

移動体通信プロトコル Mobile IPv6 の実装および評価

湧川 隆次[†] 植原 啓介[‡] 村井純[§]

慶應義塾大学 政策・メディア研究科[†] 慶應義塾大学 SFC 研究所[‡] 慶應義塾大学環境情報学部[§]

本論文では、移動体通信における移動透過性を保証する Mobile IPv6 プロトコルの設計実装を行い、それを用いたプロトコルの検証及び性能評価について述べる。Mobile IPv6 で IPsec を利用する時の問題点を指摘し、本実装での解決法を示した。評価においては、スループットやラウンドトリップタイム、また Mobile IPv6 の送受信時のコストを計測した。その結果、Internet Protocol version 6 (IPv6) プロトコルの性能を低下させること無く移動透過性を提供することができた。

Evaluation of the Design and Implementation of Mobile IPv6

Ryuji Wakikawa[†] Keisuke Uehara[‡] Jun Murai[§]

Graduate School of Media and Governance, Keio University[†]

Keio Research Institute at SFC[‡]

Faculty of Environmental Information, Keio University[§]

This paper presents preliminary results of our Mobile IPv6 prototype implementation. We evaluated throughput, round trip time and Mobile IPv6 processing overhead. From our results we showed that our implementation incurs minimal overhead and thus does not effect IPv6 protocol performance. This paper also details many changes and additions to the specification that were necessary for implementation, issues of IPsec integration in the protocol design, and new challenges for mobile IPv6.

1 はじめに

小型化、高性能化、様々な通信デバイスの登場、移動を支援する研究の充実により携帯型計算機は広く使われるようになった。移動を支援する技術の一つに Mobile IP [1] がある。Mobile IP は、移動により起こる IP アドレスの変更を隠蔽する移動透過性と通信相手ノードが移動ノードの IP アドレスを一意的に識別できるため常時発呼可能性を保証する。

また、Internet Protocol version 4 (IPv4) [2] アドレスの濁個に対応する IPv6 [3] の導入により、車、携帯電話といった計算機以外の多様なデバイスに IP アドレスを割り当てられる事が想定できる。車の通信環境下では複数の通信デバイスを駆使して通信を行うことが想定でき、また高速で移動するために IP アドレスが変化することが予想される。このとき、アドレスが変わることにより、通信が遮断される問題がある。また、携帯電話では移動に伴う IP アドレスの変化により、通話相手はその携帯電話への発呼をすることが出来なくなる。Mobile IPv6 を用いることにより、これらの問題は解決できる。

このように、Mobile IPv6 は今後のインターネットにおいて非常に大きな役割を果たす可能性がある。本稿では、Mobile IPv6 の FreeBSD [4] 上への実装及び評価について述べる。

本研究は、WIDE プロジェクトのインターネット自動車プロジェクトの一環として行われている。

2 Mobile IPv6

Mobile IPv6 の標準化は、現在 The Internet Engineering Task Force (IETF) の「IP Routing for Wireless/Mobile Hosts (Mobile IP)」Working Group (以後、Mobile IP グループ) で議論されている。Mobile IP グループでは、これまで Mobile IPv4 の仕様として RFC2002 を公開した。現在、IPv6 用の Mobile IP の仕様について RFC 化へ向けての議論が続いている。最新の Mobile IPv6 の仕様 [5] は、Internet-Drafts¹ として公開されている。

Mobile IPv6 は、Mobile IPv4 とは全く異なった処理が行われる。ここでは、Mobile IPv6 プロトコルについて紹介する。

2.1 Mobile IPv6 の用語

- Mobile node (MN)
リンクからリンクへ移動するノード
- Correspondent node (CN)
MN と通信する相手ノード。CN は、移動体ノードでも固定ノードでも可。
- Care of address (CoA)
MN が移動先リンクで取得する IP アドレス。CoA は、どのように取得しても構わない。また、複数の CoA を同時に MN が持っても構わないが、HA に対しては唯一の CoA を登録する。HA に対して

¹現在の最新版は、draft-ietf-Mobile IP-ipv6-12.txt である。本稿は、全てこの draft に基づいている

登録された CoA を Primary CoA と呼ぶ。

- Home Link (ホームリンク)
MN の持つホームアドレスと同じプレフィックスを持つネットワーク。
- Home agent (HA)
MN のホームリンク上のルータ。MN がホームリンクから離れている時、HA は MN 宛のパケットを代わりに受け取り、MN の CoA に向けて転送する。
- Home address (ホームアドレス)
ホームリンク上の移動体ノードに割り当てられた IP アドレス。
- Foreign Link (移動先リンク)
MN のホームリンク以外のリンク。
- Binding (binding)
MN のホームアドレスと MN の CoA の関係を示すもの。MN のホームアドレスと CoA の関係や有効期間といった情報である。binding は全てのノードで、必要に応じて MN の binding をキャッシュすることが出来なくてはならない。

2.2 IPv6 宛先オプション

Mobile IPv6 の処理を行うため、以下の 4 つの新たな IPv6 宛先オプションが定義された。

- Binding Update
MN が CN や HA に対して現在の binding を通知するために用いられる。本オプションが含まれるパケットは、全て IPsec を用いて認証を行わなくてはならない。Mobile IPv6 と IPsec の関係については後述する。
- Binding Acknowledgment
送られて来た Binding Update オプション内に、応答の要請が求められている場合のみ本オプションを用いて応答する。本オプションが含まれる全てのパケットは IPsec により守られている必要がある。
- Binding Request
本オプションは、MN に対して Binding Update オプションを送信するよう要求する時に用いられる。本オプションは、主に CN がキャッシュしている binding の有効期間が切れそうな時に MN に対して送信する。本オプションには、認証は必要無い。
- Home Address
本オプションは、MN がパケットを送信する時に自分のホームアドレスを知らせるために用いる。MN が移動先リンクに接続されている時、MN は送信元アドレスとして CoA を用いる。しかし、本オプションを含む事により、受け取ったノードは、送信元アドレスをホームアドレスに置き換えられ、結果として MN との通信はホームアドレスで行われているように見える事が可能となる。

Mobile IPv6 ではサブオプションもいくつか定義されているが、本稿では触れないこととする。

2.3 基本処理

MN は、ホームリンクや移動先リンクに関わらず、常にホームアドレスでアドレス付けされる。MN がホームリンクに接続されている時は、MN 宛のパケットはインターネットの通常の経路制御に基づきホームアドレスに対して届けられる。これは、MN のホームアドレスのサブネットプレフィックスがホームリンクのそれと同じであるためである。

MN が移動先リンクに接続されている場合は、MN は 1 つかそれ以上の CoA を取得しなくてはならない。CoA は、移動先リンク上で有効な IP アドレスである。CoA は、stateless な Address Autoconfiguration [6] あるいは stateful な Dynamic Host Configuration Protocol IPv6 (DHCPv6) [7] といった方法で取得する。また、手動による取得も可能である。

CoA 取得後、MN は HA に対して CoA の一つを primary CoA として登録する。この登録は、IPv6 Binding Update 宛先オプションを含んだパケットを HA に対して送信する事により行われる。HA は受信し処理が終了すると、IPv6 Binding Acknowledgment 宛先オプションを含んだパケットで応答する。図 1 に示す。本処理が正常に終了すると、これ以降、HA は MN のホームアドレス宛のパケットを傍受しそれを MN の primary CoA 宛にトンネルを用いて転送する。この処理を、図 2 にまとめた。

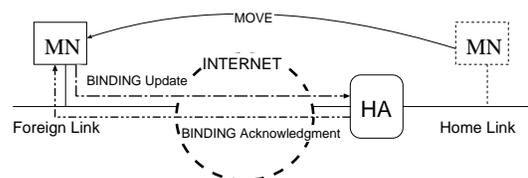


図 1: MN の移動後の処理

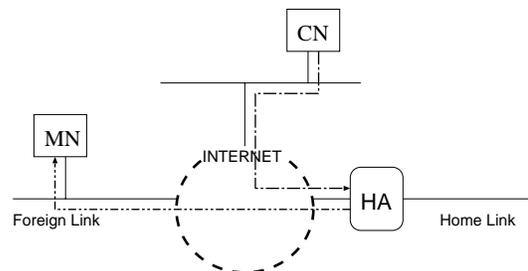


図 2: Binding 登録処理終了後の HA の役割

CN が MN に対してパケットを送信する場合は、CN は自分が持つ binding キャッシュを調べる。binding キャッシュ内に MN に対する binding が無い場合は何

も処理を加えずにネットワークへパケットを送信する。この場合は、パケットは通常通り経路制御されホームリンクに届く。その後、HAがそれを受け取りMNに対して転送する事により最終的にMNが受け取ることが出来る。MNはHAから転送されたパケットを受け取ると、そのパケットの送信ノードであるCNが自分のbindingをキャッシュしていないと分かり、MNはCNに宛ててBinding Updateを送る。これにより、CNはMNのbindingをキャッシュする事が出来る。本処理を図3にまとめた。キャッシュした後は、次に述べるキャッシュが有効時の場合と同じ処理が行われる。

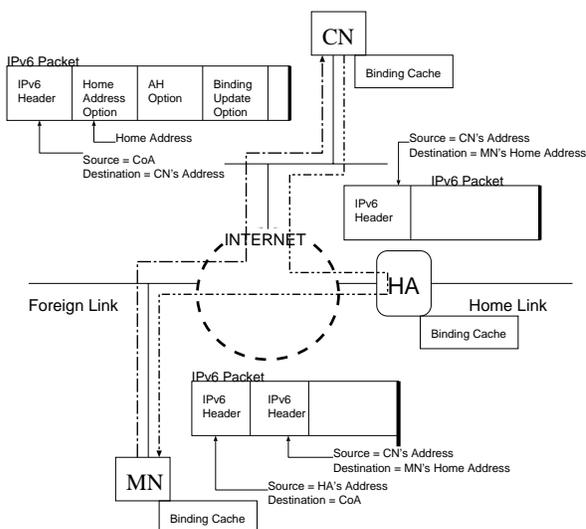


図 3: binding キャッシュが無い場合の通信

CNがMNのbindingをキャッシュしていた場合は、CNは送信するパケットにIPv6 経路制御オプションヘッダを含んで送る。IPv6 ヘッダの宛先アドレスは、binding キャッシュから得られるMNのCoAが入り、IPv6 経路制御オプションの次ホップアドレスはホームアドレスとなっている。これにより、MNは受け取った後、IPv6 経路制御オプションを処理する段階で次ホップアドレスが自分のホームアドレスである事が分かりホームアドレスで受信する事が出来る。また、MNから送信されるパケットにはIPv6 Home Address宛先オプションが入り、ここにMNのホームアドレスが入り、IPv6ヘッダの送信元アドレスには、MNのCoAが含まれている。CNは、パケットを受信後IPv6 Home Address宛先オプションを処理し、IPv6ヘッダの送信元アドレスのCoAとホームアドレスを入れ換える。これにより、IPより上位層に対してMNのホームアドレスと通信しているように分かる。本処理を図4にまとめた。

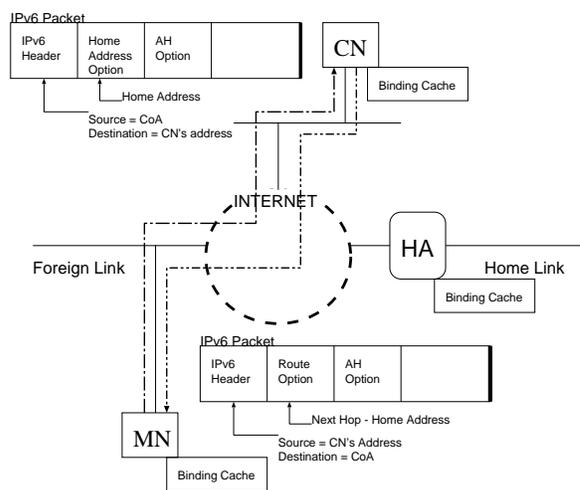


図 4: binding キャッシュがある場合の通信

2.4 Mobile IPv6 と IPsec の関係

仕様書内では、Mobile IPv6のbindingに関するIPv6 Home Address宛先オプション、IPv6 Acknowledgment宛先オプションは、IPsec Authentication Header (AH)[8]のトランスポートモードを用いて認証するように定義されている。しかし、これにはいくつかの問題がある。

最初に、IPsecについて簡単に説明する。IPsecでは、認証、暗号化のアルゴリズム及び鍵管理の仕組みをプロトコル外で行う。そのため、通信を開始する前に、これらの情報を通信相手と共有する必要がある。これが、Security Association(SA)である。このSAは手動で確立する事も出来るが、鍵管理プロトコルを用いる事も出来る。プロトコルとしては、Internet Key Exchange(IKE)[9]がある。IKEではSAを確立するために、IKE自身の通信のためのSAを確立し、そのSAを用いてIPsecの実際の通信に用いる情報の交換を行い、SAを確立する。IPsecでは、認証のためのAuthentication Header(AH)と暗号化のためのEncapsulating Security Payload(ESP)[10]の2つのIPオプションヘッダが定義されている。これらのヘッダ内にはSAで共有した情報を用いて計算したチェックサムや暗号化されたペイロード等の情報が含まれている。つまり、SAを持たないノードはパケットを受信時に、IPsecの再計算を行うことができない。Mobile IPv6では、IPsecのAHヘッダを用いて認証を行うように定義されている。つまり、Binding Updateの交換等にIPsecでの認証を用いる事により、成りすましや不正なパケットによる危険を回避する。

Mobile IPv6 と IPsec に関わる 1 つ目の問題は、IKE を用いた鍵交換である。Security Association(SA)が無い場合 IPsec は IKE を用いて SA を確立する。しかし、この時点で Mobile IPv6 の状態としては、bind-

ingが有効でない状態である。つまり、IKEでSAを確立しようとした時点では、相手ノードに対してMobile IPv6を用いて通信できないという矛盾が生じてしまう。Mobile IPv6の仕様では、DNSへの問い合わせの等の継続しない一時的な通信は直接CoAで通信することを推奨している。そこで、本問題を回避するためIKEによるSA確立時もCoAを用いて、ホームアドレスとCNのアドレスとの間でSAを確立することにした。

2つ目の問題は、現在の仕様ではIPsecの計算をする時点でIPv6の送信元アドレスにホームアドレスを用いるかCoAを用いるか明確に定義されていないことである。これが定義されていないと、計算結果の値が全く異なってしまう。本実装では、IPsecの処理はホームアドレスを用いて行っている。実際の通信はCoAで行っているが、IP層より上位層からはホームアドレスを用いて通信しているように見える。そのため、IPのパケットとしては、ホームアドレスを用いて全てを処理する事にした。

3つ目の問題は、IPv6 Home Address宛先オプションの処理についてである。本オプションを含むパケットを受け取ったノードは、AHを処理する前に送信元アドレスを本オプション内にあるMNのホームアドレスに置き換える必要がある。本処理が、IPsecが提供する認証や暗号化に影響を与える可能性が考えられる。本問題には、IPsecに関する検証が必要である。

4つ目の問題としては、Binding Update宛先オプションを含むパケットでIPsecの認証を行う際にSAをSecurity Association DataBase (SADB)から取得する必要がある。この時、ホームアドレスを用いてデータベースからSAを検索するかCoAを用いて検索するかは述べられていない。本実装では、2つ目の問題と同じ理由でホームアドレスを用いて検索を行っている。

最後の問題は、Home Address宛先オプション及びBinding Update宛先オプション等をIPv6オプションヘッダ内に配置する順序である。Mobile IPv6の仕様書内では厳密には言及されていない。現在、AHの前にHome Address宛先オプションを置きBinding Update宛先オプションをAHの後ろに置く方法と、AHの前にHome Address宛先オプションとBinding Update宛先オプションをまとめて置く方法が検討されている。前者は、IPv6オプションヘッダを前から処理するというIPv6仕様で推奨されている方法である。受け取ったノードはオプションヘッダの先頭から処理できる。しかし、後者の場合Home Addressオプションを処理しIPsecのAHの処理を行った後、前に戻りBinding Updateオプションを処理しなくてはならない。一方、二つのオプションを一つの宛先オプションとして送信するためデータサイズを小さくする事が可能である。本実装では、IPv6の仕様

に沿い、前者の仕様を採用している。しかし、さまざまな実装との互換性を保つためには、ヘッダ内の何処にオプションが配置されても処理できるよう実装する事が必要となる。

このように、現時点の仕様ではIPsecとの関係が曖昧なため、実装する時点での問題が多い。現時点でIETFやIETFのmailing list等を使って議論が続いている。

3 Mobile IPv6の設計

本章では、Mobile IPv6を実装する上での設計について考察する。

3.1 Mobile IPv6の設計

Mobile IPv6の実装方法としては、Bump In The Stack(BITS)方式とNative Implementationの2通りある。BITS方式では、IPv6スタックの外にMobile IPv6スタックを構築し独自に処理を行う方法である。BITS方式では、Mobile IPv6の処理をユーザレベルで実装しソフトウェアとして実現することも可能である。しかし、Mobile IPv6ではIPv6のオプションヘッダを使うなどBITS方式での実現は非常に困難である。またBITS方式ではオーバーヘッドが大きくなるため望ましくない。Native Implementationは、IPv6スタック内にMobile IPv6スタックを実装する方法である。Mobile IPv6はその仕様上、IPv6と非常に密接な関係にある。例えば、IPv6スタックの経路決定等の送信処理、及びIPv6オプションヘッダの処理など受信処理を変更する必要がある。またIPsecや移動検知等を行わなくてはならないため、本設計では後者方法を選択した。Mobile IPv6スタックは、IPv6層及びインタフェース層の中に構築した。

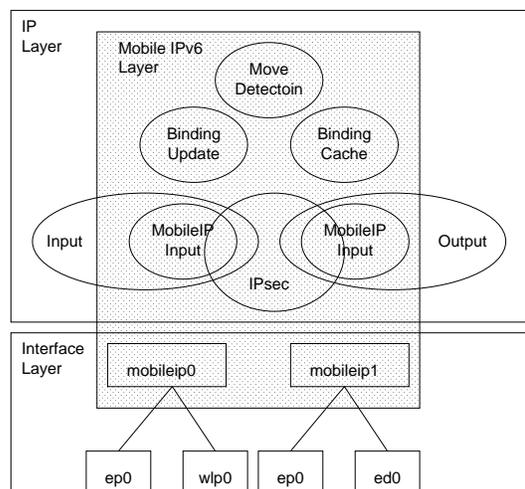


図 5: Mobile IPv6 スタックの設計

図5に、Mobile IPv6 プロトコルスタックの設計を示す。図5は基本的にMobile IPv6のMNの設計図

であるが、ノードの種類によって機能が削除されたり一部追加される以外、基本的には同じである。

Interface 層にある mobileip0 と mobileip1 は、Mobile IPv6 用に新たに導入した仮想インターフェースである。本インターフェースには、MN の場合、自分のホームアドレスが付けられる。IP 層の上位層は全て、本仮想インターフェースを介して通信を行う。1 台のノードで複数のホームアドレスを持つ可能性もあるため本仮想デバイスは複数構築できる。仮想デバイスは、実際に存在するイーサネットインターフェース ep や ed といったネットワークインターフェース²と関連付けされる。ホームアドレスに対する CoA は、ホームアドレスの割り当てられた仮想デバイスに関連付けされたネットワークインターフェースのグローバル IPv6 アドレスである。図 5 にある通り、複数のネットワークインターフェースを関連付けすることが可能である。これは、複数の CoA の利用を Mobile IPv6 の仕様の中で許可しており、将来的には複数インターフェースを有効利用することが考えられる。

Mobile IPv6 層には、binding cache(管理)機構や binding update(管理)機構等があり、それぞれ定期的に binding update を再送するための残り生存時間や binding の検索といったサービスを提供する。また、Move Detection 機能では、定期的に受け取るはずの経路広告を受け取り現在の CoA やホームアドレスのプレフィックスと比較を行い、現在のリンクがホームなのか移動先なのかを判断する。このとき、移動が検知されると、それに応じた処理を行う。

3.2 データ構造

本節では、Mobile IPv6 の仕様上必要なデータ構造及び実装上定義したデータ構造の設計について述べる。

図 6 に MN のデータ構造を示す。図中の文字は、データ構造名である。MN が仕様上必要なデータ構造は、binding_cache 構造体、binding_update_list 構造体、home_agents_list 構造体である。binding_cache 構造体は、binding をキャッシュするための構造体であり、MN はパケットを送信する時に毎回 binding を検索する。そのため、検索の効率化が性能に影響するために、ハッシュデータ構造で構築する。binding_update_list 構造体は、MN が binding update を行ったホストの記録であり、binding の生存時間が無効になる前に binding update を再送するなどのために持つ。binding_update_list 構造体は、リストでそれぞれ管理する。home_agents_list 構造体は、MN の HA の情報を格納する。MN のホームリンク上には複数の HA がサービスを行っている場合は、複数のエントリが本データ構造内に構築される。

binding_queue 構造体は、IPv6 宛先オプションを使った binding 関連の Mobile IPv6 のデータを、実際

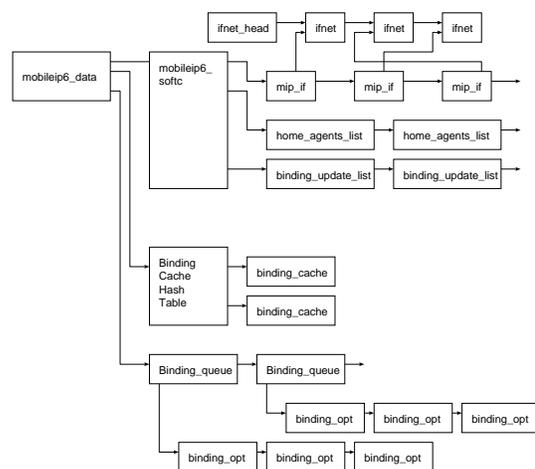


図 6: Mobile IPv6 MN のデータ構造

の通信データに piggyback するために用いる。宛先が同じで且つ本構造体を送るべきオプションがある場合は piggyback されて送られる。

mobileip6_data 構造体は、binding_cache データ構造へのポインタや、Mobile IPv6 の統計情報等のデータを持つ。本構造体は、MN、CN、HA 全てで持つ。mobileip6_softc 構造体は、ホームアドレス毎にエントリが作られる。本構造体には、CoA の情報や、binding_update_list 構造体へのポインタ、home_agents_list 構造体へのポインタなどがある。

CN の場合、図 6 中の、mobileip6_data、binding_cache、binding_queue 構造体を持つ。HA の場合、mobileip6_data、binding_cache、binding_queue、home_agents_list 構造体を持つ。home_agents_list は、HA 同士で通信を行う時に必要である。

mip_if は複数のインターフェースをサポートするためのデータ構造である。本データ構造の詳細は実装のところで詳しく述べる。

4 Mobile IPv6 の実装

今回、Mobile IPv6 を FreeBSD-3.4RELEASE + kame-20000214-freebsd34-stable[11] を用いて実装した。Mobile IPv6 は、draft を基に実装した。現在まで、MN と CN の実装をほぼ終え、End to End 通信を実現した。

4.1 データ構造

本節では、Mobile IPv6 で重要なデータ構造について説明する。まず、Mobile IPv6 のデータ構造の核である mobileip6_softc 構造体を説明し、その後 binding_cache 構造体について解説する。

図 7 は、Mobile IPv6 用の仮想インターフェースの状態を保持するデータ構造である。本構造体には、仮想インターフェースに関連付けされたインターフェースへのポインタや、ホームアドレスの ifnet 構造体 (mip_if)、HA の IPv6 アドレスといった情報が格納

²無線インターフェースなど、通信インターフェースなら種類は問わない

される。また、現在の本ホームアドレスに対するリンクの位置の情報や、送信された binding update list の状態も保持されている。

```
struct mobileip6_softc {
    int     whereami;
    struct ifnet  mip_ifp;
    struct mip_if *mif;
    struct binding_update_list *bu_head;
    struct home_agents_list  *ha_head;
};

struct mip_if {
    struct mip_if *next;
    struct sockaddr_in6 addr;
    char *if_name;
    struct ifnet *ifp;
    u_int8_t status;
#define PCOA_REGISTER    0x01
#define PCOA_UNREGISTER 0x02
#define PCOA_NONE       0x03
#define PCOA_UNKNOWN    0x04
    u_int8_t  primary_flag;
    u_int8_t  unit;
    u_int8_t  reserved2;
};
```

図 7: Mobile IP6_softc 構造体

図 8は、binding キャッシュの情報である。IPv6 の送信時には、全てのノードが、必ず宛先アドレスを元に binding キャッシュ内に binding の情報があるかを検索するためハッシュデータ構造を用いて実現されている。本実装ではハッシュ表サイズは64バイトである。現時点では、広域なテストはまだ行っていないが、テストの結果によってはハッシュ表サイズやハッシュ関数の改良が必要である。

4.2 送信処理

本節では、Mobile IPv6 の送信処理について MN を中心に説明する。受信処理は、それぞれの IPv6 オプションをパケットの先頭から順に処理するという単純な実装であるため解説は割愛する。

図9は、kame の実装である ip6_output の処理内で行われる Mobile IPv6 の処理を示したものである。色が付いている関数は既存の kame のものである。

パケットが、どのホームアドレスを選択するかは、通信の開始時の Protocol Control Block (pcb) を割り

```
struct binding_cache {
    struct binding_cache *next;

    struct in6_addr  home_addr;
    struct in6_addr  coa;

    u_int32_t        lifetime;
    u_int8_t         cache_flags;
    u_int8_t         cache_status;
    u_int8_t         prefix_len;
    u_int16_t        seqno;
    time_t           lasttime;
};
```

図 8: binding cache 構造体

当てる際に選択ポリシーに従い決定する。

本ノードが MN として初期化されていれば、現在 MN がホームリンクに接続されているか移動先リンクに接続されているかを調べ、ホームリンクに接続されている時は Mobile IPv6 の処理を省く。insert_home_address 関数で、MN が送信する時に必要な IPv6 Home Address 宛先オプションを作り、パケットが入れられている mbuf[12] の中に挿入する。このとき、送信元アドレスにホームアドレスが入り、Home Address 宛先オプション内には、select_CoA で取得した CoA が入っている。これは、前述したように IPsec 処理においては送信元アドレスにホームアドレスを用いて計算するためである。CoA アドレスの選択は、select_CoA_address 関数で行われる。現在のところ CoA の選択は、Primary CoA のみを返すように実装されている。しかし、今後ポリシーによった CoA の選択などを行えるように実装を進める予定である。

送信する前に、全てのノードは binding キャッシュを調べる。lookup_binding_cache は引数として宛先アドレスを受け、それを元に binding があるか調べ、binding を返す。binding があれば、insert_routing_header 関数で IPv6 経路制御オプションを作り、パケットが格納されている mbuf の中に挿入する。また、lookup_binding_queue 関数を用いて、piggyback して送るべき binding 関連の IPv6 宛先オプションを探す。送るべき IPv6 オプションがあれば、insert_destination_header 関数を用いて mbuf の中に挿入する。但し、IPv6 Binding Request 宛先オプションは IPsec が不要無いので、次の check_IPsec は省かれる。Mobile IPv6 以外の IPv6 宛先オプションが先に指定されていれば、そちらのオプションが優先される。

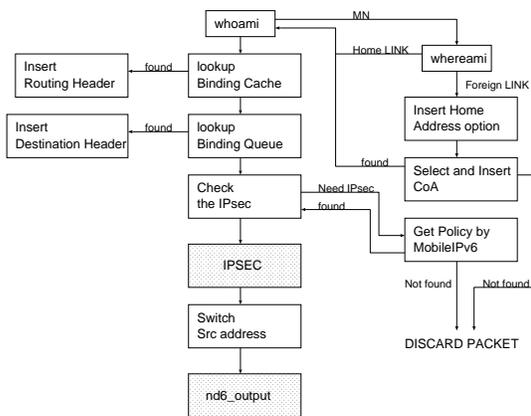


図 9: IPv6 の送信処理の概要

IPsec は、認証用のポリシデータベース内の Mobile IPv6 用のポリシを探す。もしなければ、IPsec の処理が行われないのでパケットは破棄される。それ以外の場合は、得られたポリシを用いて通常通りの IPsec の処理へと進む。現在の実装では、事前に SA を確立する必要がある。しかし、設計の段階では IKE など考慮に入れた設計となっているため将来的には IKE を利用する事が出来る。

最後に IPv6 ヘッダの送信元アドレスをホームアドレスから CoA アドレスに交換する。同じく、Home Address 宛先オプション内のアドレスをホームアドレスに交換する。

5 Mobile IPv6 の評価

本章は、実装した Mobile IPv6 を用いて性能評価を行った結果について述べる。評価項目は、処理コストである。MN と CN 間でのラウンドトリップタイムを計測し、Mobile IPv6 を導入することにより起きる遅延についての考察を行う。また、FTP を用いてスループットの性能についても考察する。最後に、MN の出力の際にかかる Mobile IPv6 のコスト及び Mobile IPv6 で導入された IPv6 宛先オプションのコストを pentium カウンタを用いて測定し評価する。

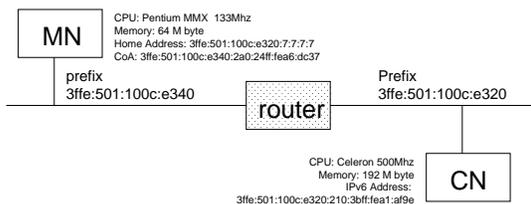


図 10: 実験ネットワーク構成図

図 10 に実験測定環境を示す。ルータを挟んで 2 つの異なるプレフィックスを持つネットワークを用意した。CN がいるネットワークが MN のホームリンク

であり、MN は移動先ネットワークで CoA を取得している。

今回の実験では、MN と CN 間での計測のみ行った。MN は CN に対して Binding Update をまず送信して CN に対して Binding キャッシュを作る。本方法は、仕様には無いのだが測定するために用いた。しかし、本方法が Mobile IPv6 の性能評価で得られるデータ面では何ら影響を及ぼさない。

5.1 ICMP を用いたラウンドトリップタイム

ラウンドトリップタイムは、ICMP を用いて計測した。ICMP のエコーメッセージを CN から MN に対して 50 回送信した。ICMP のデータ長は 64 バイトで、送信間隔は 1 秒である。結果は表 1 に示す。

実験結果より、ラウンドトリップタイムは Mobile IPv6 の影響として約 0.3ms 程のコストが分かる。しかし、Mobile IPv6 を用いた場合は、MN が常に IPv6 Home Address 宛先オプションをパケットへ挿入し、CN がそれを処理する時間と、CN が IPv6 経路制御オプションを挿入し MN がそれを処理する時間が含まれている。また、それ以外にも CN における binding キャッシュの検索等の処理も行われている。これらを考慮すると、0.3ms の誤差はほぼ無視できる範囲であることが分かる。

表 1: ICMP を用いたラウンドトリップタイム

Mobile IPv6	最小時間	平均時間	最高時間
無し	1.069 ms	1.521 ms	2.871 ms
有り	1.334 ms	1.818 ms	2.717 ms

5.2 TCP を用いたスループット

TCP を用いた実験では、25 メガバイトのファイルを FTP を用いて送信してスループットを測った。計測は 5 回行い平均を取った。結果は表 2 に示す。TCP を用いた場合、Mobile IPv6 の有り無しでは、スループットの違いは、殆んど見られなかった。これにより、Mobile IPv6 の処理が TCP に与える影響は無視できる事が分かった。

表 2: TCP を用いたスループット

Mobile IPv6	スループット
無し	213.37 KB/s
有り	209.32 KB/s

5.3 Pentium カウンタを用いたコスト評価

Pentium カウンタを用いて、IP の output 関数及び input 関数にかかるコストを計測した。output は、MN で測定し、input は CN で測定した。これは、MN

の output で IPv6 Home Address 宛先オプションの挿入等が行われ、CN の input でそのオプションの処理を行っているためである。CN と MN では CPU のクロック数が異なるため input と output を一概に比べることは出来ないが、本評価では Mobile IPv6 の有無によるこれら関数の影響を調べるのが目的であり、input 関数と output 関数を比べるものではない。

測定は、ICMP のエコーメッセージを 1 秒毎に送信して行った。測定結果を表 3 に示す。それぞれ、Mobile IPv6 の処理のコストが無いときに比べてかかっている。しかし、どちらもほぼ無視できる時間無に行われている。これにより、IP の送信、受信どちらの処理にも影響無く実装されていることが分かる。

表 3: Pentium カウンタを用いたコスト

Mobile IPv6	input	output
無し	0.0258 μ s	0.229 μ s
有り	0.0294 μ s	0.328 μ s

6 関連研究

Mobile IPv6 の実装としては、FreeBSD 上に実装した Ericsson Telebit がある。現在の kame プロジェクトのソースコード内には Ericsson の実装が含まれている。しかし、IPsec の認証部分の欠如などの問題により、互換性テストは行えていない。また、CMU Monarch Project では 4.4BSD UNIX システム上に実装している。しかし、本コードは INRIA の IPv6 コードを元に作られており、リリース自体が 1997 年にされた以降更新が無いため古い仕様のみである。NEC が、同じく FreeBSD 上に実装した物や AIX 上に実装した Bull の実装などが有る。

7 今後の課題

今回の実装及び評価は、Mobile IPv6 スタックの一部を実装し行ったものである。今後の課題は、Mobile IPv6 スタックを全て実装終了する。また、今回の実験では他の実装との互換性に関する評価を行っていない。理由の一つに、無償で取得できる Mobile IPv6 スタックが少なく、また完全に実装されているスタックが無償では手に入られなかったからである。今後は、それらの互換性に関する評価も行う。

本稿では、MN と CN それぞれ 1 台ずつで実験を行ったが、binding キャッシュの探索などはキャッシュ内のエントリが増えれば増える程、性能が劣化する。そのため、今後は台数を増やして実験を行う必要がある。

本研究は、WIDE プロジェクトのインターネット自動車プロジェクトの一貫として行われている。インターネット自動車の様な、移動体通信では複数の通信インターフェースを用いて通信 [13] を行う必要が

ある。例えば、無線デバイスが使えるところは無線デバイスを用い、利用不可能なところは携帯電話を利用する必要がある。そのため、Mobile IPv6 に置いて、複数のインターフェースの支援を行う為の拡張を行う。

謝辞

本研究を進める上で様々な議論や助言を頂いた WIDE プロジェクト、インターネット自動車プロジェクト及び Rover プロジェクトの皆様にご感謝の意を表します。また、本研究において、Mobile IP を始め御指導を頂いた株式会社東芝の石山政浩氏に心から感謝します。

参考文献

- [1] C.Perkins.: "IP Mobility Support," *Request for Comments: 2002 Oct. 1996*
- [2] J.Postel.: "Internet Protocol," *Request for Comments: 791, Sep. 1981*
- [3] S.Deering, R.Hinden.: "Internet Protocol version 6 (IPv6) specification", *Request for Comments: 2460 Dec. 1998*
- [4] Mckusick, M. k., Bostic, K., Karels, M.j.: "The Design and Implementation of the 4.4BSD Operating System.," Addison-Wesley, 1996.
- [5] C.Perkins.: "Mobility Support in IPv6," *Internet-Drafts(draft-ietf-mobileip-ipv6-12.txt): 27 Apr. 2000*
- [6] S.Thomson, T.Narten.: "IPv6 stateless address autoconfiguration," *Request for Comments: 2462 Dec. 1998*
- [7] Jim Bound, C. Perkins.: "Dynamic Host Configuratin Protocol for IPv6(DHCPv6)," Work in progress: Feb 1999.
- [8] S.Kent, R.Atkinson.: "IP Authentication header", *Request for Comments: 2402 Nov. 1998*
- [9] D.Harkins, D.Carrel.: "The Internet Key Exchange (IKE)", *Request for Comments: 2409 Nov. 1998*
- [10] S. Kent, R. Atkinson.: "IP Encapsulatin Security Payload (ESP)", *Request for Comments: 2406 Nov. 1998*
- [11] KAME Projects.: "http://www.kame.net", 31 July. 2000.
- [12] W. R. Stevens, W. Richard.: "TCP/IP Illustrated, Volume 2", Addison Wesley; 10/94; ISBN: 020163354X
- [13] 砂原秀樹, 比良木貴志, 植原啓介, 尾家祐二: "移動体端末装置における通信インターフェースの自動選択機能の実現," DiCoMo, 1997.