の識別子として利用することが従来のままでは不可能であった。この IP アドレスの変化の要因としては以下の 2 点が挙げられる。

計算機の移動

IP アドレスは各ネットワークのサブネット毎に定められたネットワークアドレス部とそのサブネット内の計算機を識別するためのホストアドレス部から成っている。そのため計算機がサブネットを越えて移動した場合 IP アドレスを変更する必要がある。

通信メディアの切替え

現在のシステムでは IP アドレスは計算機にではなく通信メディアに与えられている。そのため移動計算機環境において通信メディアを切替えるということは IP アドレスを変更することを意味する。

現在の TCP/IP では通信相手を識別するために IP アドレスを使用している。そのため上記のような理由で IP アドレスが動的に変化してしまうということにより通信が維持できなくなる。

3.2.2 多様な通信メディア

移動計算機環境では計算機の動作中に通信メディアを動的に切替えるということが考えられる。しかも、PC カードを使用した場合は通信メディア用の PC カードが差し替えられるためデバイスそのものが計算機からは認識できなくなったり新たに出現することになる。そのため計算機の動作中に通信メディアの初期化や終了処理が必要となってくる。

また通信メディアの種類が豊富にありそれぞれの通信メディアによって異なる特徴を持っている。例えば、無線 LAN などでは基地局に異なるチャンネルを割り当てることで基地局ごとに異なるサブネットを与えることができるためネットワークに接続するためには基地局を選択する必要がある。特に PPP 接続の場合は回線の接続を明示的に行なわなければならない。これらの処理はイーサネットではまったく必要としない。これらをシステム側で行なうためには計算機のシステムでリンク層レベルの制御を積極的に行なう必要がある。

現在の計算機でネットワークに接続するためにはこれらリンク層レベルの制御だけでなくネットワーク層でのアドレスや経路なども制御する必要がある。これらネットワーク層でのアドレスや経路というものは使用している通信メディアに依存している。そのため、通信メディアを切替えるとネットワーク層での処理も必要となってくる。現状ではネットワーク層とリンク層の処理はそれぞれ独立して行なわれている。つまりリンク層の制御はシステム側が行ないネットワーク層でのアドレスなどは例えば DHCP などによって行なわれる。通信メディアを動的に切替える移動計算機環境においてはメディアを切替える度にそれぞれの処理を独立して行なわなければならない。

3.2.3 メディア切替えのタイミング

無線 LAN を利用する場合、基地局位置が通信可能領域を制限するので場所にあわせて接続する基地局を変更する必要がある。この問題を解決する方法はハンドオフ処理と呼ばれ移動計算機環境では重要となる技術の一つである。

通常ハンドオフの議論は計算機が無線基地局間を移動する場合に限定して行なわれている。しかし、我々が想定している移動計算機環境では通信メディアとして使用するのは無線メディアだけではなく有線メディアなど多種多様なメディアがありこれらについても考慮する必要がある。

3.2.4 通信メディアの切替えによる通信特性の変化

上記に示したように移動計算機環境において様々な通信メディアをその時々に応じて切替えて使用する必要がある。ここで、それぞれの通信メディアの特徴について考えてみる。

有線ネットワークのイーサネットはバンド幅は 10Mbps から 100Mbps というのが現在の主流になっている。エラー率も低くネットワークに接続するには非常に有効である。無線 LAN では例えば Netwave があるが、このメディアのバンド幅は 1Mbps とイーサネット

トと比較すると非常に狭い。またエラー率においてもイーサネットより高い。電話回線を用いた PPP 接続ではバンド幅は 56Kbps 程度で非常に狭い。またエラー率も格段に高くなっている。

このように通信メディアそれぞれに異なる特性があることがわかる。もし、移動透過性が保証され、通信メディアの動的な切替えを行なった時でも通信を維持できるようになったとしても、これらの通信メディアを切替えた時の特性の変動により通信性能は極端に低下する恐れがある。つまり単にネットワークに接続するだけでは十分なネットワーク環境であるとはいえない。

この対処法として様々な研究もなされているが、それらの研究はそれぞれの通信メディアのみに対応した方法であり、通信メディアが動的に変更される移動計算機環境においてはあまり効果は期待できない。

3.3 まとめ

本章では現在の移動計算機環境における問題点として以下の項目を挙げた。

- IP アドレスの動的な変化への対応の欠如
- 多様な通信メディアへの対応の欠如
- 通信メディア切替えのタイミング
- 通信メディアの特性への適応の欠如

これらは移動計算機環境を実現する上で不可欠なものであると考えられる。次章ではこの問題点を改善する方法を提案し、それらと IETF Mobile IP を用いたネットワークアーキテクチャである JAIST Mobile IP について述べる。

第 4 章

JAIST Mobile IP

柔軟な移動計算機環境を実現するためには 3 章 2 節で述べたような問題点を解決しなければならない。本章ではそれらの問題点の改善法を提案し、IETF Mobile IP にそれらの改善法を採り入れたネットワークアーキテクチャである JAIST Mobile IP について述べる。また、これを実現する具体的な方法を本章の最後に述べる。

4.1 改善方法

現在の計算機環境では移動計算機環境を実現するには 3 章 2 節で述べたような問題点がある。本節では、それらの問題点の改善方法を示す。

IP アドレスの動的な変更

IP アドレスが動的に変化することにより通信が維持できなくなる原因は、IP アドレスが計算機の識別子としてではなくネットワーク接続点として扱われているためである。通信相手は IP アドレスを識別子として扱っているがネットワーク的には単なる接続点として使用されている。

そこで本システムでは移動計算機において明確な計算機識別子を提供する。具体的には計算機の識別子としての IP アドレスとネットワーク接続点としての IP アドレスの 2 つを割り当てるようにする。ホームネットワーク上においてのみ計算機の識別子としての IP アドレスとネットワーク接続点としての IP アドレスが一致する。つまりホームネットワーク上では従来のネットワークでの振る舞いと同様になる。このように明確な計算機の識別子を提供することで通信相手は常にそのアドレス宛てにパケットを送信するようになり、IETF MIP

での移動透過性を実現する方法と同じようにすることで、ネットワークに接続している限り通信の維持が可能となる。

多様な通信メディア

通信メディアの動的な切替えにおける問題は、ネットワークに接続するためにはリンク層、ネットワーク層の両方を制御する必要があるのに対し、リンク層レベルの制御とネットワーク層レベルの制御が独立して行なわれていることである。そのために通信メディアの切替え時の通信の維持が非常に困難になっている。

そこで本システムでは通信メディアを仮想メディア層と物理メディア層の2層に分離する。このように分離することによりプロトコルスタック側からは仮想メディアを常に使用しているように見せ、物理メディア層で実際の通信メディアの管理を行なうことを可能とする。そして今まで独立していたリンク層とネットワーク層の処理を一元的に行なう。

通信メディアの切替えのタイミング

移動計算機環境では通信メディアの効率良いハンドオフが必要である。本システムでは有線無線間、無線無線間など異なる通信メディア間でも効率の良いハンドオフを行なうための機構を提供する。この機構は環境サーバから現在使用可能な通信メディアの情報を得て、独自のポリシーに従い通信メディアの切替えを判断する。環境サーバとは個々の移動計算機内に存在し、その計算機内の資源の状態を監視しており、それらの環境情報をデータベースとして一元的に管理している。そしてアプリケーションからの要求に応じてそれらの環境情報を提供する。

通信メディアの切替えによる通信特性の変化

通信メディアの動的な切替えが発生する移動計算機環境では通信メディアの特性の相違により通信メディアを切替えると通信性能が極端に低下してしまう恐れがある。また通信メディアの切替えは計算機が稼働中に起こる。そのため通信性能を極端に低下させないため動的に通信メディアの特性に適應する機構が必要となる。

そこで本システムでは通信メディアの特性に動的に適應できる機構を提供する。ここで我々が注目したのは移動計算機がネットワークを移動したときにネットワーク全体から見ると移動計算機が使用している通信メディアが end-to-end

での通信のボトルネックになり得るということである。現状のネットワーク環境ではネットワーク基盤は比較的バンド幅やエラー率などが優れた性能のものを使用している。そのために移動計算機が無線メディアなどバンド幅の狭い通信メディアを使用しているとその部分がボトルネックになってしまう。本システムで提供する機構ではこのボトルネックとなり得る部分の通信性能を向上することにより end-to-end の通信性能の向上を計る。

4.2 JAIST Mobile IP

本稿では、柔軟な移動計算機環境を実現するために、現在の計算機環境の問題点を上記の方法を用いて改善し、これと IETF Mobile IP を組み合わせた新たなネットワークアーキテクチャとして JAIST Mobile IP を提案する。本節では JAIST Mobile IP のシステムアーキテクチャおよびそれを実現する際の設計について述べる。

4.2.1 システムアーキテクチャ

JAIST Mobile IP の全体構成を図 4.1に示す。

この図において R1,R2,R3,R4 はルータであり、それぞれのサブネットはこれらのルータを経由して基幹ネットワークに接続されている。R 2 はホームエージェント、PA はプロキシエージェントである。プロキシエージェントは IETF Mobile IP のフォーリンエージェントを拡張したものである。ホームエージェントの機能は IETF Mobile IP に準ずる。また、移動計算機宛のパケットの流れやホームエージェントへの登録におけるプロトコル、エージェント発見プロトコルも IETF Mobile IP と同様である。

通信においてことなる点は、移動計算機とプロキシエージェント間で特別なプロトコルを用いる点である。移動計算機がホームネットワーク以外のネットワークに接続する場合、そのネットワーク上のプロキシエージェントを利用することで、通信メディアの特性の変化に適応することを可能としている。例えば、エラー率の高い通信メディアを使用しているときは、移動計算機とプロキシエージェント間でローカルな再転送を行ない、エンド間通信のパフォーマンス向上をはかる。また、バンド幅の狭い通信メディアを使用している時には、パケットの圧縮を行なうことでバンド幅の有効な利用をはかる。

IETF Mobile IP における移動計算機は、ホームエージェントへの登録や移動の検知、ルーティングテーブルの変更といった機能を持つ。JAIST Mobile IP における移動計算機は、これらの機能に加えて通信メディアを動的に切替える機能、およびプロキシエージェ

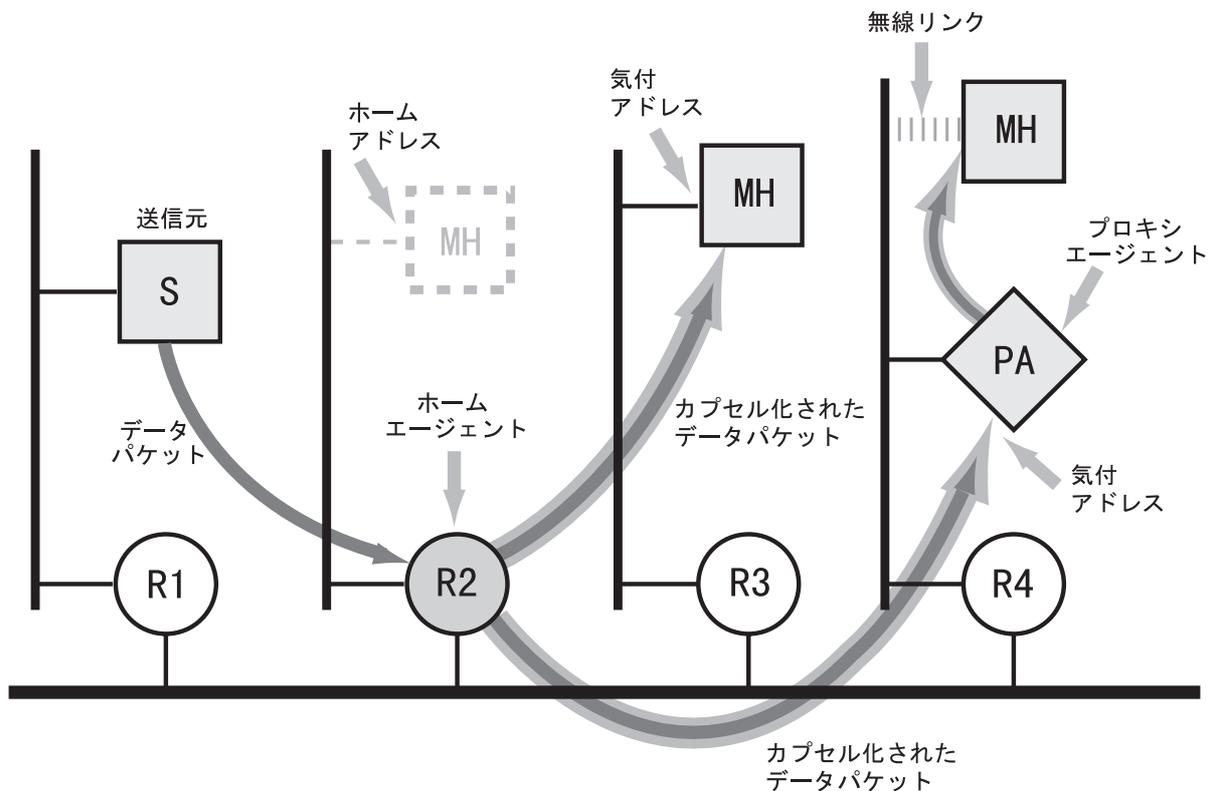


図 4.1: JAIST Mobile IP

ントへの対応としてメディア特性に適応する機能を持つ。通信メディアを切替える際に必要となる情報は後述するハンドオフコントローラが環境サーバから取得し、メディア切替の判断を下す。また、IETF Mobile IP のフォーリンエージェントに相当する機能を移動計算機内に持つことで、プロキシエージェントが存在しないネットワークへの移動も可能である。ただし、プロキシエージェントが存在しないネットワーク上では通信メディアに適応した通信を行なうことはできない。

このネットワークアーキテクチャでは、PPP やイーサネットなど異なる通信メディアも同様に扱え、また、それらの通信メディアを使用しアドレスを取得できれば計算機の移動透過性を実現できる。また、ホームネットワークとは異なるネットワークで接続した際に、そのサブネット上にプロキシエージェントが存在していれば、その時の通信メディアに適応した通信が可能となる。

4.2.2 設計

4章1節で示した改善案を利用するため移動計算機やプロキシエージェントに以下の拡張を施す。本節ではこれらについて詳細に述べる。

プロトコルスタックと通信メディアの分離

IPアドレスの動的な変更における問題点は計算機の明確な識別子の欠如であった。また多様な通信メディアでの問題点は、ネットワークに接続するためのリンク層制御とネットワーク層制御の独立であった。本稿ではこれらの問題に対処するために移動計算機に識別子を与え、通信メディアを仮想メディア層と物理メディア層に分離する手法を提案した。本システムではこれらを実現する方法として図4.2のような機構を提供する。図4.2の左側が従来のプロトコルスタックと通信メディアの関係で右側が本システムで提案するものである。

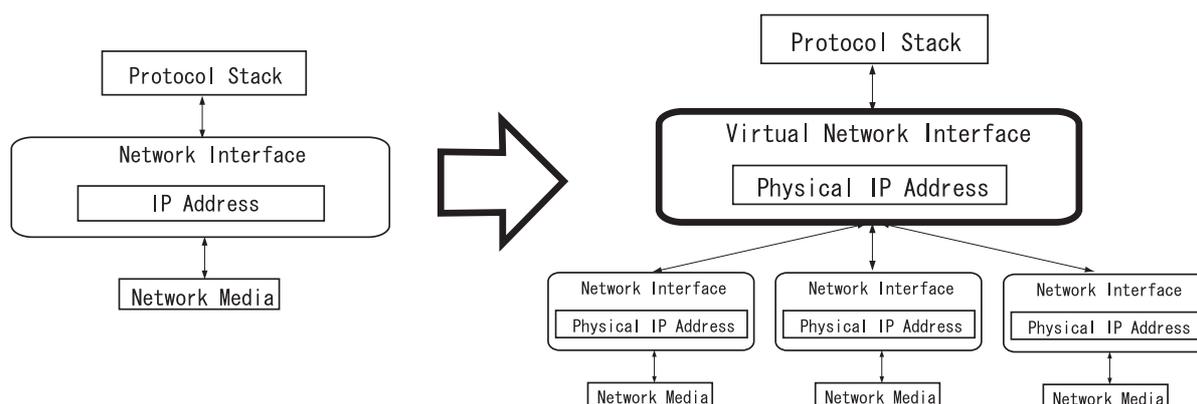


図 4.2: プロトコルスタックと通信メディアの分離

本システムではプロトコルスタックを下位の通信メディア層と仮想通信メディア層に分離する。仮想通信メディアには計算機の識別子としてのアドレスを与え、下位の通信メディアには従来と同様ネットワーク接続点としてのアドレスを与える。本システムでは計算機の移動透過性を実現するプロトコルとして IETF Mobile IP を使用するので計算機の識別子としてのアドレスにはホームアドレスを与え、ネットワーク接続点としてのアドレスは DHCP などとその都度割り当てる。これにより計算機の識別子とネットワーク接続点を明確に分離する。

また、下位の通信メディア層では複数の様々な通信メディアを管理する。この層では

ネットワークへの接続に必要な IP アドレスの確保や ARP などの処理も行なう。これにより各通信メディアの処理の違いはこの層で吸収することができる。

このように分離した利点としては、プロトコルスタックからは仮想通信メディア層を単一のネットワークインターフェースとすることができ、明確な計算機の識別子を提供することができるという点が挙げられる。さらに、プロトコルスタックからは下位の通信メディア層を隠蔽し通信メディアへのインターフェースを仮想通信メディア層のみとすることで通信メディアの切替えをプロトコルスタックからは隠蔽できる。

さらに下位の通信メディア層で通信メディア毎の処理を吸収しているために容易に通信メディアを切替えることができるようになる。このメディアの切替え機構を下位の通信メディア層と上位の仮想通信メディア層の間に挿入する。(図 4.3)

プロトコルスタックから見えるネットワークインターフェースは仮想通信メディアへのインターフェースのみなので、プロトコルスタックの経路は常にこの仮想インターフェースを使うように設定するだけで済む。実際に使用するゲートウェイなど物理通信メディア層でネットワークに接続するために行なう DHCP などを用いて決定する。また通信メディアの切替え機構が仮想通信メディア層の下位に位置しているために上位のプロトコルスタックはメディアの切替えを意識する必要がなくなる。こうすることにより計算機が移動したり使用する通信メディアを変更する場合、再構成が必要となるのは仮想ネットワーク層から物理ネットワーク層へのインターフェースを切替えるだけとなる。

通信メディア切替え機構は必要な要素が集中しており、非常にシンプルになっている。接続するネットワークや使用する通信メディアが変化したときに再構成が一箇所に集中しているため、通信メディアの切替えが非常に容易に行なうことができる。そして再構成に外部のモジュールを必要とせず、単独で環境の変化に適応できる。

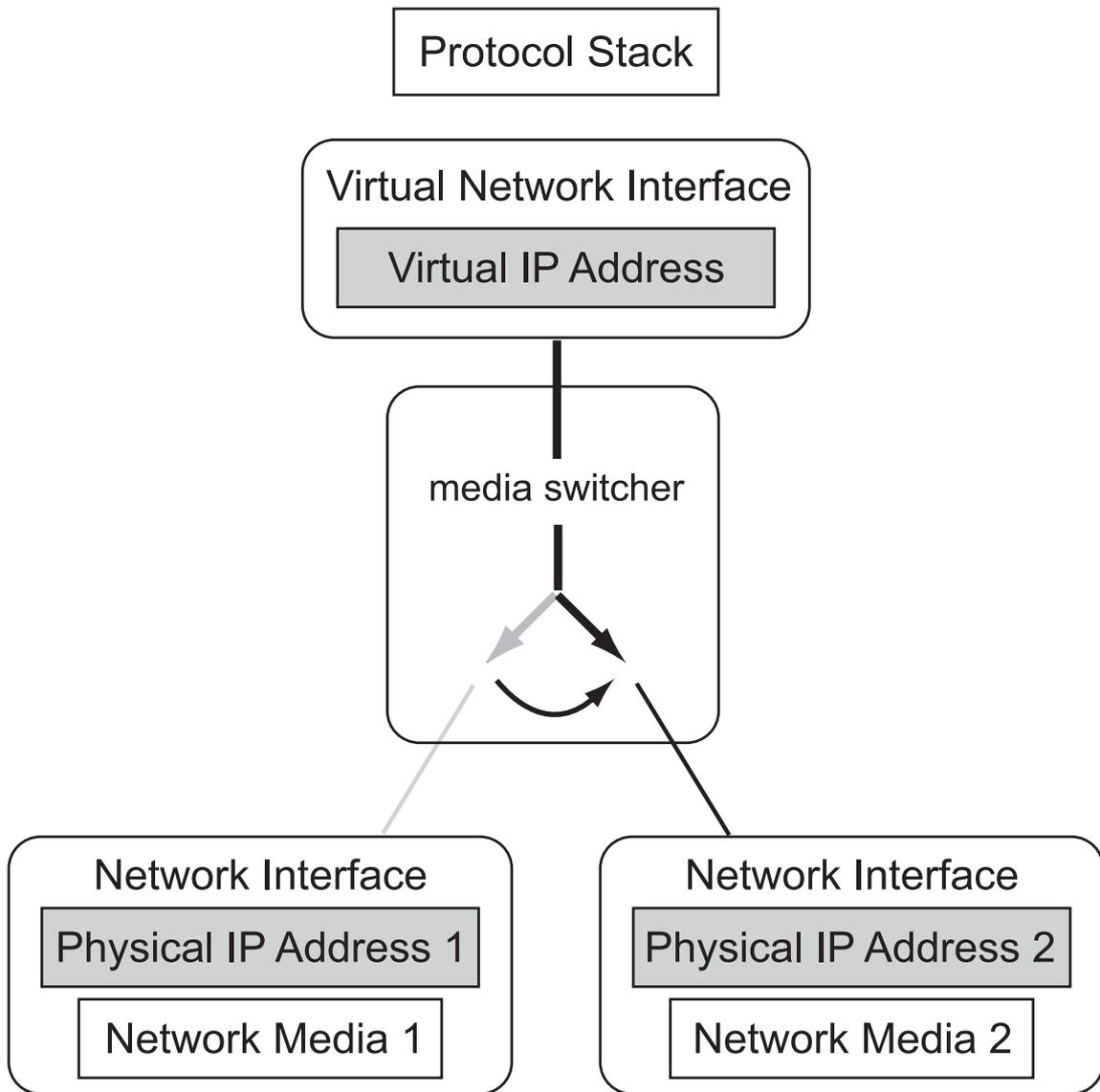


図 4.3: 通信メディア切替え機構

通信メディアの切替えのタイミング

通信メディアの切替えのハンドオフ処理の効率改善のためには通信メディアの通信状態を把握する必要がある。本システムでは通信メディアに関する情報を本研究室で提案している環境サーバから提供してもらう。そしてハンドオフコントロール機構を設け、この情報を元に通信メディアの切替えの判断を行なう。まず、環境サーバについて概要を説明する。

環境サーバ

環境サーバは計算機の資源や環境の情報をデータベースとして一括管理し、それらを統合化されたインターフェースとともにアプリケーションに提供する。アプリケーション側では個別にカーネルから情報を得る必要はなく環境サーバに問い合わせを行なうことで資源や環境など様々な情報を取得することができる。これによりアプリケーションはそれぞれの計算機のシステムに依存した情報をシステム間の差異を考慮しないで取得でき、また、インターフェースを統一することにより情報の提供を容易にできる。例えば、環境サーバが提供する情報により、現在使用中の通信メディアの種類やその時のバンド幅を知ることやバッテリーの残量が低下していること知ることが同一のインターフェースを利用して可能となる。

環境サーバが扱う情報は多種多様であり、全ての環境やその属性を統合的に扱えるようにするため、情報を階層を持ったクラスの属性として定義している。各クラスはクラス名と属性名と属性値の組からなる複数の属性を持つ。属性は属性値の取得や変更、初期化のための手続きを持つ。環境情報はクラスから生成されるオブジェクトとして実体化される。アプリケーションは環境サーバが提供するインターフェースを用いて属性の操作を行なう。操作対象となる属性は [属性名@オブジェクト名] で表される。オブジェクト名は問い合わせのための名前空間上での名前であり階層上の問い合わせ場所を示す。

アプリケーションが環境サーバから情報を提供してもらうには2通りの方法がある。1つ目は知りたい情報についてその都度環境サーバに問い合わせる方法である。2つ目の方法は知りたい情報を環境サーバに登録しておき環境サーバから定期的もしくはその情報が変化した場合に通知してもらう方法である。アプリケーションはその用途によってこれらを使い分ける必要がある。

本システムではハンドオフを効率的に行なうためにハンドオフコントロール機構を提供している。ハンドオフコントローラでは、この環境サーバからの情報を得ることにより

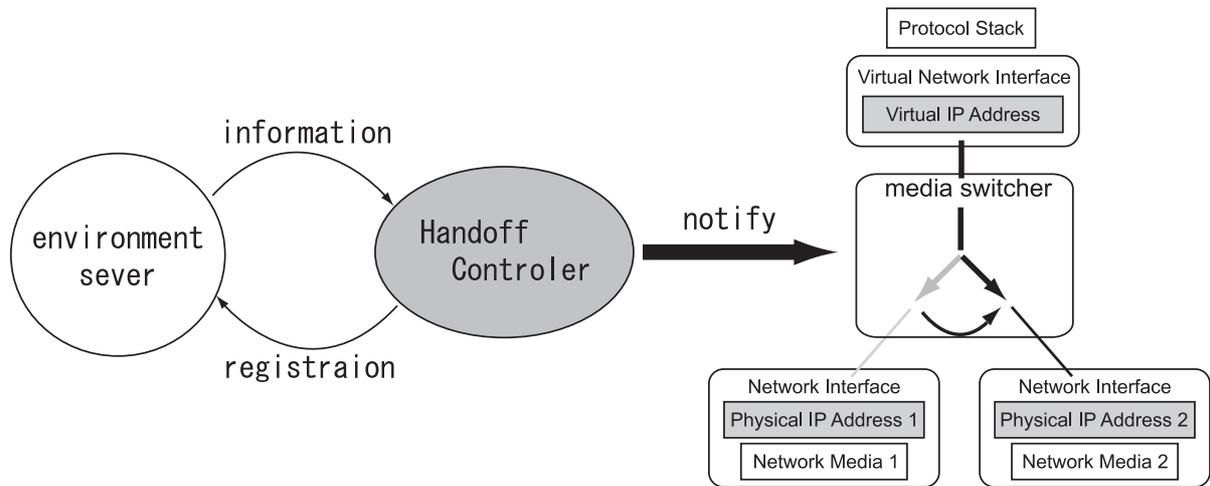


図 4.4: 環境サーバとハンドオフコントローラ

通信メディアを切替えるかどうかを判断する。この機構の概念図を図 4.4 に示す。

ハンドオフコントローラは通信メディアの情報を通知してもらうように環境サーバへ登録しておく。こうすることにより環境サーバは通信メディアの状態や通信状態が変化するとハンドオフコントローラへ通知する。ハンドオフコントローラはその通知により使用する通信メディアを判断し、もし通信メディアを切替える必要があればメディア切替え機構へ指示を出す。通信メディアの切替えを判断する時のポリシーを以下に示す。

- 有線メディアが使用可能な場合は常に有線メディアを使用する。
- 有線メディアが使用できない場合、次に優先する通信メディアは無線 LAN などの無線メディアである。ただし PPP 接続をする必要がある物は除く。
- 有線メディアや無線 LAN が使用できない場合で PHS などを利用した PPP 接続が可能な場合はそのメディアを使用する。

ハンドオフコントローラでは、これらのポリシーに基づいて行なうハンドオフ処理において発生するパケットロスを最小限にする機能も有する。例えば無線 LAN でネットワークに接続している時に基地局が切替わった、つまりサブネットが変わったという情報を環境サーバから取得すると直ちにそのネットワーク上で通信を可能とするための制御を行なう。また、PPP にはダイアリングの遅延というのが存在するためパケットロス率は他のメディアと比較すると非常に高くなってしまふ。そこで PPP へ切替える必要があると判

断した場合はダイアリングの遅延を減少させるため先行してダイヤルするといった制御も行なう。

通信メディアの切替えによる通信特性の変化

移動計算機環境では計算機の稼働中に通信メディアを切替えてネットワークに接続し続けることが可能となる。しかし、各通信メディアにはバンド幅やエラー率などでそれぞれ異なった特性があるために、end-to-endでの通信性能が極端に低下する恐れがある。本システムではこの問題を解決するために通信メディアの特性に適応する機構を提供する。

通信メディアの切替えは動的に起こるため通信メディアの特性に適応する機構も動的に変更できるような適応性の高い仕組みが必要となる。そこで本システムでは、IETF Mobile IPが規定するフォーリンエージェントの位置に着目し、移動計算機とフォーリンエージェントでパケットスヌーピングを行なう機構を提供する。パケットスヌーピングの機能を付加したフォーリンエージェントを本システムではプロキシエージェントと呼ぶ。

移動計算機がプロキシエージェントを通じて通信を行なう場合、移動計算機とプロキシエージェントの間は通信路の末端部分に相当し、無線やPHSといった貧弱な回線になることが多い。つまり、プロキシエージェントは通信路上のボトルネックの部分に位置しているため、プロキシエージェントを利用した通信メディアの特性への適応は通信路全体のパフォーマンス向上につながる。図4.5に移動計算機とプロキシエージェント内のスヌープ機構の図を示す。

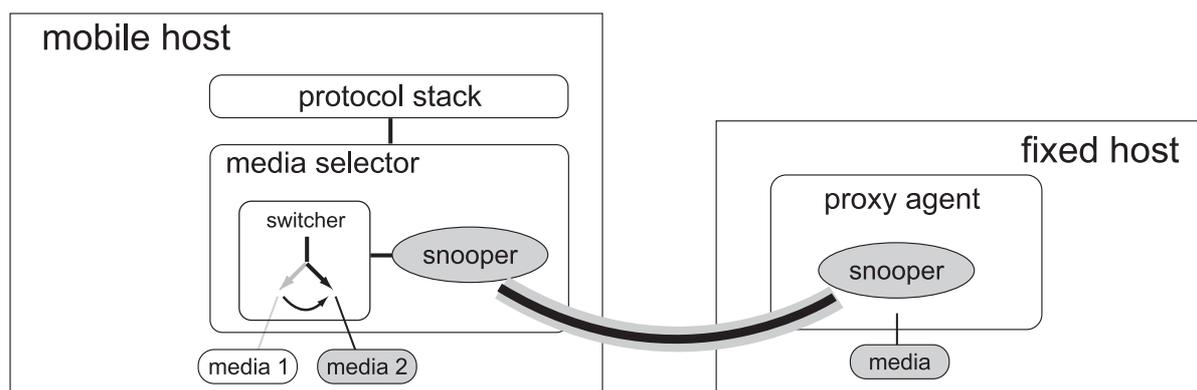


図 4.5: パケットスヌープ機構

この図の snooper というのがパケットスヌープ機構である。このスヌープ機構では移動計算機とプロキシエージェント間を行き来するパケットを監視しながら対応する処理

を行なうことで使用中の通信メディアの特性に合わせた通信を行なうことを可能としている。IETF Mobile IP では、移動計算機上のアプリケーションから送信されるパケットには何も手を加えないが、本システムではこれらのパケットにもカプセル化を施し、どちら向きのパケットも必ずプロキシーエージェントを通るよう拡張を加えている。これにより、移動計算機とプロキシーエージェント間を行き来する全ての IP パケットをスヌープ機構で監視する。またこのようにスヌープ機構を IP 層の下位に配置することで、通信メディアの特性の変化が起こったとしてもスヌープ機構での処理を切替えるだけで上位からは隠蔽して通信メディアの特性に適応できる。具体的には、それぞれの通信メディアの特性に合わせたパケット処理モジュールを用意し、これをメディアの切替えに合わせて変更するだけで良い。

4.3 まとめ

本章では問題点の改善法とそれらを IETF Mobile IP に組み込んだ JAIST Mobile IP について述べた。

第 5 章

実装

5.1 実装

4 章をもとに、RT-Mach 上に通信メディア切替え機構、通信メディアへの適応機構、ハンドオフコントローラを実装した。以下ではそれらについて述べる。

5.1.1 全体構成

本稿で提案するシステムを実現する上で中心となる移動計算機およびプロキシーエージェントの全体構成を図 5.1 に示す。

移動計算機はメディア切替え機構と環境サーバを持つ。メディア切替え機構はプロトコルスタックとデバイス間に位置し、通信メディアの選択、切替え、通信メディアへの適応を可能とし、それぞれ内部にあるハンドオフコントローラ、メディアスイッチャ、スヌーパが機能する。環境サーバは、移動計算機の環境情報をデータベースとして一元的に管理し、統一されたインターフェースによりアプリケーションにそれらの情報を提供するサーバプログラムである。メディア切替え機構では、環境サーバからの情報はハンドオフコントローラが受け取る。それらの情報をもとにハンドオフコントローラで状況に応じた通信メディアを選択する。その結果をメディアスイッチャへ通知することでメディアスイッチャは通信メディアを切替える。

通信メディアの特性に適応するための機構がスヌーパである。スヌーパは従来の IETF Mobile IP システムのフォーリンエージェントを拡張したプロキシーエージェント内と移動計算機のメディア切替え機構に提供されている。移動計算機とプロキシーエージェント内のスヌーパはお互いに協調動作することにより通信メディアへの適応を可能として

mobile host

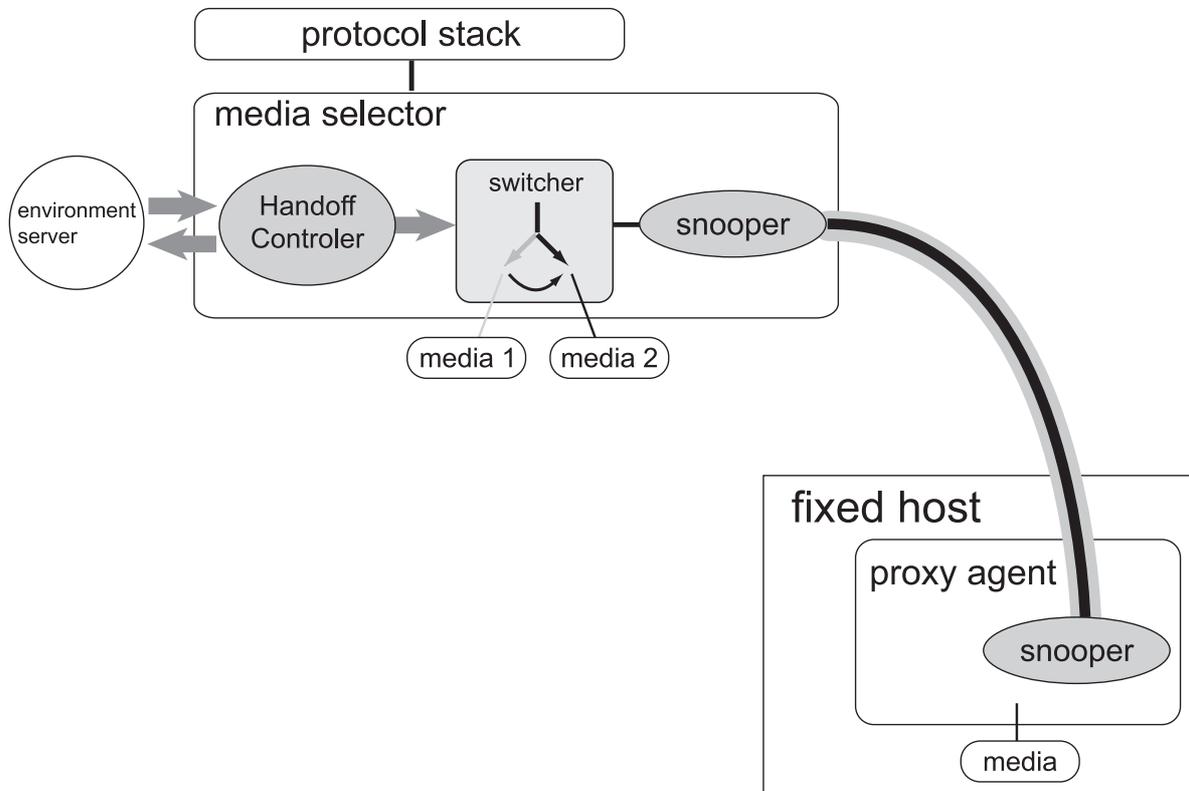


図 5.1: 全体構成

いる。

5.1.2 RT-Mach への実装

図 5.1の全体構成をもとに、メディア切替え機構とプロキシーエージェントの実装を行った。ベースとなる OS には RT-Mach マイクロカーネルと Unix サーバ (Lites) を採用した。構成を図 5.2に示す。

RT-Mach は、マイクロカーネルである。マイクロカーネルでは基本的な機構のみをカーネルサービスとして提供しており、その他のサービスはユーザレベルのサーバとして実装される。今回使用する Lites サーバは 4.4BSD のエミュレーションを行なうサーバであり、この中にプロトコルスタックなどが提供されている。このようにデバイスとプロトコルスタックが分離されているためメディア切替え機構の実装が容易になる。

メディア切替え機構はユーザレベルのプロセスとして提供するが、プロトコル階層的

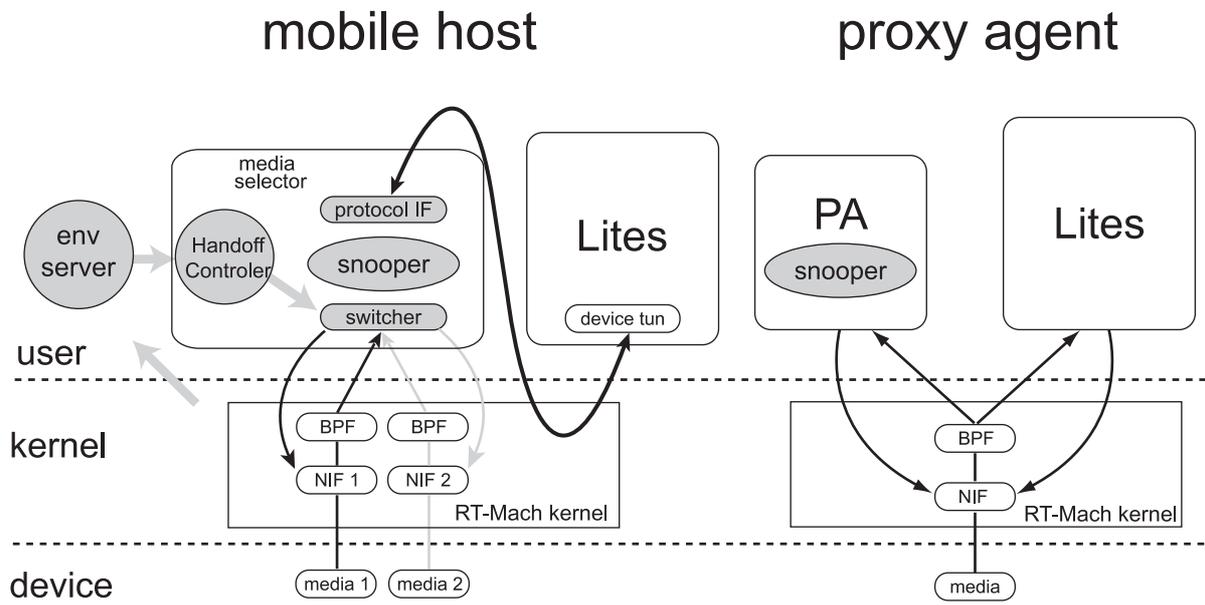


図 5.2: RT-Mach への実装

には IP 層の下に位置する。メディア切替え機構をこの位置に配置することにより、通信メディアの切替えに伴う IP アドレスの変化を上位の層から隠蔽することができ、通信メディアの仮想化が実現されている。移動計算機の Lites はトンネルデバイスを通通信メディアとして認識し、アドレスはトンネルデバイスに割り当てられたものを使用して仮想的にネットワークに接続する。こうすることにより、Lites 上のシステムは常にこのアドレスを使用して通信を行なおうとする。Lites がトンネルデバイスに書き込んだパケットはメディア切替え機構によって取り込まれ、内部で適切な処理を行なった後、ネットワークインターフェースへ渡される。ネットワークインターフェースからのパケットは内部で処理を行ないトンネルデバイスへ渡される。

この実装方式の利点としてはメディア切替え機構をプロトコルスタックより下位に位置することにより従来のネットワークアプリケーションをそのまま利用できることである。また、ユーザレベルのプロセスとして実装されているため拡張や変更が容易にできるといった利点もある。

プロキシエージェントも移動計算機でのメディア切替え機構と同様に RT-Mach 上に実装する。このプロセスでは、RT-Mach のカーネルレベルのパケットフィルタを用いることにより必要なパケットをフィルタリングすることで Lites とは完全に独立したものとして実装されている。

以下でメディア切替え機構とプロキシエージェントの内部構成について述べる。

メディア切替え機構

図 5.3 にメディア切替え機構の内部構成を示す。メディア切替え機構はプロトコルインターフェース、スイッチャ、デバイスインターフェースから構成される。

デバイスインターフェースはカーネルにあるデバイスとのパケットの受渡しを行なうインターフェースである。デバイスドライバとの入出力は RT-Mach マイクロカーネルのパケットフィルタを利用して直接接続して行なう。またデバイスインターフェースは異なる通信メディア毎に用意されており、各通信メディアに特化した処理 (イーサネットでの ARP など) やアドレスの取得はここで行なわれる。スイッチャからデバイスインターフェースへのインターフェースを統一し、IP パケットのやりとりをすることにより、スイッチャはリンク層を意識する必要がなくなり、イーサネットや PPP など異なるメディアを同等に扱うことができる。

プロトコルインターフェースは Lites のトンネルデバイスとの入出力を行なうインターフェースを提供する。この他にトンネルデバイスの初期化や終了処理も行なう。

スイッチャの主な機能は、プロトコルインターフェースとデバイスインターフェースを結びつけることである。スイッチャはデバイスインターフェースからプロトコルインターフェースへのパケットの送信とプロトコルインターフェースからデバイスインターフェースへのパケットの送信といった処理を行なう 2 つのスレッドを持つ。スイッチャのその他の機能としては、移動先のネットワーク上にプロキシエージェントが存在しない時にホームエージェントへの登録を行なうためのプロキシエージェントの最低限の機能を持つ。これは ICMP ルータディスカバリやレジストレーションメッセージの中継を行なう。また、ハンドオフコントローラからの通知によりデバイスインターフェースの切替えも行なう。切替える際に 2 つのスレッドを同時に切替える必要があるためミュテックスとコンディション変数を用いた同期機構を用意している。

移動先ネットワークにプロキシエージェントが存在する場合は、移動計算機が使用している通信メディアの特性に適應するためスヌーパを使用する。スヌーパを使用する時の内部構成を図 5.4 に示す。

移動計算機内のスヌーパはデバイスインターフェースとプロトコルインターフェースの間に位置する。スヌーパへのインターフェースは upSnoop と downSnoop の二つがある。upSnoop はデバイスインターフェースからプロトコルインターフェースへ流れるパケットを受け取る。downSnoop は逆にプロトコルインターフェースからデバイスインターフェースへのパケットを受け取る。正確にはこれらのインターフェースはスイッチャにあ

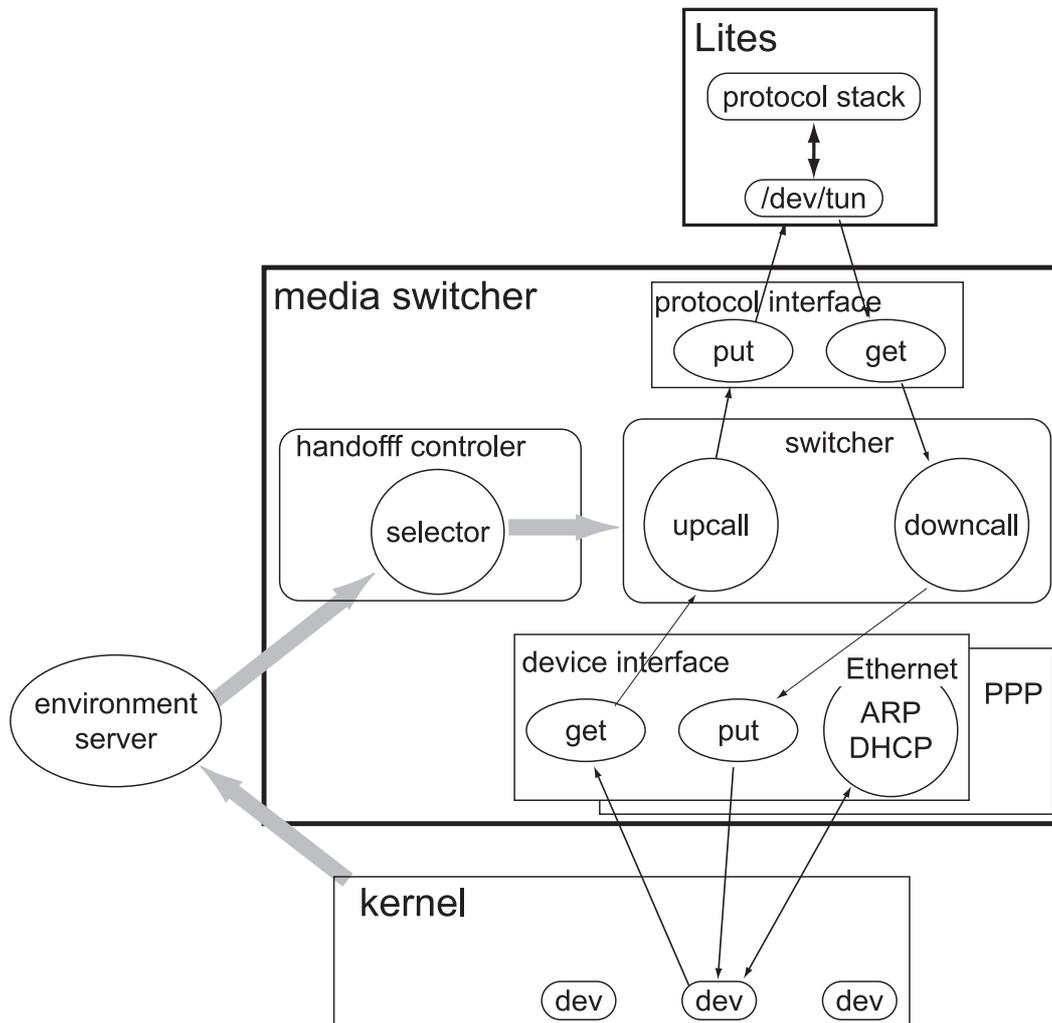


図 5.3: メディア切替え機構

る upcall と downcall のスレッドに提供している。スヌープには upSnoop と downSnoop の 2 つのスレッド 以外に様々な通信メディアへの適応モジュールがある。これらのインターフェースはそれぞれのモジュールで統一されているため、新たなモジュールの追加や変更は容易に行なうことができる。現在実装したモジュールは、PHS 等のバンド幅の狭い通信メディアを利用する時に使用するパケット圧縮モジュールと無線 LAN 等のエラー率の高い通信メディアを利用する時に使用するエラーリカバリモジュールの 2 つである。

upSnoop は、アップコールから受け取ったパケットのヘッダをチェックし、これを元にしたどのモジュールを使用するかを判断する。これと同時にこのパケットを元の IP パケットに戻しプロトコルインターフェースに渡す。downSnoop はダウンコールから受け取った

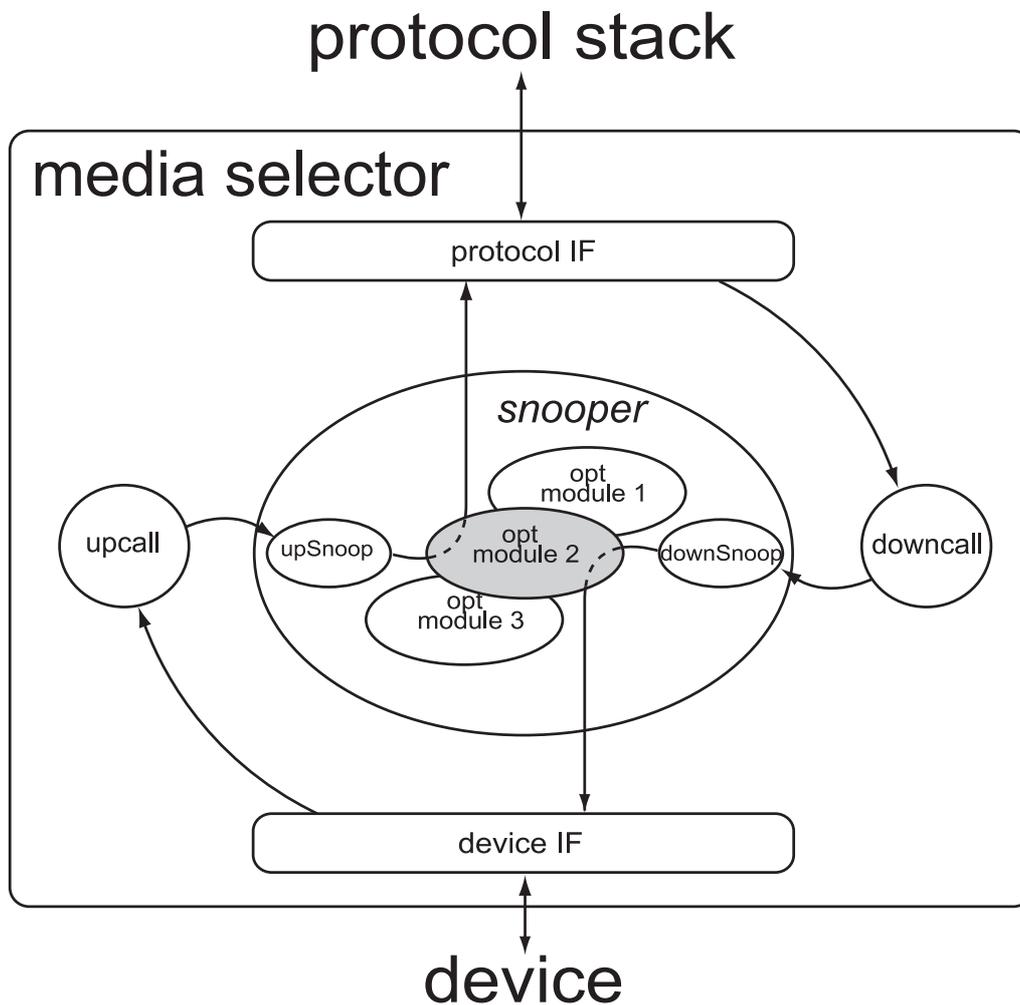


図 5.4: 移動計算機内のスヌーパ

IP パケットにプロキシエージェントへ送るためのヘッダを付加し、現在使用している通信メディアの特性により使用するモジュールを判断し処理を行なう。そしてそのパケットをデバイスインターフェースへ渡す。

プロキシエージェント

図 5.5にプロキシエージェントの内部構成を示す。プロキシエージェントは Mobile IP での登録用のスレッド、ICMP ルータディスカバリへの返答用のスレッド、基本的には MHsnoop と HAsnoop の 2 つの手続きおよび様々なメディア適応モジュールから構成される。

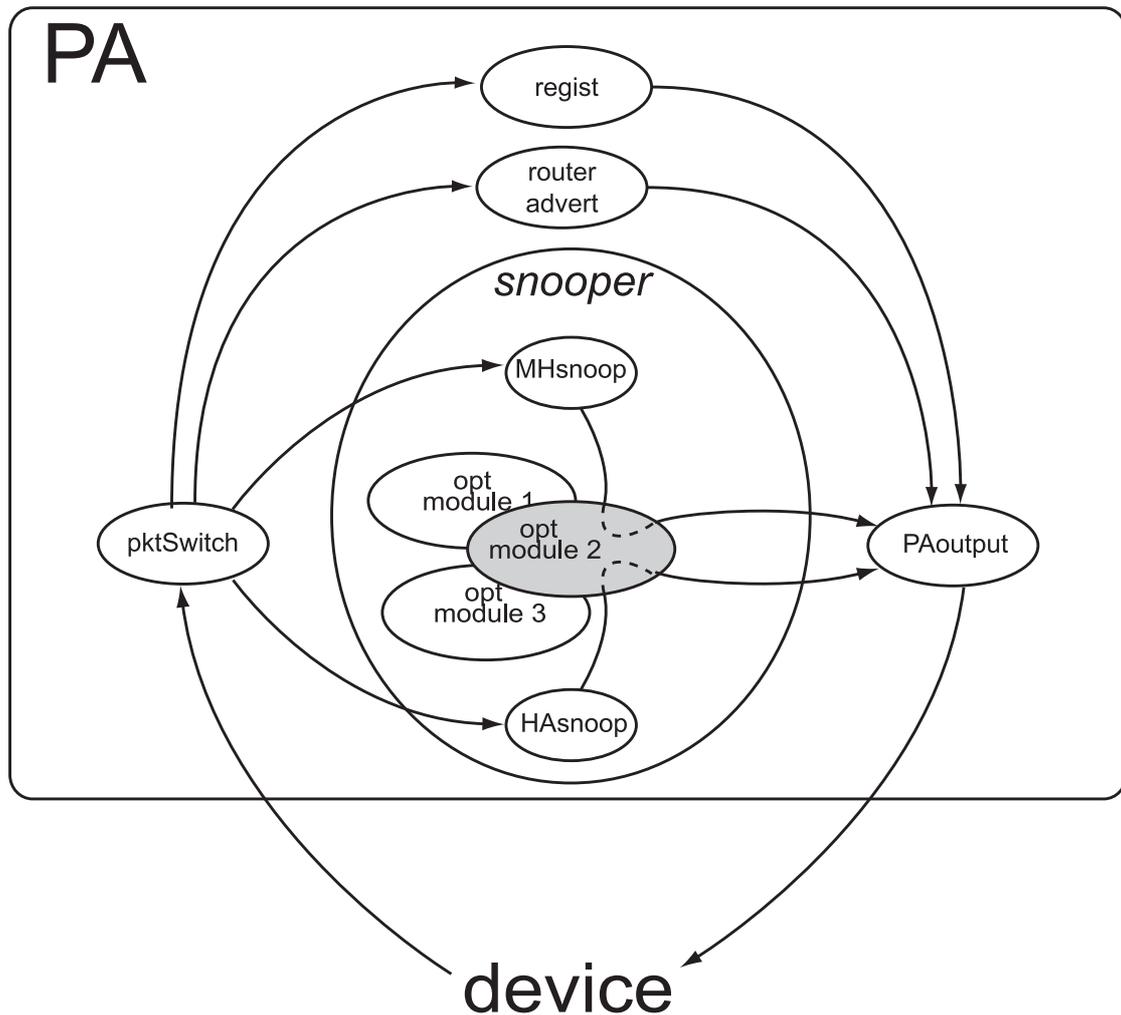


図 5.5: プロキシエージェント

プロキシエージェントは RT-Mach マイクロカーネルのパケットフィルタを用いてデバイスと直接繋がっている。そのため Lites サーバとは完全に独立して動作している。最初にプロキシエージェント内の手続き pktSwitch が必要なパケット取り込み、そのパケットに対応する手続きへ振り分けてる。パケットが移動計算機からの ICMP ルータ懇願である場合は図の router advert スレッドへ、移動計算機やホームエージェントからの登録パケットの時は図の regist スレッドへ、移動計算機からの適応処理を施されたパケットは MHsnoop へ、ホームエージェントからの IPIP パケットは HAsnoop へそれぞれ渡される。

スヌーパ内のメディア適応モジュールを機能ごとに分け、インターフェースを統一する

ことによってこれらの切替えが非常に容易になり移動計算機ごとに適応するのではなくパケットごとに適応する可能となっている。このため移動計算機が使用している通信メディアが通信中に状態が変化してもそれに動的に適応することができる。

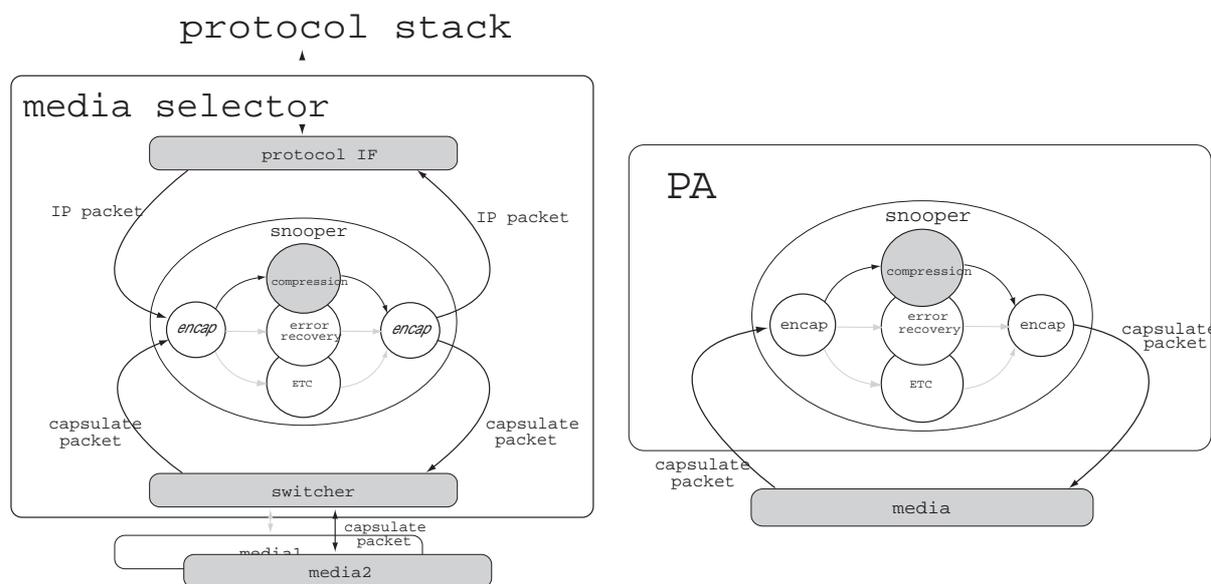


図 5.6: パケットスヌープ機構の構造

メディア切替機構およびプロキシエージェント内のスヌーパの構造を図 5.6に示す。カプセル化されたパケットがスヌーパ内を通過する際に、状況に応じて適切なモジュールに渡すことで、メディアの特性を考慮した通信をするための処理を行なう。これらのモジュールは統一されたインターフェースを持ち、新しいモジュールの追加、変更は非常に容易である。現状で考案しているモジュールは、パケット圧縮モジュールとエラーリカバリモジュールである。

パケット圧縮モジュール

パケット圧縮モジュールは、PHS などバンド幅の狭い通信メディアを使用している場合に、移動計算機とプロキシエージェント間の双方向のパケットに圧縮を施し、回線を通るデータのサイズを小さくすることでバンド幅の有効利用する。このモジュールは圧縮処理によるタイムロスが生じてもバンド幅を十分に使い切ることができるような状況で使用した場合に効果がある。また、TCP や UDP のポート番号などのヘッダの情報からペイロードの特性が予測で

きるパケットに関してはそれに応じて、ヘッダ圧縮、ペイロード圧縮、パケット連結などの各種パケット圧縮法を使い分ける。

エラーリカバリモジュール

エラーリカバリモジュールは、無線 LAN 等、エラー率の高い回線に接続している場合に、TCP のデータ転送において高速なエラー回復を可能にするため、移動計算機とプロキシエージェント間でローカルな再転送を行なうモジュールである。

従来の TCP はパケットロスの主要因が輻輳にあるような有線ネットワークで十分に機能するように設計されている。一方、近年登場して来た無線メディアにはビットエラー率が有線の場合よりも高くなるという特徴があり、無線メディアを利用したネットワークのパケットロスは多くの場合このビットエラーによって生じている。このような特徴を持つネットワークで従来の TCP をそのまま用いると、輻輳回避や高速リカバリのアルゴリズムが不必要に働いて通信性能を低減させてしまう可能性がある。そこで、メディアセクタと PA プロセス内のスヌーパにエラーリカバリモジュールを持たせ、移動計算機とプロキシエージェント間でローカルな再転送を可能にする。つまり、移動計算機とプロキシエージェント間で生じたパケットロスを通信相手の通信相手の TCP に対してできる限り隠し、従来の TCP が持つ最適化メカニズムが不必要に起こるのを防ぐ。

基本的には TCP のデータパケットをキャッシングし、確認応答パケットを監視することで失われたパケットのローカルな再転送を可能にする。移動計算機へ向かうデータのエラーリカバリには、プロキシエージェント内のエラーリカバリモジュールに、移動計算機から確認応答が届いていないパケットのキャッシングを行なう機能と、移動計算機からの確認応答を監視する機能を持たせることでこれを実現する。TCP は順番狂って到着パケットに対して重複確認応答を返す。そこで、プロキシエージェントのエラーリカバリモジュールは、移動計算機から返された返された重複確認応答をもとに失われたパケットを特定し、データパケットキャッシュの中から対応するパケットを再転送する。また、移動計算機からの通常の確認応答は、キャッシュに蓄えているパケットをクリアするためのイベントとなる。移動計算機から送信されるデータのエラーリカバリには、プロキシエージェント内のエラーリカバリモジュールに TCP データのシーケンス番号を監視させることで対処する。パケットロスを検出し

た場合、移動計算機の TCP に対して選択的確認応答を送り、対応するパケットを再転送させる。

第 6 章

評価と考察

本章では本稿で提案する JAIST Mobile IP と SONY CSL で提案されている VIP、Mobile IP の標準として認可された IETF Mobile IP の手法について様々な機能の比較を行なう。

また、RT-Mach 上に実装したメディア切替え機構を用いた場合と従来の Mobile IP との性能差を比較する。従来の Mobile IP としてカーネギメロン大学で作成された Mobile IP を使用する。この Mobile IP は IETF Mobile IP の仕様に従って実装されている。

これらの測定では移動ホストに Thinkpad 560 を使用した。ホームエージェントは FreeBSD2.2.7 上で動作しており、その他の移動ホストやフォーリンエージェント、プロキシエージェントは RT-Mach 上で動作させている。通信メディアとしてはイーサネットと無線 LAN の Netwave を使用した。

6.1 3 種類の手法の比較

表 6.1 に JAIST Mobile IP、IETF Mobile IP、VIP の機能の比較を示す。移動透過性についてはそれぞれのシステムで実現されている。しかし、IETF Mobile IP ではホスト識別子を提供しているがネットワーク接続点としても利用しておりそのために通常のルーティングでは通信が不可能になるといった問題がある。その点では JAIST Mobile IP、VIP は計算機識別子を明確に提供しており、通常のルーティングで通信ができる。

メディアの切替えについては VIP、IETF Mobile IP では厳密に仕様が定められていない。そのためにメディアの切替えとこれらのシステムが別次元となっている。現在のネットワーク環境では通信メディアとネットワークへの接続は非常に密接な関係にありそれらを同時に実現するシステムにする必要があると思われる。JAIST Mobile IP では通信メディア切替え機構を提供することでこれらを可能としている。

移動先の環境やその時の状況に応じて通信メディアを切替えてネットワークに接続する際に、通信メディアの特性の変化により通信性能が極端に低下することが予想される。IETF Mobile IP、VIP ではこれらの特性については何も考慮しておらずネットワークへの接続性のみ重点を置いている。JAIST Mobile IP ではプロキシエージェントと移動計算機にスヌーパを提供し、それらが協調動作することにより通信メディアの特性への適応を可能としている。IETF Mobile IP ではフォーリンエージェントを拡張することでこれらを行なうことは将来的には可能であると思われる。しかし、VIP ではフォーリンエージェントやプロキシエージェントのようなサービスを行なうホストを必要としていないために移動先に存在する計算機と協調動作することを実現するのは難しい。

VIP では拡散キャッシュ法を用いることにより VIP の拡張を施されたルータなどの計算機内でアドレス変換テーブル (AMT:Address Mapping Table) が動的に変更されるためパケットのルーティングが最適化される。しかし、JAIST Mobile IP や IETF Mobile IP ではこのような仕組みが無く三角ルーティングと呼ばれる問題が生じる。これの対処法としては IP のオプションであるソースルートオプションを利用する方法が考えられる。中継するルータがソースルートオプションをサポートしていれば、冗長なルーティングがなくなる。また、IP v6 上で実装することによりこの問題を解決することも可能である。しかし、システム側ではサポートしていないため表 6.1 ではルーティングに関しては×にしている。

表 6.1: 3 種類の手法の比較

	移動透過性	ホスト識別子	メディア切替え	通信メディアの特性への適応	ルーティング
JAIST Mobile IP					×
IETF Mobile IP				×	×
VIP				×	

このように機能の面では JAIST Mobile IP は非常に有用であると思われる。これから

の移動計算機環境では移動透過性だけではなく様々な通信メディアとの連携が必要である。そうすることにより通信メディアの切替えや計算機の移動を行なったとしても、それらの通信メディアの特性に適応することが可能となる。

6.2 IETF Mobile IP との性能比較

図 6.1のような実験環境で IETF Mobile IP の実装例である CMU Mobile IP [15] と JAIST Mobile IP との比較を行なった。

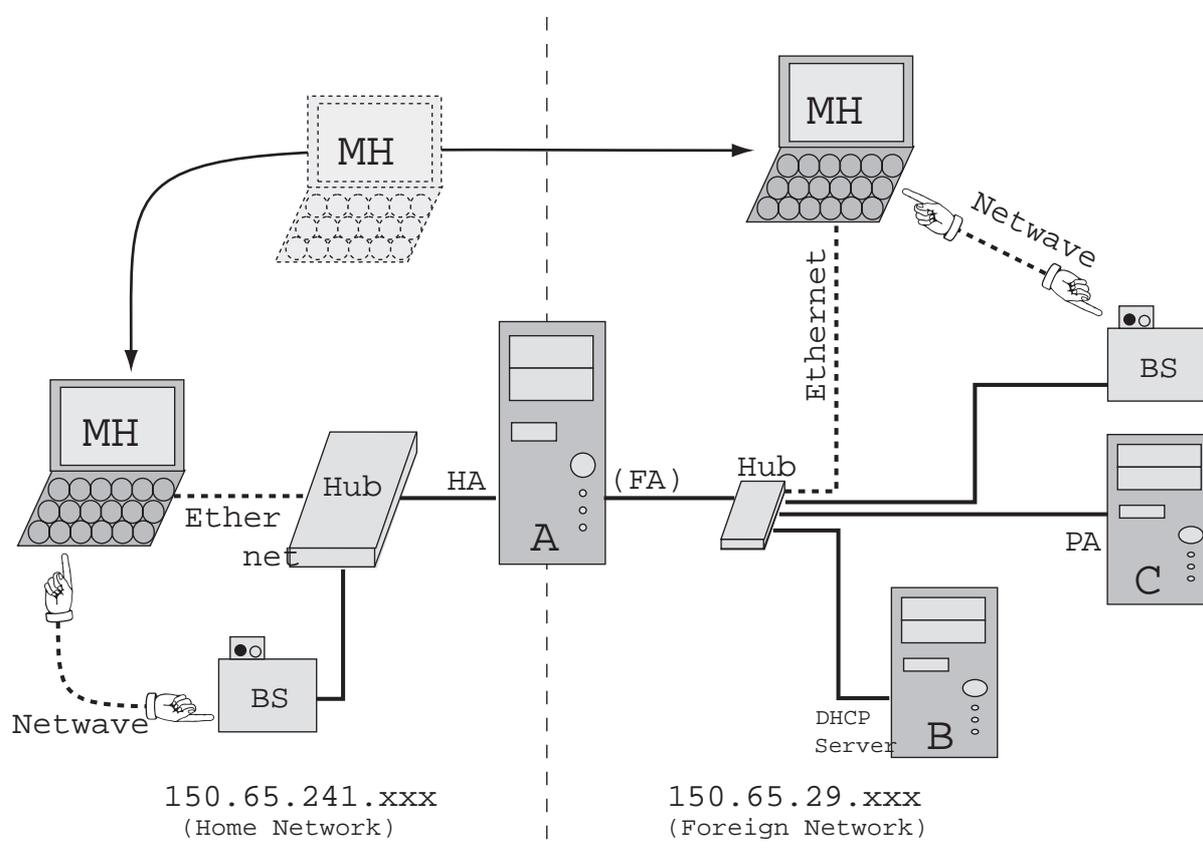


図 6.1: 実験環境

実験環境ではサブネットをサブネット I (150.65.241.xxx) とサブネット II (150.65.29.xxx) の 2 つを用意した。図 6.1 の点線よりも左側がサブネット I のサブネットで、点線よりも右側がサブネット II のサブネットである。また、移動計算機のホームネットワークはサブネット I としている。ホスト A は、サブネット I のネットワークに対してホームエージェント

ント (HA) として動作する。また、ホスト A はサブネット I とサブネット II を結ぶルータともなっている。ホスト B は、DHCP サーバとして動作している。JAIST Mobile IP で移動計算機がアドレスを取って来る際にこのサーバを利用する。ホスト C は、JAIST Mobile IP で利用するプロキシエージェントとして動作する。また CMU MOBILE IP を動作させる時にはフォーリンエージェント (FA) として動作する。このフォーリンエージェントは RT-Mach 上で動作するように改良を加えてある。移動計算機は、ネットワークに接続するのに通信メディアとして無線メディアの Netwave と有線メディアのイーサネットを使用する。また、その他のホストはハブを介して 10base-T のイーサネットで接続されている。図中の BS は無線メディアの基地局でありバンド幅は 1M である。

表 6.2: 計算機のスペック

	OS	CPU	Memory
移動計算機 (MH)	RT-Mach+Lites	Pentium 133MHz	40M
ホスト A	FreeBSD 2.2.7	Pentium 120MHz	32M
ホスト B	FreeBSD 2.2.7	486DX4	16M
ホスト C	RT-Mach+Lites	PentiumII 450MHz	512M

それぞれの計算機のスペックは、表 6.2 のようになっている。この環境で移動計算機がホームネットワーク、フォーリンネットワーク上にいるとき、ベンチマークの一つである `ttcp` を用いて、ホスト B から移動計算機へ `tcp` のパケットを流したときの時間を測定した。特に、JAIST Mobile IP ではフォーリンネットワーク上において、プロキシエージェントを使用しない時、プロキシエージェントを使用する場合 (最適化モジュールなし、エラーリカバリモジュール使用) について測定した。通信メディアとして Netwave を用いる場合は、送信したファイルサイズを 409,600 バイト、10Base-T のイーサネットを用いる場合は 1,638,400 バイトとした。この結果を表 6.2 に示す。また、通信メディア切替え機構内での `upcall` (パケットを送信) と `downcall` (パケットを受信) でのパケットを取り出しから出力するまでの時間を Pentium プロセッサが持つタイムスタンプカウンターを用いて測定した。この結果を表 6.2 に示す。さらにプロキシエージェントでのパケット処理時間を最適化モジュールを使用しない、エラーリカバリモジュールを使用する時について測定した。この結果を表 6.2 に示す。

表 6.2 を見ると通信メディアとしてイーサネットを使用した場合、ホームネットワーク

表 6.3: 測定結果

	CMU Mobile IP		JAIST Mobile IP			
	home	foreign	home	PA なし	PA あり (モジュールなし)	PA あり (エラーリカバリ)
イーサネット (sec)	3.16	3.93	5.11	6.6	7.3	-
Netwave (sec)	8.66	9.52	9.95	9.52	9.53	11.9

表 6.4: メディア切替え機構でのパケット処理の時間

	home	PA なし	PA あり (モジュールなし)	PA あり (エラーリカバリ)
upcall (μ sec)	10.2	16.7	18.8	22.6
downcall (μ sec)	113	94.7	102	103

上では CMU Mobile IP での通信速度が JAIST Mobile IP と比べて約 1.6 倍となっている。フォーリンネットワーク上では PA を使わない場合は約 1.7 倍、PA を使った場合は約 1.9 倍となっている。これらの時にメディア切替え機構およびプロキシエージェントに流れた移動計算機宛てのパケットの総数は約 1243 であった。また、移動計算機から送信されたパケット数は約 291 であった。この結果と表 6.2、表 6.2 の結果よりホームネットワークでの送信、受信でのパケット処理のオーバーヘッドは約 45.6 ミリ秒であり、フォーリンネットワークで PA を使用しない時のオーバーヘッドは約 48.3 ミリ秒で、PA を使用しモジュールを使用しない時のパケット処理のオーバーヘッドは約 53.1 ミリ秒である。

CMU Mobile IP の時の時間にこれらのオーバーヘッドを単純に加えただけでは 6.2 を見ると JAIST Mobile IP での結果になっていない。今回実装したメディア切替え機構では

表 6.5: プロキシエージェントでのパケット処理の時間

	モジュールなし	エラーリカバリ
MH→PA (μ sec)	10.9	13.0
HA→PA (μ sec)	14.1	21.1

サーバ方式をとっているために RT-Mach カーネルとメディア切替え機構、Lites 間でのパケットの入出力が非常に多くこれがオーバーヘッドの要因になっている。ペンティアムカウンタで RT-Mach カーネルからパケットを送受信する時と Lites のトンネルデバイスからパケットを送受信する時の時間を測定した。結果を表 6.2 に示す。

表 6.6: デバイスとのパケット送受信の時間

	デバイスから 取ってくる	デバイスへ 置く	トンネルから 取ってくる	トンネルへ 置く
時間 (μ sec)	92.5	413	365	620

この表と先ほど述べたパケット総数よりカーネルからメディア切替え機構、そして Lites へという一連の処理の流れでのデバイスとのやりとりでのオーバーヘッドは約 885msec、また、逆の流れでのデバイスとのやりとりでのオーバーヘッドは約 226msec となる。

これ以外のオーバーヘッドとしては今回の実装では受信用、送信用の 2 つのスレッドを使っており、これらのスレッドの同期の時間が考えられる。スレッドの同期には約 69 μ sec かかり、今回の測定では約 1600 回同期を行っていた。つまり、同期でのオーバーヘッドは約 110msec である。

このように、JAIST Mobile IP でのネットワークアーキテクチャによるオーバーヘッドはほぼ無視でき、また、それ以外のオーバーヘッドは今回のアーキテクチャによるものではなく実装上でのオーバーヘッドであることがわかる。

通信メディアとして Netwave を使用した場合は、イーサネットを使用した場合とは異なり CMU Mobile IP と JAIST Mobile IP との通信速度に差がほとんどない。これは、Netwave の通信速度が遅いためパケット処理の時間よりも通信の速度のほうがボトルネックになっているために起こっていると考えられる。イーサネットの場合は通信の速度が十分に速いためパケット処理の時間がボトルネックになっている。

JAIST Mobile IP で移動計算機がフォーリンネットワーク上へ移動し通信メディアとして Netwave を使用して、かつ、プロキシエージェントと協調動作する時にエラーリカバリモジュールを使用しているにもかかわらず表 6.2を見ると通信性能が改善されていない。これは実験の際に無線基地局と移動計算機間の距離が近く、エラーがほとんど起こらなかったために、エラーリカバリモジュールの機能が十分に活かされていないことが原因であると思われる。

また、全体を通してみると JAIST Mobile IP でのオーバーヘッドはカーネルとの入出力の処理にかかる時間に影響されていることがわかる。今回実装したメディア切替え機構はサーバ方式で実装している。本研究ではシステムの拡張性と実装の容易さを考慮した上でこの方式を選択した。この方式ではメディア切替え機構はユーザプロセスとして動作するため非常に拡張性が高くなっているが、一方ではカーネルとの入出力回数やプロセスの切替えが増加するという欠点がある。カーネルとユーザプロセスとの入出力やプロセスの切替えには非常に時間がかかってしまう。今回実装した JAIST Mobile IP でのオーバーヘッドの原因はこのサーバ方式を選択したためにカーネルとの入出力回数が増加し、かつプロセスの切替えが多発したためであると考えられる。そして、スレッドの同期などもオーバーヘッドの原因となっている。

6.3 まとめ

本章では JAIST Mobile IP と IETF Mobile IP、SONY CSL の VIP と比較を行ない本システムの有用性について検討した。また、CMU で実装された Mobile IP と JAIST Mobile IP とで通信性能について比較を行ない本システムのパフォーマンスについて検討した。

IETF Mobile IP や VIP では移動透過性の提供に重点をおいていたが、JAIST Mobile IP では移動透過性の実現からはじまり、通信メディアの切替え、通信メディアの特性への適応と移動計算機環境を実現する上で重要となる機能を提供している。

パフォーマンスの面ではサーバ方式を選択したためカーネルとの入出力のためのオーバーヘッドが増加し CMU Mobile IP などと比較すると悪くなっている。しかし、Netwave な

ど通信速度が遅い通信メディアにおいてはメディア切替え機構などでのオーバーヘッドより通信そのものがボトルネックとなっているためパフォーマンスはほぼ同じとなった。

以上のことより今回実装したサーバ方式での JAIST Mobile IP はイーサネットなど通信速度が速い通信メディアを使用しているとパフォーマンスは低下してしまうが様々な機能を有しており移動計算機環境では有用であると思われる。

第 7 章

まとめと今後の課題

現在の移動計算機環境では通信メディアの変化や計算機の移動により通信を維持できないといった問題がある。本稿では計算機の移動や通信メディアの変化を隠蔽するメディアの切替え機構を提案し IETF Mobile IP と組み合わせることによって計算機の移動透過性を実現するシステムを提案した。この際に以下の 4 つの問題点に焦点を当てた。

- 明確な計算機識別子の欠如
- 多様な通信メディアへの対応
- メディア切替えのタイミング
- メディア切替えによる特性の変化

これらの問題点を改善するポイントとなったのは計算機の識別子の提供と通信メディアの仮想化、環境サーバの利用、フォーリンエージェントの拡張である。これらのための機構を提供することにより上記の 4 つの問題点を改善した。

通信メディアの切替え機構や我々の研究室で提案している環境サーバからの通信メディアの情報をもとに無線メディア間だけでなく有線メディアと無線メディア間など様々なメディア間でのハンドオフ処理を効率良く行なう機構や IETF Mobile IP でのフォーリンエージェントを拡張したプロキシエージェントと移動計算機間で移動計算機が使用している通信メディアの特性に適応できる機構と組み合わせることにより通信効率の改善を可能とした。これらのシステムを RT-Mach 上に実装し、イーサネット、無線 LAN、PPP など様々なメディアの切替えや計算機の移動を行なっても通信が維持でき、なおかつ通信メディアの特性に適応した通信を行なうことができた。

しかしいくつかの問題も残されている。ファイアーウォールを使用しているネットワークでは本システムの性質上通信が行なえなくなる恐れがある。これは本システムでは移動ホストが送信するパケットのソースアドレスがホームアドレスとなっているために起こる。これを改善するための機構が必要である。

また、通信メディアへ適応するためのモジュールとしてはエラーリカバリモジュールと圧縮モジュールを用意しているがその他の特性を持つ通信メディアのためのモジュールについて検討して新たなモジュールを取り入れる必要がある。

これらのモジュールは現在では移動計算機とプロキシエージェントの両方が持つておく必要があるが新たなモジュールを追加する際に全てのプロキシエージェントに追加するのでは非常に手間がかかってしまう。そこで、プロキシエージェントが最初からモジュールを持っているのではなく移動計算機がプロキシエージェントに提供する方法についても検討する必要がある。

移動計算機環境ではセキュリティについても検討しなければならない。現在実装したJAIST Mobile IP では従来の Mobile IP が提供している認証機構のみを利用している。そのため移動計算機とプロキシエージェント間ではセキュリティについてはほとんど考慮していない。そのために「なりすまし」などが非常に容易に行なえるため実用的ではない。実用化するにはこれらを防ぐための認証機構を導入する必要がある。

今後はこれらの問題を改善し、より柔軟な移動計算機環境を提供する。

参考文献

- [1] 中島 達夫, “モバイルコンピューティングのための動的適応可能なソフトウェアアーキテクチャ”, 第 16 回 IPA 技術発表会, 1997.
- [2] 中島 達夫, “モバイルコンピューティング” 無線 LAN アーキテクチャ, 第 5 章, 共立出版, 1997.
- [3] 小林 勝, 中島達夫, “動的なメディア選択が可能な Mobile IP の設計と実装”, IPSJ OS, February, 1998.
- [4] Mary G.Baker, Xinhua Zhao, Stuart Cheshire, Jonathan Stone, “Supporting Mobility in MosquitoNet”, Proceedings of the 1996 USENIX Technical Conference, San Diego, CA, January, 1996.
- [5] C. Perkins, “IP Mobility Support”, RFC2002, October, 1996.
- [6] C. Perkins, “IP Encapsulation within IP”, RFC2003, October, 1996.
- [7] F. Teraoka, Y. Yokote and M. Tokoro “A Network Architecture Providing Host Migration Transparency”, ACM SIGCOMM’91, 1991.
- [8] Jon Inouye, Jim Binkley, Jonathan Walpole “Dynamic Network Reconfiguration Support for Mobile Computers”, Proceedings of the Third ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom ’97), Budapest, Hungary, September 26-30, 1997.
- [9] R. Drome, “Dynamic Host Configuration Protocol”, RFC1541, October, 1993.
- [10] T. Tokuda, T. Nakajima, and P. Rao, “Real-Time Mach: Towards a Predictable Real-Time System”, Proceedings of the 1st USENIX Mach Symposium, October, 1990.

- [11] J. Helander, "Unix under Mach: The Lites Server", Helsinki University of Technology, Master's Thesis, 1994.
- [12] Akihiro Hokimoto, Kuniaki Kurihara, Tatsuo Nakajima, "An Approach for Constructing Mobile Applications using Service Proxies", The 16th International Conference on Distributed Computing System(ICDCS'96), March, 1997.
- [13] Tatu Nakajima, Hiroyuki Aizu, Masaru Kobayashi, Kenji Shimamoto, "Environment Server: A System Support for Adaptive Distributed Application", The 2nd International Conference on Worldwide Computing and Its Applications'98 (WWCA'98), March, 1998.
- [14] Hari Balakrishnan, Srinivasan Seshan, Randy H.Katz, "Improving Reliable Transport and Handoff Performance in Cellular Wireless Networks", ACM Wireless Networks, December 1995.
- [15] David A.Maltz, David B.johnson, "The CMU Monarch Project:IETF Mobile IPv4 Implementation User's Guide", Carnegie Mellon University, June 1997.