LIN6のマイクロモビリティサポート

原田 友紀子[†,☆] 國 司 光 宣[†] 寺 岡 文 男[†,††]

本稿では，IPv6上のモビリティプロトコルであるLIN6にマイクロモビリティサポート機能を付加したHierarchical LIN6（HLIN6）を提案し，設計，実装，評価する．HLIN6では，ネットワークの階層化とLIN6のアドレス構造を応用したデータ配送方法によりマイクロモビリティサポートを実現する．そのため，従来のマイクロモビリティプロトコルに比べデータ配送時のヘッダオーバヘッドが少ない等の利点を持つ．HLIN6をKAME ProjectのIPv6スタックとLIN6 kernelを利用し実装した．評価を行った結果，HLIN6はLIN6に比べ，無視できる程度のオーバヘッドでマイクロモビリティサポートを実現可能であることが分かった．

Micro Mobility Support in LIN6

Yukiko Harada,[†] Mitsunobu Kunishi[†] and Fumio Teraoka[†,††]

This paper describes a micro mobility protocol called Hierarchical LIN6 (HLIN6). HLIN6 is based on LIN6 and supports node mobility in IPv6 under the environment where mobile nodes move frequently. HLIN6 employs a hierarchy method in its design. HLIN6 does not use tunneling for routing in a “Domain” because HLIN6 makes good use of the LIN6 address architecture. Thus, HLIN6 has no header overhead. We implemented a prototype system of HLIN6 in the KAME IPv6 stack and the LIN6 kernel. The results of evaluation of HLIN6 show that the processing time specific to HLIN6 is negligible and that HLIN6 supports micro mobility.

1. はじめに

インターネットにおける移動ノード（Mobile Node: MN）の通信のためのモビリティプロトコルとしては，Mobile IP [1),2)]の標準化が進んでいる．しかしMobile IPには，Home Agent（HA）を1つしか配置できないために耐故障性が乏しい，トンネリングやオプションヘッダの多用によりデータ配送時のヘッダオーバヘッドが大きい，最適経路での通信が容易ではなく運用上はMNへのパケットは冗長経路を通る等の問題がある．そこで，当研究室では，Mobile IPの問題点を解決するモビリティプロトコルであるLIN6 [3)]を開発し，実装，標準化を進めている．LIN6では次世代インターネットプロトコルであるIPv6 [4)]を用いることを前提としているが，これはIPv4 [5)]のアドレス不足等の問題を解決するプロトコルとしてIPv6は重要な技術であると考えているためである．

しかし，LIN6やMobile IP等のモビリティプロトコルには共通する問題があり，MNの頻繁な移動を考慮していないため，MNが小さな無線セル間を頻繁に移動する場合には，以下の問題が発生する．まず，MNは移動時に位置登録のための制御メッセージを送受信するため，頻繁な移動を行うとバックボーンのバンド幅が浪費されてしまう．また，MNが位置登録をするノードとMN間の通信遅延が大きい場合，位置登録に時間がかかるため，古い位置情報に従って配送されたパケットが損失する．移動が頻繁に発生する場合は，この位置登録遅延によるパケット損失が多量になるという問題がある．このようなMNの頻繁な移動時に発生する問題を解決するためのプロトコルを，マイクロモビリティプロトコルと呼ぶ．Mobile IPに対しては多数のマイクロモビリティプロトコルが提案されているが，LIN6に対しては提案されていなかった．本稿では，LIN6にマイクロモビリティサポート機能を付加したHierarchical LIN6（HLIN6）を提案し，実装，評価する．

† 慶應義塾大学大学院理工学研究科
Graduate School of Science and Technology, Keio University
†† ソニーコンピュータサイエンス研究所
Sony Computer Science Laboratories, Inc.
☆ 現在，日本アイ・ビー・エム株式会社
Presently with IBM Japan, Ltd.

2. LIN6 概要

まず，本稿でマイクロモビリティサポート機能を付加する LIN6 の概要を説明する．

2.1 LIN6 の基本概念

LIN6では，ネットワークアドレスが位置に関する情報（位置指示子）とノード自体を識別する情報（ノード識別子）を分離することなく保持していることを問題として提起している．この問題はネットワークアドレスの二重性と呼ばれ，LIN6ではこの問題を解決するため，ネットワークアドレスが保持している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している．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て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て経路制御を行うことで，移動透過性を保証する．LIN6 のノード識別子は LIN6 ID と呼ばれ，EUI-64 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，ネットワークの位置情報を示す位置指示子として，LIN6 では既存の 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 1** に示す．また，送受信におけるアドレスの変換手順を**図 2** に示す．

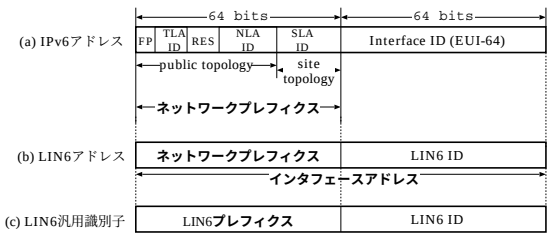

図 1 IPv6 アドレス・LIN6 アドレス・LIN6 汎用識別子
Fig. 1 Address of IPv6・address of LIN6・LIN6 generalized ID.

図 1 (a) は，既存の IPv6 ユニキャストアドレスの構造を示したものであり，上位 64 bit がノードのネットワークへの接続点を示すネットワークプレフィクス，下位 64 bit がノードが接続したネットワークで一意となるようなインタフェース識別子という構造である[6]．図 1 (b) 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 アドレスはパケットを配送するために用いられる．図 1 (c) は，LIN6 汎用識別子と呼ばれる 64 bit の LIN6 ID を 128 bit に拡張した識別子である．LIN6 汎用識別子は，あらかじめ決められた 64 bit の固定値に対して 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．LIN6 汎用識別子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，このアドレスを利用して発呼することにより位置に依存しない発呼が可能となる．

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づ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 IP における HA のようにホームアドレスと Care of Address（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．

2.2 LIN6 の通信手順

LIN6では**図 3** で示すような手順で通信を行う．まず，移動ノード（MN）は Access Router（AR）が送信する Router Advertisement（RA）から得た（図 3 (A1)）現在のネットワークプレフィクスを，MN の MA である MA1 に登録する（図 3 (A2)）．このときに送受信

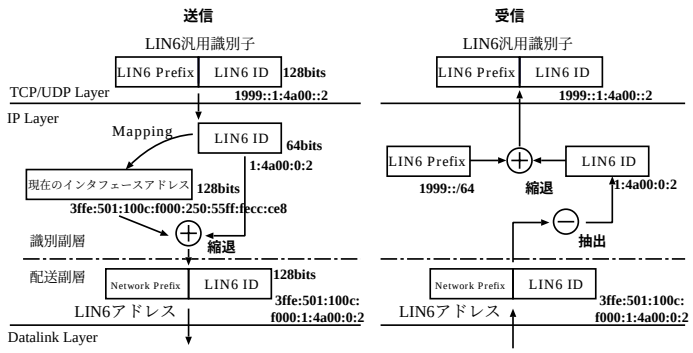

図 2 LIN6 のアドレス変換手順
Fig. 2 Address exchange procedure of LIN6.

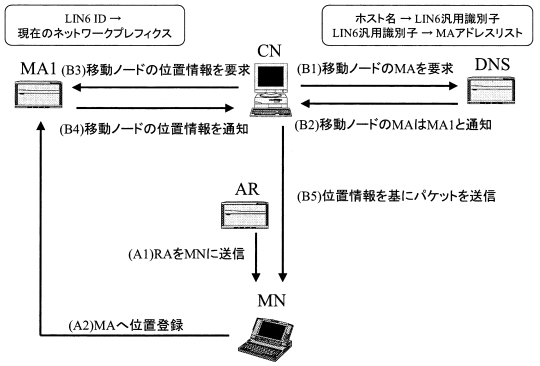

図 3 LIN6 の通信手順
Fig. 3 Communication procedure of LIN6.

されるメッセージを Mapping Registration/Ack と呼ぶ．通信ノード（CN）が通信を開始する場合には，まず DNS に対して MN の mapping を管理している MA を問い合わせる（図 3 (B1)）．DNS には，あらかじめ MN 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 MN の mapping を管理する MA を取得することが可能である（図 3 (B2)）．次に CN は，MN の mapping を管理している MA に対して，現在のネットワークプレフィクスを要求する（図 3 (B3)）．このときに送信するメッセージを Mapping Acquire と呼ぶ．Mapping Acquire を受信した MA は，CN に対して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 3 (B4)）．この通知メッセージを Mapping Report と呼ぶ．このような手順を踏むことにより，CN は MN 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 3 (B5)）．

次に MN が移動した場合の処理について説明する．MN が移動先の AR から RA を受信（図 3 (A1)）し移動を検知した際には，Mapping Registration メッセージを MA に対して送信することにより，ネットワークプレフィクスを更新する（図 3 (A2)）．なお，CN と通信中に MN が移動した場合には，CN に対しても Mapping Update メッセージで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，MN が移動した際もパケット損失をある程度抑えた通信をすることが可能である．しかし，MN の移動が頻繁である場合や MN と CN の距離が離れている場合には，多量のパケット損失や多量の制御メッセージによるバンド幅の浪費が生じる．このため，本稿では LIN6 にマイクロモビリティサポート機能を付加する．

3. 関連研究

LIN6 にマイクロモビリティサポート機能を付加する際の，プロトコル設計のために，既存の Mobile IP のためのマイクロモビリティプロトコルについて分析を行った．IETF で活発に議論されているプロトコル等，主要とされている Mobile IP に対する 5 つのプロトコルと Mobile IPv6 に対する 5 つのプロトコルについて，データの配送方法に注目して Host Routing 方式，Hierarchy 方式，Bicasting 方式，Forwarding 方式の 4 つの方式に分類し，分析を行った．

Host Routing 方式は，MN の頻繁な移動を隠蔽する範囲として Domain という概念を導入することによりネットワークを階層化し，Domain 内移動時の位置登録等の処理を Domain 内部のみで行うことによってマイクロモビリティを実現する方式である．Domain 内のルーティングは Host Routing により行うため，Domain 内移動時のバックボーンへの負荷や位置登録遅延を低減可能であるが，Domain 内のすべてのルータに変更が必要であり導入が困難であるという問題がある．Cellular IP [7]，Cellular IPv6 [8]，HAWAII [9] は Host Routing 方式に分類できる．

Hierarchy 方式も Domain という概念を導入しネットワークを階層化することによりマイクロモビリティを実現しているが，Domain 内でも Mobile IP に対応した IP ルーティングを行うため Host Routing 方式とは区別する．Hierarchy 方式は，Host Routing 方式と同様に Domain 内移動時のバックボーンへの負荷や位置登録遅延を低減可能であるが，Domain 内のデータ配送時にトンネリングによるオーバヘッドが発生するという問題がある．Hierarchical Mobile IP [10]，Hierarchical Mobile IPv6 [11] は Hierarchy 方式に分類できる．

Bicasting 方式は，HA や Hierarchy 方式の Domain 境界に設置するルータ等において MN へのパケットをコピーし，現在接続している位置と移動する位置の両方にパケットを送信することによりマイクロモビリティを実現する．コピーした 2 つ以上のパケットを配送する（bicasting）ため，移動時のパケット損失を回避できるが，Layer 2（L2）の情報が必要でありプロトコルが複雑であるという問題がある．Fast Handoff [12]，Simultaneous Binding [13] は Bicasting 方式に分類できる．

Forwarding 方式は，MN の移動後に MN が以前に接続していた位置に配送されたパケットを，新しい位置に転送することによりマイクロモビリティを実現す

る．Bicasting方式と同様に移動時のパケット損失を回避できるが，L2の情報が必要でありプロトコルが複雑であるという問題がある．Low Latency Handoff[14]，BETH[15]はForwarding方式に分類できる．

4. HLIN6の提案

本稿で提案するHierarchical LIN6（HLIN6）の目的，設計指針，ネットワーク構成等のプロトコルの概要について説明する．

4.1 HLIN6の目的

HLIN6は，頻繁に移動しながらLIN6を用いて通信する際の以下の問題点を解決することを目的として，提案し設計を行う．

- 位置登録遅れのためのパケット損失に起因する通信性能の悪化
- 位置登録のための制御パケットによるバックボーンにおけるバンド幅の浪費

4.2 HLIN6の設計指針

HLIN6の設計にあたり，指針とする事項を以下に示す．

(1) 制御パケットが頻繁にバックボーンへ流出することを防止し，バックボーンのバンド幅の浪費回避と，位置登録遅延の低減によるパケット損失率低下を実現する．

(2) プロトコルを簡潔にするため，MNの移動時の多少のパケットロスは容認し，パケット損失回復はFECやTCP等の上位層プロトコルに頼る．

(3) LIN6のアドレス構造を応用し，データ配送時のヘッダオーバヘッドを回避する．

既存研究の分析結果より，指針の(1)を満たすためにはHost Routing方式かHierarchy方式が有効である．Hierarchy方式にはDomain内のデータ配送時にヘッダオーバヘッドが生じるという問題があるが，指針の(3)で示すようにLIN6のアドレス構造を利用することによりトンネリングを行わないためヘッダオーバヘッドを回避可能である．したがってHLIN6ではHierarchy方式を採用する．LIN6にHierarchy方式を適用しているため，本稿で提案するプロトコルをHierarchical LIN6（HLIN6）と呼ぶことにする．

LIN6にHierarchy方式を適用し階層化することにより，Domain内の頻繁なMNの移動時には，LIN6を用いた通信に比べ以下の効果が期待される．

- パケット損失を低減
- バックボーンへの制御パケットの流出防止

Hierarchy方式は，Bicasting方式やForwarding方式と組み合わせることも可能である．しかしBicasting方式やForwarding方式はプロトコルが複雑であり指針の(2)を満たさないため，HLIN6ではHierarchy方式のみを採用する．

また，Hierarchy方式ではDomainを多階層に構成することも可能であるが，本設計ではプロトコルを簡潔に保つために多重の階層化は行わず，あるDomainは他のDomainに属さないことを前堤とする．

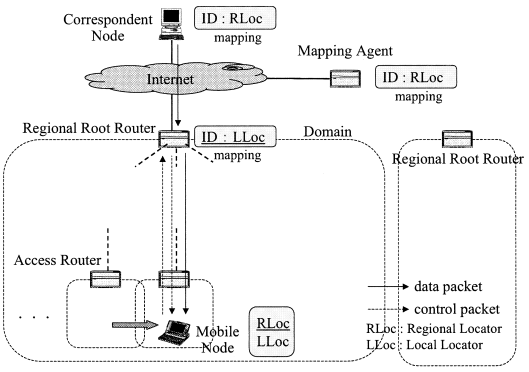

図4 HLIN6のネットワーク構成
Fig. 4 Network composition of HLIN6.

4.3 HLIN6のネットワーク構成

設計指針に従い設計したHLIN6のネットワークの構成について説明する．HLIN6のネットワーク構成を**図4**に示す．

HLIN6のネットワークは，通常のLIN6のネットワークの構成要素であるMA，CN，MNに加え，MNの頻繁な移動を外部に隠蔽する範囲であるDomainという概念を導入する．また，Domain内の移動をDomain外部に隠蔽するために必要となる，位置登録処理の受付けとLocator（後述）の書き換え処理を行うRegional Root Router（RR）も導入する．RRはDomainとバックボーンの境界に配置する．さらに，MNのDomain間移動検出やRRへの登録処理を可能とするRouter Advertisement（RA）を送信するために，本プロトコルに対応したAccess Router（AR）が必要である．

HLIN6ではDomain外のルーティングに用いるRegional Locator（RLoc）と，Domain内のルーティングに用いるLocal Locator（LLoc）という2種類の位置指示子であるLocatorを導入する．あるDomainに接続しているMNの位置は，Domain外ではRLocを用いて，Domain内ではLLocを用いて表される．RLocはDomainを表す64 bitのネットワークプレフィクスであり，MNがDomain内で移動する際には変化しない．LLocはDomain内でMNが接続してい

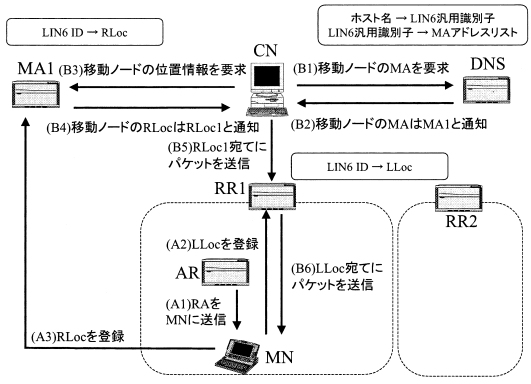

図 5 HLIN6 の通信手順
Fig. 5 Communication procedure of HLIN6.

るサブネットを表す 64 bit のネットワークプレフィクスであり，AR 間の移動の際はつねに変化する．MA や CN は，Mapping Table に MN の LIN6 ID と RLoc の対応関係を保持し，RR は MN の LIN6 ID と LLoc の対応関係を保持する．また MN は RLoc と LLoc を保持する．

4.4 HLIN6 の通信手順

HLIN6 の通信手順について説明する．HLIN6 の通信手順を**図 5** に示す．

まず，移動時の処理について図 5 を参照しながら説明する．MN が RA を受信する（図 5 (A1)）ことにより新しい Domain に移動したことを検知した際には，RLoc と LLoc が変化するため MN はまず LLoc を RR に登録する（図 5 (A2)）．このとき MN と RR 間で送受信されるメッセージを，Regional Mapping Registration/Ack と呼ぶことにした．次に MN は RLoc を MA に登録する（図 5 (A3)）．このとき MN と MA 間で送受信されるメッセージは LIN6 と同様であり，Mapping Registration/Ack である．さらに MN が CN と通信中である場合は CN に RLoc の変化を知らせるため，MN は Mapping Update を CN に送信する．この際，RR に障害が発生した等何らかの理由で RR への登録が失敗したことを検出すると，MN は LLoc を MA に，通信中の場合は CN にも LLoc を登録し直し，通常の LIN6 の通信ができるように対処する．MN が RA を受信し（図 5 (A1)）Domain 内でサブネット間を移動したことを検知した際には，LLoc のみが変化するため RR へ LLoc が登録される（図 5 (A2)）のみであり，RR と MN 間で Regional Mapping Registration/Ack が送受信される．

ここで，移動検出について説明する．MN は新しいサブネットにおいて AR から RA を受け取ることにより，ネットワークプレフィクスの変化を知り，AR 間の移動を検知する．これは従来の LIN6 と同様であり，Domain 内移動時にも Domain 間移動時にも利用される情報である．HLIN6 では，さらに Domain 間の移動も検知する必要があるが，これは AR が送信する RA に RLoc の情報を付加し，その値を比較することにより Domain 間の移動を検知する．

以上で説明した移動時の処理を行うため，HLIN6 は LIN6 に比べ Domain 内の移動の場合は，位置登録のための制御メッセージによるバックボーンのバンド幅の浪費を防ぐこと，MA や CN への登録処理による遅延がなくなるために遅延の少ない移動処理（パケット損失の低減につながる）を実現することが可能になる．

次に，データパケットの配送手順について図 5 を参照しながら説明する．MN へのデータパケットは，RR（図 5 の場合は RR1）までは通常の LIN6 の手順に従い配送される（図 5 (B1) から (B5)）．RR までは，データパケットの宛先 IP アドレスの上位 64 bit（LIN6 アドレスにおいて位置を表す部分）には，RLoc（図 5 の場合は RLoc1）が設定されている．MN へのデータパケットが RR に到着すると，RR では宛先 IP アドレスの下位 64 bit（LIN6 アドレスにおいてノードを識別する部分）から MN の LIN6 ID を取得する．そして，RR において MN の LIN6 ID を基に RR の Mapping Table を検索し，現在 MN が接続しているサブネットのネットワークプレフィクスである LLoc を得る．そして宛先 IP アドレスの RLoc を LLoc に書き換え，MN に向けて再び転送する（図 5 (B6)）．LIN6 のアドレス構造では上位 64 bit のみで位置を表しているため，このようなアドレス書き換えが可能になっている．MN から通信相手へのデータパケットの送信手順は LIN6 と同様であり，RR でのアドレス書き換え等の処理はいっさい行われない．

なお，同一 Domain 内に接続している MN どうしが通信する場合は RR を経由した通信になり，ルーティングパスが冗長になる．しかし，MN と RR 間の距離は位置登録遅延を改善できるほど短いことが前提であり，この冗長なパスによる影響は実用上問題ないと考えられる．

このように，HLIN6 では Domain 内のルーティングはトンネリングではなく LIN6 のアドレス構造を応用したアドレスの書き換えにより行うため，Hierarchy 方式を採用している既存のマイクロモビリティプロトコルとは異なり，データパケットのヘッダオーバヘッドのない通信が可能になる．

4.5 RR の冗長化による対故障性の向上

Domain が 1 台の RR によって外部のインターネッ

トに接続するという形態では，RRが故障するとそのRRのDomain内のMNは外部のインターネットとの通信ができなくなる．そのため，Domainは複数のRRによって外部インターネットに接続する形態が望ましい．HLIN6では以下に示す方法により，Domainが複数のRRによって外部インターネットと接続できるようにし，対故障性を高めている．

Domain内に複数存在するRRはそれぞれ別のRLocを持ち，RR間にはあらかじめ優先度を設ける．Domainに存在するすべてのRRに関するRLocと優先度は，ARからRAによってMNへ通知される．MNはRAで通知されたすべてのRRにLLocを登録するとともに，すべてのRLocと優先度をMAに登録する．

MNと通信を開始しようとするCNはMAにMNの位置情報を問い合わせるが，MAはこのMNに関して登録されているすべてのRLocと優先度をCNに返答する．CNはまず優先度が最も高いRLocを使用してパケットを送信するが，もしこのRLocに対応するRRが故障していると，途中でICMPによるエラーメッセージが返送されるか，またはMNからの応答がタイムアウトする．するとCNは次に優先度の高いRLocを使用してパケットを再送する．このようにすることにより，CNとMNが通信中にその通信が経由しているRRが故障しても，CNがそれを検出することができ，別のRRを経由する通信に継続することができる．LIN6の性質により，RLocの切替えはトランスポート層以上には影響を与えないため，TCPやUDPの通信はそのまま継続できる．

5. HLIN6の実装

本稿で提案するHLIN6を，KAME Projectが配布しているIPv6スタックとLIN6 kernelを用いて実装した．HLIN6の実装に際し，LIN6に対して以下の追加または拡張を行った．

- ARの拡張
  - RRに関する情報（RLoc等）を含むRouter Advertisementの送信機能
- MNの拡張
  - 拡張されたRouter Advertisementの受信機能
  - Domain間の移動検知機能
  - RRへLLocを登録しMAへRLocを登録する機能
- RRの導入
  - （LIN6 ID:LLoc: Lifetime）のTable管理機能
  - パケットのアドレス書き換え機能

なお今回の実装では，HLIN6の基本的な性能評価を行うことが目的であるため，RRの冗長化に関する実装は行っていない．

現在，HLIN6はBSD系のOS上で動作しており，ライセンスフリーなソフトウェアとして公開している．

6. 評　　価

6.1 定性的な評価

HLIN6と3章で説明した既存のマイクロモビリティプロトコルとを比較するために，定性的な評価を行う．現時点で実装されている既存のマイクロモビリティプロトコルは存在しないため，本稿では他のマイクロモビリティプロトコルとの比較として定性的な評価を行う．

3章で示したように数々のマイクロモビリティプロトコルが提案されているが，HLIN6と同じHierarchy方式を採用し，またIPv6を前提としているHierarchical Mobile IPv6（HMIPv6）を比較対象とした．HLIN6とHMIPv6の定性的な特徴を比較した結果を，**表1**に示す．

表1で示すように，HLIN6ではDomain内のデータ配送時にトンネリングは行わずアドレスを書き換えるため，データのヘッダオーバヘッドのないプロトコルになっている．さらに，Stateで表したメモリオーバヘッドの点でもHLIN6は優れている．HMIPv6ではノード識別子と位置指示子に各々16 bytesのアドレスを用いているため，ノード識別子と位置指示子の対応関係を持つためには32 bytes必要である．一方，HLIN6ではノード識別子もLLoc等の位置指示子も8 bytesであるため，Tableで保持すべきエントリのバイト数はHMIPv6の半分となる．

以上の結果より，HLIN6はHMIPv6に比べ，ヘッ

表1 HLIN6とHMIPv6の定性的な評価
Table 1 Qualitative evaluation of HMIPv6 and HLIN6.

	HMIPv6	HLIN6
モビリティプロトコル	Mobile IPv6	LIN6
データ配送時のヘッダオーバヘッド	有	無
RRの持つStateのバイト数	32 bytes*MN	16 bytes*MN

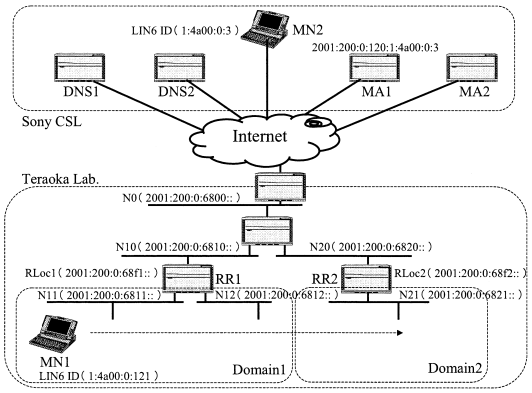

図 6 実験ネットワークの構成
Fig. 6 Composition of this experiment network.

ダオーバヘッドがなく，Domain 境界のルータにおけるメモリオーバヘッドが少ないプロトコルであるといえる．

6.2 定量的な評価

HLIN6 の定量的な評価を，**図 6** に示す実験ネットワークを構築し行った．MN における移動処理時間，RR におけるアドレス書き換え処理時間，RR における登録処理時間の 3 つの項目に対して測定した．

6.2.1 MN における移動処理時間の測定

1 つ目の評価として，HLIN6 にマイクロモビリティ機能が付加できているかを確認するため，MN における移動処理時間を，LIN6 の移動時，HLIN6 の Domain 内移動時，HLIN6 の Domain 間移動時の各場合について測定した．移動処理時間とは，MN が移動（接続する AR を変更）した際に新しい位置の登録等の処理に費やす時間であり，この時間内は MN へのパケットを正しく配送できないためパケット損失が発生する可能性がある．この移動処理時間を短くすることにより，通信中の移動時のパケット損失が低減し，通信性能を改善することが可能になる．HLIN6 は，位置登録ノードを Domain 内に設置し，Domain 内の頻繁な移動時に移動処理時間を短くすることによりマイクロモビリティサポートを実現している．よって，HLIN6 と LIN6 の移動処理時間を比較することにより，HLIN6 のマイクロモビリティサポート機能が正しく動作しているかを確認できる．

まず，HLIN6 の Domain 内移動時と LIN6 の移動時の移動処理時間の測定結果を**図 7** に示す．図 7 の測定値は，MN，MA，RR における tcpdump により得た結果であり，単位は ms である．図 7 に示すように，LIN6 や HLIN6 というプロトコルの違いにかかわらず，MN の移動時にはまず Layer 2（L2）のハンドオーバ処理が行われる．次に，接続しているサブネットのプレフィクス情報等を得るために MN が AR へ Router Solicitation（RS）を送信し，その応答である Router Advertisement（RA）を受信するという IPv6 のプロトコルに従った処理が行われる．そして，RA の情報を基にネットワークインタフェースのアドレスを付け替える処理が行われる．そのため，移動処理時間の中のこれらの処理に費やす時間は，HLIN6 や LIN6 というプロトコルや MN の移動パターン（Domain 内移動，Domain 間移動）にかかわらず一定である．つまり，HLIN6 と LIN6 の移動処理時間の違い，また移動パターンの違いによる登録処理時間の違いに影響を与える要素は登録処理時間のみであるといえる．登録処理時間は，登録先ノードにおける処理時間と，MN と登録先ノード間の Round Trip Time（RTT）を加えた値になる．登録先ノードにおける処理時間は，RTT に比べ小さな値になることが多く，また変動の大きな値ではなくつねに約 0.5 ms 程度である．そのため，登録処理時間は MN と登録先ノード間の RTT に依存するといえる．図 7 では，登録先ノードは LIN6 の場合は MA，HLIN6 の Domain 内移動の場合は RR となり，MN と MA 間の RTT よりも MN と RR 間の RTT の方が小さくなっているため，HLIN6 の Domain 内移動の場合の移動処理時間の方が小さくなっている．

次に，HLIN6 の Domain 間移動時と LIN6 の移動時の移動処理時間の測定結果を**図 8** に示す．

図 7 に対する考察と同様に，移動処理時間は MN と登録先ノード間の RTT に依存するため，HLIN6 の Domain 間移動の場合は MN と RR 間，MN と MA 間の RTT に影響を受けることになる．ここで図 8 から分かるように，MN と RR 間の RTT よりも MN と MA 間の RTT の方が大きく，また Regional Mapping Registration の応答を待たずに Mapping Registration を送信可能である．そのため HLIN6 の Domain 間移動時の移動処理時間は，MN と MA 間の RTT に依存すると考えられ移動処理時間は LIN6 の場合よりも多少大きい程度になるといえる．

これらの結果より，各プロトコル，各移動パターンでの移動処理時間は以下の式で表すことができる．

- LIN6
  移動処理時間
  $$= L2 + 30.2 + RTT_{(MN-MA)}(ms) \quad (1)$$
- HLIN6（Domain 内移動時）
  移動処理時間
  $$= L2 + 30.3 + RTT_{(MN-RR)}(ms) \quad (2)$$
- HLIN6（Domain 間移動時）

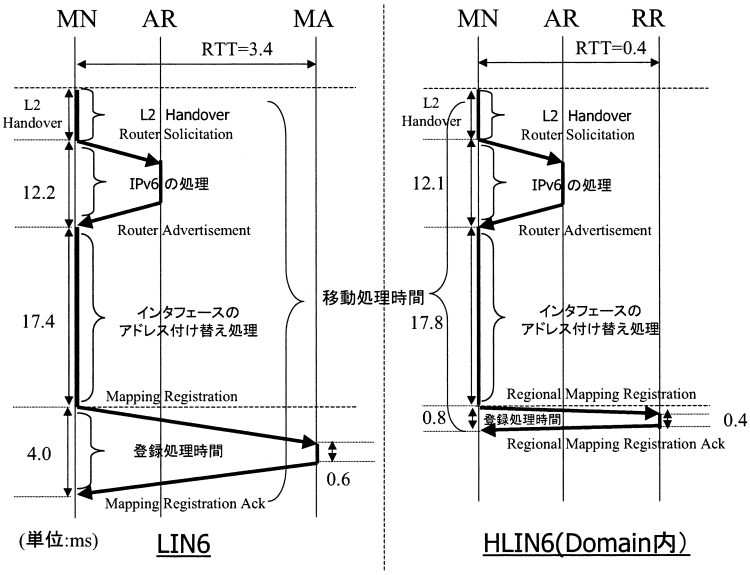

図 7 移動処理時間の測定結果（1）
Fig. 7 The measurement result of move processing time in MN (1).

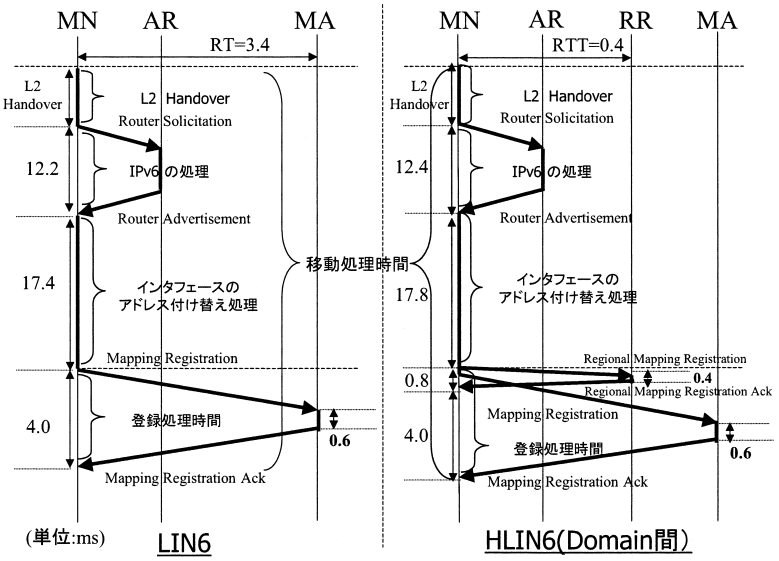

図 8 移動処理時間の測定結果（2）
Fig. 8 The measurement result of move processing time in MN (2).

移動処理時間

$$\simeq L2 + 31.2 + RTT_{(MN-MA)}(ms) \qquad (3)$$

上式の $L2$ とは L2 ハンドオーバの処理にかかる時間であり，本測定では値として求められなかったためこのように示している．2 項目は，IPv6 の処理時間，インタフェースのアドレス付け替え処理時間，登録先ノードにおける処理時間を足した値である．これらの式は，前述したように 1 項目と 2 項目は HLIN6 や LIN6 というプロトコルの違いや移動パターンの違いには影響されないためほぼ一定であると見なすことができるため，移動処理時間は MN と登録先ノード間の RTT に依存することを示している．ただし，HLIN6 の Domain 間移動時の式では，MN と RR 間の RTT よりも MN と MA 間の RTT の方が大きく，また Regional Mapping Registration の応答を待たずに Mapping Registration を送信可能であるため，

$RTT_{(MN-RR)}$ による影響は無視できる程度であると考え，移動処理時間には $RTT_{(MN-MA)}$ のみを加えるようにしている．

LIN6 では MA を複数配置可能であるが，MN の近くにつねに配置することは，MA 間のデータの一貫性を保つことを考えると現実的ではなく，$RTT_{(MN-MA)}$ は 100 ms 近くなる可能性もある．一方，RR は MN と同じ Domain 内に存在し通常 MN との距離は近いため，$RTT_{(MN-RR)}$ は大きくても数 ms 程度であると考えられる．よって，HLIN6 の Domain 内移動時には移動処理時間が小さくなり，LIN6 に比べ移動したことによるパケット損失を低減可能であるといえ，HLIN6 では LIN6 にマイクロモビリティサポート機能を付加できていることが分かった．

6.2.2 RR におけるアドレス書き換え処理時間の測定

2 つ目の評価として，RR におけるアドレス書き換え処理時間を測定した．RR におけるアドレス書き換え処理時間は，LIN6 では発生しない HLIN6 特有の MN へのデータ配送時のオーバヘッドとなる処理である．この処理時間が大きすぎると MN へのデータ配送時の遅延が増大し，通信性能に悪影響を与えることになり，HLIN6 の欠陥となりうる．そのため，アドレス書き換え処理時間が RR に登録されているエントリ数にかかわらず許容できる値であるかを，本測定結果より考察する．登録されているエントリ数にかかわらず許容できる値であることは，RR におけるアドレス書き換え処理の規模拡張性を保証できることを意味する．

図 9 に，アドレス書き換え処理時間の測定結果を示す．測定は Pentium Clock Counter を用いて行い，RR2 の `ip6_input()` において kernel 内の Mapping Table をパケットの宛先 IP アドレスの下位 64 bit（LIN6 ID）で検索し，検索した LLoc で宛先 IP アドレスの上位 64 bit（RLoc）を書き換えるまでの時間を測定した．RR に登録されているエントリ数が 1，10，100，1,000，5,000，10,000，25,000，50,000 の各場合について，登録されているエントリ中の 1 エントリを利用してアドレス書き換えを行う場合の処理時間を 10 回測定し，その平均値を測定値とした．横軸のエントリ数は対数表示をしている．

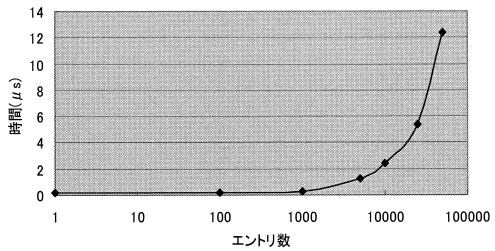

図 9 RR におけるデータ配送時のアドレス書き換え処理時間
Fig. 9 The measurement result of processing time for rewriting of a destination address in RR.

アドレス書き換え処理に費やす時間の大部分は，Mapping Table からエントリを検索する時間である．Mapping Table はハッシュテーブルであるため，エントリ数が増えるとハッシュキーが衝突し，衝突した場合は線形リストで管理されているため処理時間が増加してしまう．今回の実装ではハッシュテーブルサイズは 64 であり，テーブルサイズを大きくすれば図 9 のグラフの勾配を減らすことは可能である．

ただし，RR では Domain 内のノード数分のエントリを保持すればよく，現実的には RR のエントリ数は 1,000 程度になると考えられる．図 9 を見るとエントリ数が 1,000 程度までは増加率が低くなっており，通常 1 つの Domain には 1,000 程度のノードが存在することを考慮すると，必要程度の規模拡張性は保証できるといえる．また，数値的には 1,000 エントリで 0.25 μs，10,000 エントリでも 2.5 μs であり，データ配送時のオーバヘッドは無視できるほど小さいことが分かった．

6.2.3 RR における登録処理時間の測定

3 つ目の評価として，RR における登録処理時間について測定した．登録処理時間が小さな値であることは，移動時のパケット損失を低減可能であることを意味し，通信性能の向上につながる．RR における登録処理時間は，図 7 や図 8 において MN と RR 間の RTT に比べ小さくなることを示したが，移動処理時間の測定時には RR には MN 以外の他のエントリが登録されていなかった．そのため，本測定により RR における登録処理時間が RR に登録されているエントリ数にかかわらず，RTT に比べ小さな値になるかを考察する．登録されているエントリ数にかかわらず RTT に比べ小さな値となることは，RR における登録処理の規模拡張性を保証できることを意味する．

図 10 に，登録処理時間の測定結果を示す．測定は Pentium Clock Counter を用いて行い，RR2 の Mapping Agent Daemon において Regional Mapping Registration を受信してから，そのエントリを userland と kernel 内の Mapping Table に登録し Regional Mapping Registration Ack を送信するまでの時間を測定した．RR に登録されているエントリ数が，1，10，100，1,000，5,000，10,000，25,000，50,000

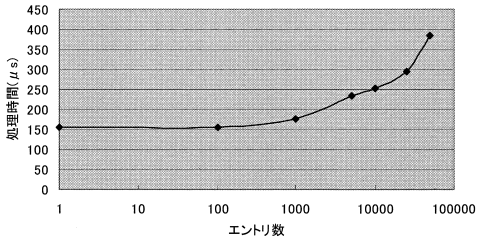

図 10 RRにおける登録処理時間
Fig. 10 The measurement result of processing time for registration of an LLoc in RR.

の各場合について，登録されていない新しい1つのエントリを登録する際の処理時間を10回測定し，その平均値を測定値とした．横軸のエントリ数は対数表示をしている．

登録するエントリはMapping Tableのリストの最後尾に追加されていくため，登録処理時間のエントリ数による増加はMapping Tableの検索時間の違いにより発生する．アドレス書き換え処理時間の場合と同様に，Mapping Tableはハッシュテーブルであるため，エントリ数が増加するとハッシュキーの衝突により処理時間が増加する．しかし，エントリ数が1000程度はまでは増加率も少なく，数値的にも170 μsに抑えられており，RRにおける登録処理時間についても必要程度の規模拡張性は保証できるといえる．

7. まとめ

本稿ではLIN6にマイクロモビリティサポート機能を付加したHLIN6を提案し，その概要，実装，評価について述べた．定性的な評価より，HLIN6はHierarchy方式を採用しつつもアドレス書き換えというLIN6の特性を生かしたプロトコルにしたために，LIN6の利点を受け継ぐとともに，帯域とメモリのオーバヘッドを低減できるマイクロモビリティプロトコルとなっていることが分かった．また，定量的な評価より，HLIN6ではLIN6にマイクロモビリティサポート機能を付加できていること，HLIN6で新しく導入したRRが必要程度の規模拡張性を有すること，RRでのアドレス書き換え処理は無視できるほどのオーバヘッドであることを確認した．

今後の課題としては，Domain内移動時の認証，RRの冗長化の実装がある．

参考文献

1) Perkins, C.: IP Mobility Support, Internet RFC 3344, *IETF* (2002).
2) Johnson, D., Perkins, C. and Arkko, J.: Mobility Support in IPv6, Internet Draft, *IETF* (2002).
3) Ishiyama, M., Kunishi, M., Uehara, K., Esaki, H. and Teraoka, F.: LINA: A New Approach to Mobility Support in Wide Area Networks, *IEICE Trans. Communication*, Vol.E84-B, No.8, pp.2076–2086 (2001).
4) Deering, S. and Hinden, R.: Internet Protocol, Version 6 (IPv6) Specification, Internet RFC 2460, *IETF* (1998).
5) Postel, J.: Internet Protocol, Internet RFC 791, *IETF* (1981).
6) Hinden, R., O'Dell, M. and Deering, S.: An IPv6 Aggregatable Global Unicast Address Format, Internet RFC 2374, *IETF* (1998).
7) Campbell, A., Gomez, J., Wan, C.Y., Kim, S., Turanyi, Z. and Valko, A.: Cellular IP, Internet Draft, *IETF* (2000).
8) Shelby, Z.D., Gatzounas, D., Campbell, A. and Wan, C.Y.: Cellular IPv6, Internet Draft, *IETF* (2001).
9) Ramjee, R., Porta, T.L., Thuel, S., Varadhan, K. and Wang, S.Y.: IP micro-mobility support using HAWAII, Internet Draft, *IETF* (2000).
10) Gustafsson, E., Jonsson, A. and Perkins, C.E.: Mobile IPv4 Regional Registration, Internet Draft, *IETF* (2002).
11) Soliman, H., Castelluccia, C., El-Malki, K. and Bellier, L.: Hierarchical MIPv6 mobility management, Internet Draft, *IETF* (2002).
12) El-Malki, K. and Soliman, H.: Fast Handoffs in Mobile IPv4, Internet Draft, *IETF* (2000).
13) El-Malki, K. and Soliman, H.: Simultaneous Bindings for Mobile IPv6 Fast Handoffs, Internet Draft, *IETF* (2001). draft-elmalki-mobileip-bicasting-v6-00.txt.
14) El-Malki, K., Calhoun, P.R., Hiller, T., Kempf, J., McCann, P.J., Singh, A., Soliman, H. and Thalanany, S.: Low Latency Handoffs in Mobile IPv4, Internet Draft, *IETF* (2002).
15) Kempf, J., Calhoun, P., Dommety, G., Thalanany, S., Singh, A., McCann, P.J. and Hiller, T.: Bidirectional Edge Tunnel Handover for IPv6, Internet Draft, *IETF* (2001).

(平成14年 3 月25日受付)
(平成14年10月 7 日採録)

原田友紀子

1977年生．2002年慶應義塾大学大学院理工学研究科前期博士課程修了．在学時代，IPv6，モビリティプロトコルに興味を持ち，LIN6の改良と応用の研究にたずさわる．現在，日本IBM（株）金融第一サービス事業部勤務．

國司　光宣

1976年生．2001年慶應義塾大学大学院理工学研究科前期博士課程修了．現在，同大学院同研究科後期博士課程に在学中．ネットワーク層でのモビリティサポートの研究，IPv6の改良に興味を持ち，LIN6のプロトコル設計，実装にたずさわる．

寺岡　文男（正会員）

慶應義塾大学理工学部情報工学科教授．1959年生．1984年慶應義塾大学大学院工学研究科電気工学専攻修士課程修了．同年キヤノン株式会社入社．1988年株式会社ソニーコンピュータサイエンス研究所入社．2001年4月から現職．博士（工学）．1991年日本ソフトウェア科学会高橋奨励賞受賞．1993年元岡記念賞受賞．2001年情報処理学会平成12年度論文賞受賞．コンピュータネットワーク，オペレーティングシステム，分散システム等の研究に従事．特に移動透過性を提供するプロトコルVIP（Virtual IP）の開発を通してIETFのMobile IP分科会の活動に貢献．2000年5月から2002年5月まで情報処理学会理事．著書に「ワイヤレスLANアーキテクチャ」（共著，共立出版）．監訳に「詳解Mobile IP」（共監訳，プレンティスホール出版）．ACM，IEEE，日本ソフトウェア科学会，電子情報通信学会各会員．