第 XIV 部

移動体通信環境

第 14 部
移動体通信環境

第 1 章　はじめに

PDA やノート PC 等の携帯端末の性能の向上と、IEEE 802.11 に代表される無線ネットワークデバイスの普及に伴い、移動先から携帯端末を利用したインターネットへのアクセス、いわゆるモバイルコンピューティングが活発に行われるようになってきた。また、インターネットへの接続点および接続方法のさらなる増加が見込まれるため、移動しながらのネットワークへのアクセスといった、ネットワークの利用方法の多様化が考えられる。

Rover は、このようなネットワークを利用した移動計算機環境について研究するグループである。我々が研究の対象とする移動計算機は主にノート型 PC 等の十分な処理能力を持つ計算機であるが、PDA 等の比較的能力の低い計算機も視野に入れた議論を行っている。より具体的には、インターネット移動体通信機構（LIN6、Mobile-IP）の開発、より高速なネットワーク間ハンドオフの手法、地理情報システムに関する議論、移動に伴う動的適応の枠組みの開発などの活動が行われている。

本年度は、これまで Rover で議論してきた研究項目のうち、WIDE メンバーが積極的に開発した、LIN6 プロトコルの性能評価について報告する。LIN6 は、IPv6 における新しい移動透過保証プロトコルである。本報告書では LIN6 の概要を示し、同時に現在 IETF で議論されている移動透過保証プロトコルである Mobile IPv6 の概要を述べる。その上で Mobile IPv6 と LIN6 の定性的および定量的な比較検討の結果を示す。

第 2 章　移動体通信プロトコル LIN6 の性能評価

将来のインターネット像を考えると、テレビ電話やテレビ会議、中継などのアプリケーションがインターネットサービスとして提供されることが想定される。このような利用形態を考慮すると、あらゆる機器がどこでもネットワーク資源に接続できるようなインフラが整備され、いつでもどこでもインターネットへと接続可能であると考えられる。このようなインターネット環境では、ノードが移動することが前提となり、位置に依存しないノードの識別や、移動しながら通信を継続することが可能な環境を実現しなければならない。

現在のインターネットでは、ノードがネットワーク間を移動する場合、接続するサブネットに応じて IP アドレスが割り当てられるため、位置に依存しないノードの識別や、移動しながら通信を継続することができない。この問題を解決するプロトコルとして、移動透過保証プロトコルが提案され、IETF で標準化が行われるとともに、様々な議論が行われている。現在、インターネット標準になりつつある移動透過保証プロトコルとして Mobile IPv6 が提案されているが、ヘッダオーバヘッドや一点障害による耐障害性の問題など、様々な問題が指摘されている。そこで我々は、Mobile IPv6 での問題点を解決するためネットワークアーキテクチャの再考を行い、LINA (Location Independent Network Architecture) と呼ばれるネットワークアーキテクチャを提案した [72]。そして、LINA を IPv6 に適応した移動透過保証プロトコルとして、LIN6 (Location Independent Networking for IPv6) と呼ばれるプロトコルを提案し、現在までにプロトコル設計、プロトタイプ実装を行い、基本性能の評価まで行った [162]。

本報告書では性能評価のまとめとして、我々が提案している LIN6 と Mobile IPv6 をプロトコル仕様を基に定性的な比較を行う。さらに、LIN6 と Mobile

IPv6 の定量的な比較評価を行うために、それぞれの実装を用いて、通信全体でかかる処理オーバヘッドを測定する。

2.1 Mobile IPv6 の概要

Mobile IPv6 は、IETF (Internet Engineering Task Force) で標準化が進んでいる、インターネット上で移動透過性を保証するためのプロトコルである [114]。

2.1.1 Mobile IPv6 の基本概念

Mobile IPv6 では、**ホームリンク**と呼ばれる移動ノードが通常接続しているネットワークを定義し、移動ノード（Mobile Node: MN）はノード自体を識別する情報として、**ホームアドレス**と呼ばれるアドレスを保持する。ホームアドレスは移動によって変化しない識別子であるため、通信ノード（Correspondent Node: CN）はホームアドレスを利用して、移動ノードに対してパケットを送信することが可能である。移動ノードが移動した場合、接続点を示すアドレスを取得する。このアドレスは**気付けアドレス（Care of Address: CoA）**と呼ばれ、パケットを移動ノードまで配送するための位置指示子として用いられる。Mobile IPv6 では、ホームアドレスと CoA の対応付けを **binding** と呼び、移動ノードはサブネットを移動するたびに binding を更新することにより通信が継続される。binding は **Home Agent (HA)** と呼ばれるルータで管理され、HA ではつねにホームアドレスに対する最新の CoA を保持している。

2.1.2 Mobile IPv6 の通信手順

Mobile IPv6 の通信手順を**図 2.1** に、Mobile IPv6 のヘッダ構成を**図 2.2** にそれぞれ示す。図 2.2 の (a)、(b)、(c)、(d) はそれぞれ図 2.1 に対応している。なお、IPv6 基本ヘッダ中の src は送信元アドレスを示し、dst は宛先アドレスを示すものとする。

移動ノードは移動の際に、CoA を HA に登録し binding を更新する。HA は binding を基に、移動ノードのホームアドレス宛のパケットを CoA へ転送する機能を持つ。この機能により、通信ノードから送信された移動ノードのホームアドレス宛てのパケットは、まず HA に到達し（図 2.1(a)）、HA から CoA へトンネリングの機構を用いて転送され（図 2.1(b)）、最終的に移動ノードへと到達する。このときの経路は、通信ノードから HA、HA から移動ノードという三角経路で配送される。移動ノードから通信ノードへの返信は、移動ノードの HA を介することなく、通信ノードへ直接配送される（図 2.1(c)）。移動ノードから通信ノードへの返信のヘッダ構成は、後で述べる binding に関するオプションを含む場合（図 2.2(c)）と含まない場合（図 2.2(c')）が存在する。

Mobile IPv6 では、三角経路でのパケット配送ではなく、最適経路での配送を行うために binding に関する 3 つのオプションを新たに導入している。通信ノードと移動ノード間ではこれらのオプションを用い、**Binding Cache** と呼ばれる binding に関するキャッシュの取得や更新を行う。通信ノードは Binding Cache が存在する場合には、Binding Cache に保持されている CoA を宛先アドレスとし、ホームアドレスを経路制御ヘッダへ追加し、パケットを送信する（図 2.1(d)）。この方法により、パケットは直接移動ノードの CoA へ送信され、経路制御ヘッダにより移動ノード宛のパケットであると認識できる。

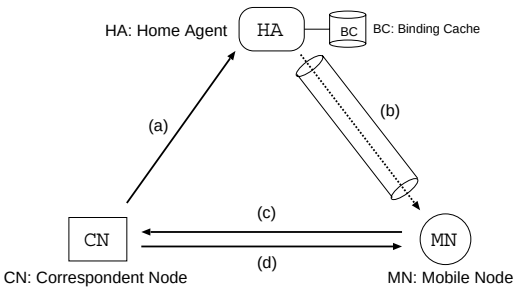

図 2.1.　Mobile IPv6 の通信

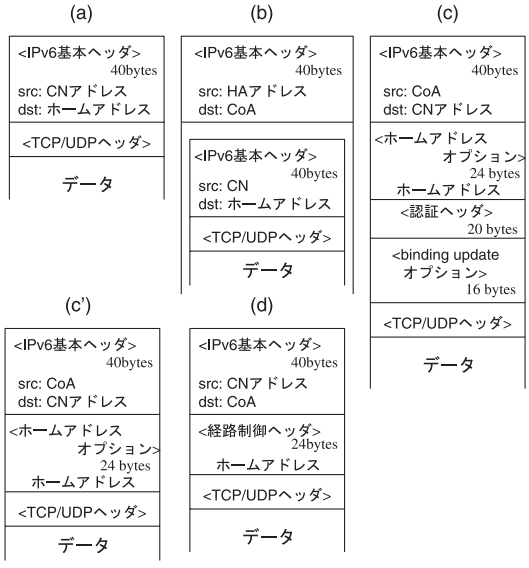

図 2.2.　Mobile IPv6 の通信時のヘッダ構成

しかし、通信初期の数パケットは Binding Cache を保持しない配送となり、HA を経由した冗長経路でのパケット配送となる。また、Binding Cache は、通常 IPsec[86] の認証ヘッダ [85] を用いて認証が行われない限り生成されない。したがって、通信ノードが移動ノードを認証できなかった場合も、初期の数パケットと同様に冗長経路でのパケット配送となる。

一方、HA に着目すると、HA は通信の核となっており、HA へパケットが到達できない場合は移動ノードと通信を行うことができない。しかし、HA はホームリンク上にのみ配置可能であるため、実質的に HA の複製を配置することは不可能であり、2.3.3 項で述べるように耐障害性が低いと考えられる。

2.2 LIN6 の概要

LIN6 は、現在の移動透過性が保証できない原因はネットワークアーキテクチャそのものにあると考え、ネットワークアーキテクチャから再考して構築した移動透過保証プロトコルである [72]。

2.2.1 LIN6 の基本概念

LIN6 では、IP アドレスが位置に関する情報とノード自体を識別する情報を分離することなく保持しているという**ネットワークアドレスの二重性**と呼ばれる問題を解決するため、位置指示子とノード識別子という 2 つの情報を概念的に分離する。位置指示子とノード識別子の概念を分離することにより、ネットワーク層より上位層では、ノード識別子を用いた位置に依存しないコネクションを確立し、ネットワーク層では、位置指示子を用いた経路制御を行うことができ、移動透過性が保証できる。

LIN6 のノード識別子は **LIN6 ID** と呼ばれ、EUI-64[68] 形式を利用したグローバルユニークな識別子として定義される。また LIN6 では、ネットワークの位置を示すものとして、IPv6 のネットワークプレフィクスを利用する。ここで、LIN6 で利用されるアドレスを**図 2.3** に示す。

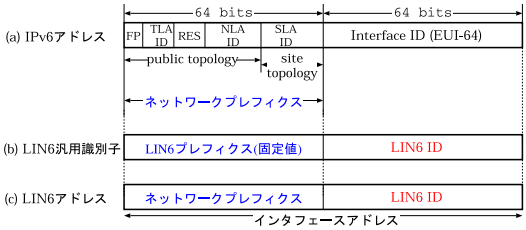

図 2.3. IPv6 アドレス、LIN6 アドレス、LIN6 汎用識別子

図 2.3(a) は、IPv6 のユニキャストアドレスの構造である。IPv6 のユニキャストアドレスは、上位 64bit が経路制御のために用いられるネットワークプレフィクス、下位 64bit がインタフェース識別子という構造である。インタフェース識別子は、インタフェースが接続するサブネットの中でユニークな識別子である。

図 2.3(b) は、**LIN6 汎用識別子**と呼ばれ、ノード識別子を 128bit に拡張した識別子である。LIN6 汎用識別子は **LIN6 プレフィクス**と呼ばれる固定値へ LIN6 ID を埋め込んでいるため、位置に依存しないアドレスとなる。このため、LIN6 汎用識別子はネットワーク層より上位の層における識別子として利用でき、このアドレスを用いてコネクションを確立することにより、移動の際にもコネクションが継続できるようになる。

図 2.3(c) は、**LIN6 アドレス**と呼ばれるネットワーク上の位置を示すネットワークプレフィクスと LIN6 ID の情報を含んだアドレスである。LIN6 アドレスは、ネットワークプレフィクスを利用し、パケットを配送するために用いられる。

LIN6 では、通信を開始する際に LIN6 ID と現在のネットワークプレフィクスとの対応付けを取得しなければならない。LIN6 では、この対応関係を **mapping** と呼び、mapping を管理する機構として **Mapping Agent (MA)** を導入する。MA は Mobile IPv6 における HA のように、ホームアドレスと CoA の管理情報を基にパケットを中継するのではなく、LIN6 ID とネットワークプレフィクスという動的な情報の管理を行い、要求に応じてネットワークプレフィクスを通知するという役割を担う。MA はある 1 台の移動ノードごとに 1 台必要な機能であるが、1 台の MA を複数の移動ノードで共有することも可能である。また、ある 1 台の移動ノードの MA は、移動する範囲に応じて複数配置することも可能であり、mapping を分散管理することも可能である。MA の分散配置に関しては、移動ノードを管理する管理者が MA の配置を決めるべきである。MA の配置の一例として、管理者は移動ノードが移動していく可能性がある組織の内の数ヵ所を選び、その組織の境界ルータ付近に配置するという方法が考えられる。この配置が望ましい理由は、MA を境界ルータ付近に配置することにより、MA と移動ノード間の通信遅

延をおさえることができるからである。

2.2.2 LIN6 の通信手順

LIN6 では、**図 2.4** で示すような手順で通信を行う。LIN6 における情報管理では、次のような前提をおいている。移動ノードと移動ノードの mapping を管理する MA の対応付けという静的な情報に関しては DNS (Domain Name System) を利用して管理を行い、移動ノードの LIN6 ID と移動ノードの現在のネットワークプレフィクスの対応づけという動的な情報に関しては MA を利用して管理を行う。

まず、移動ノードは MA へ現在のネットワークプレフィクスを登録する。登録の際には、IPsec の認証ヘッダを用い、移動ノードを認証する。通信ノードが通信を開始する場合には、まず DNS へ移動ノードの mapping を管理している MA を問い合わせる（図 2.4(1)）。DNS には、あらかじめ移動ノードの LIN6 ID とそれを管理する MA のアドレスという静的な対応情報を登録しているため、DNS から移動ノードの mapping を管理する MA を取得することが可能である（図 2.4(2)）。次に通信ノードは、移動ノード mapping を管理している MA へ現在のネットワークプレフィクスを要求する（図 2.4(3)）。MA は通信ノードへ事前に登録されているネットワークプレフィクスを通知する（図 2.4(4)）。この様な手順を踏むことにより、通信ノードは移動ノードの現在のネットワークプレフィクスを得ることが可能であり、この情報を基に LIN6 アドレスを構築し通信を開始できる（図 2.4(5)）。

移動ノードが移動した際には、MA へネットワークプレフィクスを更新するとともに、通信をしていた通信ノードに対しても新しいネットワークプレフィクスを通知する。この通知により、移動した際もパケットロスを最小限に抑えた通信をすることが可能である。

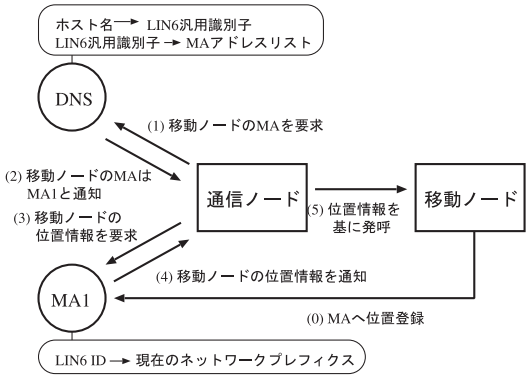

図 2.4.　LIN6 の通信手順

移動した際のネットワークプレフィクスの通知では、IPsec の認証ヘッダを用いて移動ノードを認証し、位置情報を更新しなければならない。もし、移動ノードが新しいネットワークプレフィクスを通知しているにもかかわらず、PKI のインフラが整備されていないなどの理由により IPsec を利用できず、認証できなかった場合、位置情報を更新することができない。この場合、新しいネットワークプレフィクスを取得するには、保持している移動ノードのネットワークプレフィクスがタイマで時間切れになった後、MA へ通知要求をしなければならないため、多大なパケットロスが生じてしまう。そこで、LIN6 の移動通知では、認証できない場合を想定し、移動したということを通知する機構を導入する。この通知を受けた通信ノードは、すぐに MA へ移動ノードのネットワークプレフィクスを要求し、MA からの返答に応じてネットワークプレフィクスを更新する。この機構の導入により、ネットワークプレフィクスを直接通知するよりはパケットロスが増大してしまうものの、移動後において認証ができない場合でも効率の良い通信を行うことが可能となる。

2.3 定性的な評価および比較

2.3.1 ヘッダオーバヘッド

まず、ヘッダのオーバヘッドについて議論する。Mobile IPv6 では、通信のたびに最大 48 バイトの拡張ヘッダを付加しなければならない。このようなヘッダオーバヘッドは、リアルタイムアプリケーション、特にデータパケットを小さくして配送を試みている VoIP などのストリーミング配送に多大な悪影響を及ぼすと考えられる。一方、LIN6 では、通信を行う際にオプションヘッダや拡張ヘッダなどは不要であり、IPv6 基本ヘッダのみを利用する。したがって、LIN6 では Mobile IPv6 のようなヘッダオーバヘッドは生じない。

2.3.2 通信経路

次に、パケットの通信経路について議論する。Mobile IPv6 で通信ノードが移動ノードに対してパケットを配送する場合、2.1.2 項で述べたように冗長経路での配送となる。また、定常状態の通信においても、IPsec[86, 85] による認証ができない場合には冗

長経路でのパケット配送になる。IPsec による認証は、PKI のインフラが整備されていないことや、不特定多数のノードと IPsec の Security Association (SA) を確立することが現実的でないと考えられるため、Mobile IPv6 の通信は冗長経路でのパケット配送となる。

LIN6 の場合には、通信ノードが現在のネットワークプレフィクスを保持していない状態でパケットを送信する際、移動ノードの MA から現在のネットワークプレフィクスを取得した後、通信を開始するという手順を踏む。したがって、つねに最適経路のパケット配送となる。

2.3.3 HA と MA の耐障害性

Mobile IPv6 における HA、LIN6 における MA は通信の核となっており、これらに障害が生じると移動ノードと通信することができなくなる。HA と MA の耐障害性について考察すると、一点障害を避けるための複製を配置することを考えた場合、2.1.2 項で述べたように HA は実質的には複製を配置することができない。しかし、MA は 2.2.1 項で述べたように LIN6 ID に依存せず、移動ノードの移動範囲にあわせて管理者が自由に複製を配置することが可能である。したがって、MA は HA に比べ耐障害性が高いと考えられる。

2.3.4 ファイアウォール内外における通信

ファイアウォールが存在するネットワークにおいて Mobile IPv6 や LIN6 を利用することについて考える。ホームリンクがファイアウォールの内側に存在し、通信ノードと移動ノードがファイアウォール外に存在する状況について考える。この状況では、移動ノードはファイアウォール外にいるにもかかわらず、Mobile IPv6 の場合、通信ノードから送信されたパケットは、ホームリンクがファイアウォール内に存在するため、HA へ到達することができず、移動ノードと通信できない。LIN6 の場合、もしファイアウォール内に MA が存在する場合においても、MA の複製をファイアウォール外に 1 つ配置すれば、ファイアウォール外の移動ノードと通信可能である。

2.3.5 IPv6 網に与える影響

IPv6 網に対して Mobile IPv6 で追加する機能としては HA があげられる。HA は IPv6 網の中では通常ルータとして扱われるため IPv6 網に対する影響はない。一方、LIN6 では IPv6 網に対して MA の機能を追加しなければならない。MA は IPv6 網の中では単なるノードであるため IPv6 網に対する影響は無い。しかし、LIN6 では、LIN6 ID から MA のアドレスを検索するために DNS を用いるため、TXT レコードなどを用い、登録するレコードを工夫するか、MA レコードを新たに追加するなどする必要がある。

2.3.6 LIN6 の Mobile IPv6 との共存可能性

LIN6 は、上記で述べたように Mobile IPv6 と比べ多くの定性的な利点を持つ。しかし、Mobile IPv6 は IETF で標準化が進んでおり、世界的に受け入れられているプロトコルなのに対し、LIN6 はまだ標準化が進んでいないプロトコルである。したがって、普及の段階では、標準化が進んでいる Mobile IPv6 との共存について考えなければならず、共存できないことは LIN6 の普及において大きな障壁となると考えられる。

しかしながら、LIN6 は、Mobile IPv6 に対して変更を加えることなく共存できる。その理由として、LIN6 ではパケットから得られたアドレスの構造で、LIN6 の通信か従来の IPv6 の通信かを判断するため、Mobile IPv6 の通信は通常の IPv6 の通信と見なすことが可能である。また、Mobile IPv6 側から LIN6 を見た場合には、LIN6 汎用識別子で示されたアドレスをホームアドレスとして使用していない限り、Mobile IPv6 側では Mobile IPv6 を利用しない通常の IPv6 の通信であると見なせる。つまり、LIN6 と Mobile IPv6 のそれぞれでアドレスの利用方法が異なるため、お互いのアドレスの利用方法が衝突しない限り、移動透過性を保証しない通常の IPv6 の通信であると見なして通信することができる。LIN6 汎用識別子のために割り当てているプレフィクス、すなわち LIN6 プレフィクスはあらかじめ定められている経路制御不可能な固定値であり、通常の通信を考えた場合には、そのプレフィクスがホームリンクを表すプレフィクスとして利用されることはない。したがって、Mobile IPv6 と LIN6 の共存は現実的であると考えられる。

2.3.7 定性的な評価のまとめ

ここで、上記で述べてきた LIN6 と Mobile IPv6

表 2.1.　LIN6 と Mobile IPv6 の定性的な評価のまとめ

	LIN6	Mobile IPv6
ヘッダオーバヘッド	IPv6 基本ヘッダのみ	最大 48 バイトの拡張ヘッダが必要
通信経路	常に最適経路での通信	認証できない場合は冗長経路での通信
耐障害性	MA の複製を柔軟に配置可能	HA はホームリンクにのみ配置可能
ファイアウォール内外の通信	MA の複製をファイアウォール内外に設置することで通信可能	HA の複製を配置できないため通信不可能
IPv6 網に与える影響	MA の機能を追加し、DNS に対しても変更が必要	HA の機能の追加が必要

の定性的な評価について 2.1 にまとめる。

表 2.1 にまとめたように、LIN6 は Mobile IPv6 と比べヘッダオーバヘッドが生じず、つねに最適経路での通信が可能なプロトコルである。さらに、通信の核である MA の耐障害性は、HA の耐障害性に比べ非常に高いということを示した。

2.4 評価に使用した実装について

本節では、基本性能評価で用いる LIN6 実装の基本設計・実装手法について述べる。なお、Mobile IPv6 の実装に関しては慶應 SFC の実装 [166][1] を用いた。

2.4.1 実装設計

LIN6 の核となる処理のほとんどは、IP パケットのアドレス変換である。したがって、LIN6 の実装は、KAME Project から配布されている IPv6 スタックに LIN6 のアドレス変換のための関数を加えることで実現する。

LIN6 ID と現在のネットワープレフィクスとの対応関係である mapping 情報は、ユーザ空間のプログラムに実装した。Mapping Agent も同様に、ユーザ空間のプログラムとして実装した。Mobile IPv6 の実装では、Binding 情報や Home Agent は全て kernel で処理している。

2.4.2 LIN6 の実装

本項では、LIN6 の実装を各要素ごとに説明する。LIN6 の実装ブロックダイアグラムを図 2.5 に示す。太線で表した部分が本研究で実装した部分である。また、矢印はデータの流れを示すものである。

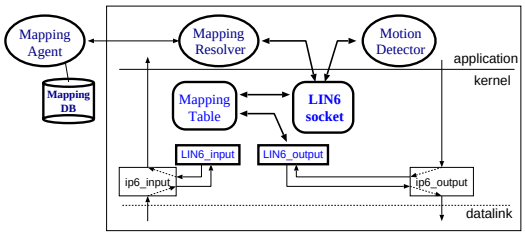

図 2.5.　実装ブロックダイアグラム

カーネル空間での処理

カーネル空間では、受信、送信ごとにパケットのアドレスを変換するための関数を経由するように実装を行った。

受信の際には、まず到達したパケットの宛先アドレスが LIN6 アドレスかどうかを判断し、宛先アドレスが LIN6 アドレスだった場合には、LIN6 汎用識別子へ変換する。LIN6 汎用識別子が得られると処理は終了し、上位層へ処理が渡される。上位層では、LIN6 汎用識別子を宛先アドレスとして処理が継続される。

送信の際には、パケットの宛先アドレスが、LIN6 汎用識別子であるかどうかを判別し、LIN6 汎用識別子であった場合には LIN6 ID を抽出する。次に、LIN6 アドレスを生成するため、抽出した LIN6 ID に対する現在のネットワークプレフィクスを MA から取得する。ネットワークプレフィクスと LIN6 ID から得られた LIN6 アドレスは、LIN6 汎用識別子と置き換えられ、宛先アドレスとしてパケットに埋めこまれる。その後は LIN6 アドレスでネットワーク層の処理が継続され、最終的にデータリンク層へと処理が渡される。

[1] 実装は、draft-ietf-mobileip-ipv6-13.txt をもとに実装されている

ユーザ空間プログラム

移動ノード上で mapping 情報を管理するプログラムである Mapping Agent や、カーネル空間の mapping 情報のキャッシュである Mapping Table の情報をやりとりをするためのプログラム、そして移動検出のためのプログラムを、ユーザ空間に実装した。これらのプログラムはそれぞれ Maping Resolver、Motion Detector と呼ばれ、Mapping Agent と通信したり、LIN6 sock を介して情報のやりとりなどを行う。

Mapping Agent

mapping 情報を管理するプログラムである Mapping Agent は、Mapping Resolver からの要求により、移動ノードに対する mapping 情報の通知や移動ノードの現在のインタフェースアドレスの登録などを行うユーザ空間のプログラムとして実装を行った。Mapping Agent の管理テーブルはハッシュテーブルとして実装し、LIN6 ID、ネットワークプレフィクス, ライフタイムなどが保持される。

2.5 定量的な評価および比較

2.5.1 実装環境

まず、本評価で用いた LIN6 の実装のモジュール図を**図 2.6** に示す。アドレス変換の処理に関してはすべてカーネル空間で行い、mapping の処理は一度ユーザ空間のプログラムで処理した後、カーネル空間へデータが渡される。カーネル空間とユーザ空間のプログラムのデータのやりとりは LIN6 socket と呼ばれる LIN6 専用のソケットを用意した。一方 Mobile IPv6 の実装[2]では、すべての処理をカーネル空間で行っている。詳しくは文献 [166] を参照していただきたい。

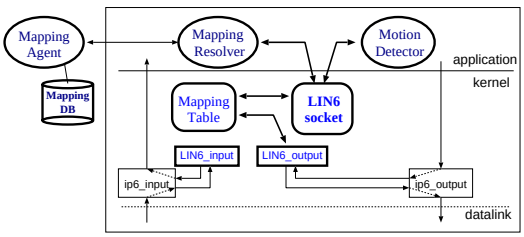

図 2.6. 実装モジュール図

2.5.2 内部の処理時間

どの処理過程でのオーバヘッドが大きいかを調べるために LIN6、Mobile IPv6 各々について、定常状態におけるカーネル内部の処理コストを測定した。

本測定のために用いた移動ノード、通信ノードの計算機は Mobile Pentium II 400MHz、128MB の PC であり、OS としては FreeBSD 4.2 + KAME-snap へ Mobile IPv6 や LIN6 を実装したものを用いた。内部処理時間の測定には Pentium Clock Counter を用い、LIN6、Moblie IPv6 それぞれのカーネル空間の関数である `ip6_input()`、`ip6_output()` 全体でかかる処理時間を測定した。また、本測定では比較対照として KAME の標準 IPv6 スタックの `ip6_input()`、`ip6_output()` についても全体の処理時間を測定し、比較を行った。移動ノードにおける送受信処理時間の測定結果を**図 2.7** に、通信ノードにおける送受信処理時間の測定結果を**図 2.8** にそれぞれ示す。測定では、送受信処理ともに ICMP Echo Request を通信ノードから移動ノード宛に 1 秒おきに 100 回送信した。そして、この処理にかかった時間を 1 回ずつ Pentium Clock Counter の値から算出し、その平均値を処理時間とした。

図 2.7、2.8 に示すように、移動ノード、通信ノードの送受信ともに処理時間は KAME IPv6 < LIN6 < Mobile IPv6 となった。この原因については次項で考察する。

2.5.3 カーネル空間の処理時間に対する考察

送受信それぞれについてオーバヘッドが生じる要因を考察する。本測定の通信では、定常状態、すなわち通信ノードが mapping 情報もしくは Binding Cache を保持している状態を仮定している。

LIN6 の IPv6 に対するオーバヘッドについては LIN6 の基本性能評価 [72, 162] で詳しく述べたため、本章では省略する。ここでは、Mobile IPv6 の IPv6 に対するオーバヘッドについて考える。Mobile IPv6 における通信ノードでは、送信処理として経路制御ヘッダを付加するオーバヘッドが生じ、受信処理ではホームアドレスオプションと認証ヘッダを解釈するオーバーヘッドが生じる。移動ノードでは、送信処理ではホームアドレスオプションと認証ヘッダを

[2] Mobile IPv6 の実装は draft-ietf-mobileip-ipv6-13.txt をもとに実装されている。

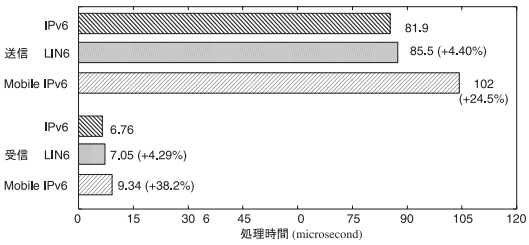

図 2.7.　定常時の送受信処理時間（移動ノード）

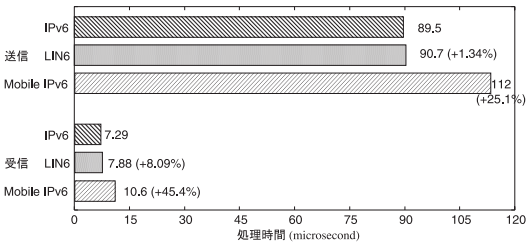

図 2.8.　定常時の送受信処理時間（通信ノード）

付加するオーバヘッドが生じ、受信処理では経路制御ヘッダを解釈するオーバヘッドが生じると考えられる。

LIN6 と Mobile IPv6 の送受信の処理コストを比較すると、LIN6 の処理オーバヘッドは IPv6 と比べたかだか数パーセントなのに対し、Mobile IPv6 の処理オーバヘッドは IPv6 より 20 パーセント以上高いことがわかった。この原因について考察すると、LIN6 では送受信ともに IPv6 ヘッダのアドレスを変換するという処理なのに対し、Mobile IPv6 では拡張ヘッダの付加、拡張ヘッダを解釈する処理が生じる。このため、アドレス変換よりもヘッダ処理のオーバヘッドが大きくなり、処理オーバヘッドが大きくなってしまったと考えられる。

2.5.4 評価環境

本評価では、実験ネットワークとして**図 2.9** に示すようなネットワークを構築し、LIN6、 Mobile IPv6 のプロトコルを用いた通信全体で生じる処理オーバヘッドを測定した。測定項目は、登録処理時間、1 パケット目の処理時間、定常状態での処理時間である。

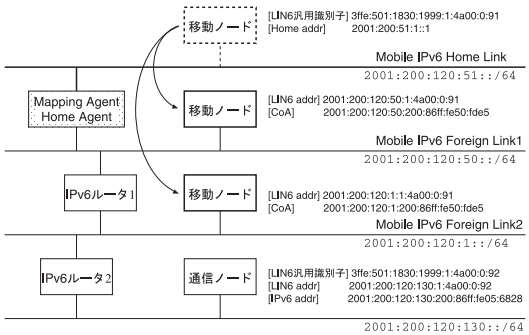

図 2.9.　実験ネットワーク

まず、図 2.9 に示すネットワークについて説明する。測定のために構築したネットワークは、一番上から Mobile IPv6 におけるホームリンク、外部リンク 1、外部リンク 2 となる。それぞれのネットワークは、IPv6 PC ルータで接続されており、ホームリンクと外部リンク間のルータは HA や MA の実装が行われているルータである。なお、LIN6 ではホームリンク、外部リンクの区別がないため、便宜的に Mobile IPv6 の用語でそれぞれのネットワークを識別する。通信ノードは一番下のネットワークに接続しており、移動しないものとする。また、移動ノードは、ホームリンクから外部リンクへ移動しながら通信ノードと通信を行う。移動ノードや通信ノードの横に示しているアドレスは、Mobile IPv6 や LIN6 で保持しなければならないアドレスである。LIN6 の場合は LIN6 汎用識別子と LIN6 アドレス、Mobile IPv6 の場合はホームアドレスと CoA である。

本測定では、通信ノード側に mapping(Mobile IPv6 では binding) がない状態で、移動ノードへ ICMPv6 の Echo Request を送信した場合の処理時間を測定した。なお、移動ノードは Mobile IPv6 外部リンク 1 (2001:200:120:50::/64) に接続している状態で測定を行った。

2.5.5 登録処理時間

まず、位置登録にかかる処理時間について測定した。移動ノードからの位置登録の処理時間は、LIN6 の場合、移動ノードから MA へ登録を行い、その確認応答を受信するまでの時間を測定した。LIN6 の登録処理時間は約 1.6 ms. となった。この時間を詳細に調べると、MA でのユーザ空間の処理時間は 270 μs.、MA にパケットが到着してから確認応答パケットを返信するまでの処理時間は 310〜950 μs. であった。処理時間に幅が存在するのはカーネル空間での位置登録パケットの処理時間が一定でないことを示している。また、移動ノードと MA 間の RTT は広域の場合 m のオーダとなるため、登録によるオーバヘッドはほとんどないと考えられる。

一方、Mobile IPv6 の場合は、移動ノードから HA へ binding の更新を行い、その確認応答を受信する

までの時間を測定した。Mobile IPv6 の登録処理時間は 300 ms. であった。Mobile IPv6 の場合、LIN6 と比べ多大なオーバヘッドが生じてしまっている。この多大なオーバヘッドは、確認応答を返信する際、データの piggyback を行うために確認応答パケットを一度 queue に貯め、タイマが時間切れになってから送信されることが原因であると考えられる。この queue のタイマは 1 秒であるため、最大 1 秒の通信処理時間がかかってしまう。確認応答へデータを piggyback させる機構は実装依存であるが、本測定で用いた実装は、つねに piggyback するためこのような多大なオーバヘッドが生じている。

登録処理時間について比較すると、Mobile IPv6 は一度 queue へ貯めてから確認応答を返信するため、LIN6 に比べ多大なオーバヘッドが生じることが分かった。したがって、移動が頻繁になればなるほど、Mobile IPv6 の処理オーバヘッドが大きくなると考えられ、非効率的である。

2.5.6 1 パケット目の処理時間

LIN6 における 1 パケット目の処理過程を、**図 2.10** の I に示す。処理時間は tcpdump コマンドで測定した値であり、RTT は ping6 コマンドで測定した参考値である。LIN6 における 1 パケット目の処理は次のような手順となる。まず、通信ノードは MA へ移動ノードの現在のネットワークプレフィクスを要求する。そして、現在のネットワークプレフィクスを受信した後 ICMP Echo Request を送信する。移動ノードは通信ノードの現在のインタフェースアドレスを保持していないため、MA から通信ノードの現在のネットワークプレフィクスを取得しなければならない。移動ノードは、通信ノードの MA へ通信ノードの現在のネットワークプレフィクスを要求する。移動ノードはその通知を受信した後、通信ノードへ ICMP Echo Reply を返信する。通信ノードと移動ノード間の通信の 1 パケット目はお互いの現在のネットワークプレフィクスを取得する必要があり、次に送信されるパケットの処理時間と比較すると処理コストが非常に大きい。

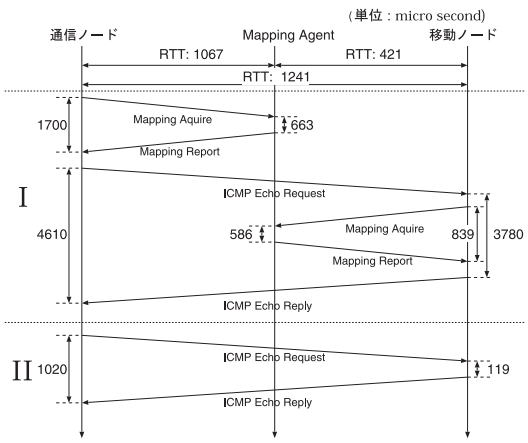

図 2.10. LIN6 の処理過程

次に、Mobile IPv6 における 1 パケット目の処理過程を**図 2.11** の I に示す。通信ノードはまずホームアドレス宛に ICMP Echo Request パケットを送信する。通信ノードから送信されたパケットは、HA で処理され、トンネリングにより移動ノードの CoA に対して転送される。ICMP Echo Request を受信した移動ノードは、通信ノードに対して Home Address オプションと認証ヘッダを追加して ICMP Echo Reply を返信する。

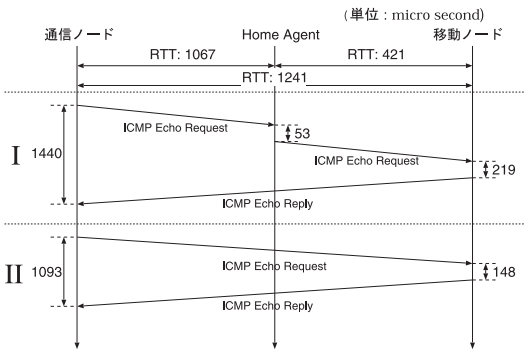

図 2.11. Mobile IPv6 の処理過程

処理時間について比較すると、通信ノードがパケットを送信しようとしてから 1 パケット目の確認応答を受け取るまでの処理時間は、LIN6 の場合は 6.3 ms. なのに対し、Mobile IPv6 の場合は 1.4 ms. となり、LIN6 の方が 4 倍程度処理時間がかかることが分かった。LIN6 の場合、移動ノードのネットワークプレフィクスを通信を開始する前に取得しなければならず、さらにその処理はユーザ空間プログラムで行っているため、コストが非常に大きくなってしまっていることが分かる。しかし、LIN6 における 2 パケット目以降はすべてカーネル空間で処理され、定常状態と同じ処理コストで送受信可能である。一方、Mobile IPv6 の場合、1 パケット目だけをみると処理コストは LIN6 より小さいが、2 パケット目以降も HA を介した通信となる可能性がある。HA を介した冗長経路でのパケット配送は、通信ノード側に Binding

Cache が生成されるまで継続されるためトータルの処理コストとしては、LIN6 と同等かそれ以上かかる可能性もある。また、今回の測定では、移動ノードと通信ノードの最適経路上に HA を配置して測定を行ったが、現実的には HA が最適経路上に存在する可能性は非常に少ないと考えられる。HA が最適経路にない場合に、実際の広域通信でかかる処理コストは、HA を経由するためにかかる RTT 分増大すると考えられる。広域通信の場合の RTT は ms. のオーダであるため、結果として LIN6 よりもオーバヘッドが増大する可能性もある。

2.5.7 定常状態の処理時間

LIN6、Mobile IPv6 における定常状態の通信を図 2.10、図 2.11 の II に示す。LIN6 では、移動ノードと通信ノードお互いのネットワークプレフィクスを保持した状態での通信となり、つねに最適経路でのパケット配送となる。Mobile IPv6 では、通信ノードで Binding Cache が生成されたことを仮定しているが、この場合 HA を介することなく直接移動ノードへとパケットが配送される。両者の定常状態の処理時間について比較すると、LIN6 の処理時間は約 1.0 ms.、Mobile IPv6 の処理時間は約 1.1 ms. となった。RTT のばらつきを考慮すると、ほぼ同じ処理コストだが、若干 LIN6 の方が性能が良い。Mobile IPv6 の場合、通信中も定期的に IPsec の処理を行い、ヘッダの処理が必要であることを考慮すると、平均的に処理コストが大きくなる可能性があると考えられる。

2.5.8 定量的評価のまとめ

測定した LIN6 と Mobile IPv6 の定量的な評価を表 2.2 にまとめる。

定量的な評価では、LIN6 は Mobile IPv6 と比べ、1 パケット目の処理において処理コストが最悪の場合 4 倍程度高くなってしまった。しかし、これは HA の配置が最適な場合にのみ言えることであり、HA の実際の配置を考えた場合には、移動ノードと HA 間の通信遅延に処理時間は大きく影響することが容易に予想される。また、実際に移動体通信で用いられる VoIP ストリーミングのようなアプリケーションを考えた場合、1 パケット目のオーバヘッドに比べ、その後の通信の処理オーバヘッドの方が大きく影響する。さらに、本評価における定常状態での処理時間測定は、Mobile IPv6 の通信で Binding Cache が作られ、最適経路で通信できることを前提としてデータを測定したが、2.3.2 項で述べたように、実際には最適経路での通信をすることは困難ではないかと考えられる。したがって、通常の通信全体の処理時間を考慮すると、Mobile IPv6 に比べ、LIN6 の方が処理コストは低いといえる。

表 2.2.　LIN6 と Mobile IPv6 の定量的な評価のまとめ

	LIN6	Mobile IPv6
内部処理の比較	アドレス変換の処理で通常の IPv6 と比べ、数% の処理オーバヘッドが生じる	オプションヘッダの処理で通常の IPv6 と比べ、20%以上の処理オーバヘッドが生じる
登録処理	AH の処理時間が μs. のオーダでかかる	piggyback のためのタイマにより ms. のオーダの処理時間がかかる
1 パケット目の処理	MA へ位置情報を問い合わせるコストが大きく、worst case では Mobile IPv6 の 4 倍	冗長経路の通信となるため HA の位置によって処理時間が変化
初期の数パケットの処理	遅延は常に移動ノード間の通信遅延となる	Binding Cache ができるまでは冗長経路の通信となり、LIN6 よりも遅延が多い
定常状態の処理	通常の IPv6 の通信とほぼ変わらない	IPsec の処理が頻繁に行われるため 10%程度オーバヘッドが大きくなる可能性がある

2.6 まとめ

本章では、IPv6 上の新しい移動透過保証プロトコルである LIN6 の性能評価を行った。

定性的な比較では、LIN6 の方が Mobile IPv6 に対してヘッダオーバヘッドや耐障害性の面で有利であることを示した。また、定量的な比較を行うために LIN6 の実装と Mobile IPv6 の実装の処理時間について測定した。実装の処理時間測定では、全体的な処理時間として登録にかかる処理時間、1 パケット目の処理時間、定常状態での処理時間を測定し、さらに定常状態でのカーネル空間内部のの処理時間についても測定を行った。その結果、LIN6 は Mobile IPv6 と比較し、1 パケット目での処理では多少劣っているものの、全体の処理として低オーバヘッドで移動透過性を実現可能であることを示した。

次世代のプロトコルスタックについて考えると、近い将来 IEEE802.11 を中心とした無線のインフラが至るところで整備され、VoIP などのストリーミングアプリケーションが利用されるようになり、モバイルアプリケーションに適応するトランスポート技術も数多く登場することが予想される。そして、ネットワーク層に限らずプロトコルスタックは様々に変化していくと考えられる。我々はこのような環境においても、プロトコル自体は上下層のプロトコルとは独立して考えることが重要であると考えているが、ネットワーク層がリンク層の情報を利用してハンドオフ処理を効率的に行うなど、層間における情報のやりとりのためのインタフェースを定義することは必要であろう。

LIN6 の今後の課題として、LIN6 の実装を公開することにより、広域実験等を行いたいと思う。さらに、LIN6 の実装の公開だけでなく標準化を行ったり、層間の情報のやりとりのためのインタフェースなどを提案することにより、LIN6 を次世代の基盤ネットワークプロトコルとして普及していくよう努力していきたいと考えている。

第 3 章　おわりに

本年度は、IPv6 における新しい移動透過保証プロトコルである LIN6 プロトコルの性能評価と Mobile IPv6 との定性的および定量的な比較検討の結果について報告した。

Rover では、今後も定期的なミーティングを通して、移動計算機に適したファイルシステムや、移動透過性保証のためのプロトコルといったモバイルコンピューティングに関する議論を活発に行っていく予定である。なお、Rover のより詳細な活動については、`http://www.imobility.org/`を参照されたい。