

LIN6 に基づくネットワークモビリティプロトコルの設計と実装

坂野 あゆみ[†] 大 岩 拓 馬[†]
國 司 光 宣^{††} 寺 岡 文 男[†]

本論文では、LIN6 に基づくネットワークモビリティプロトコルである LIN6-NEMO を提案し、設計・実装・評価を行う。現在 IETF で議論されている NEMO Basic Support Protocol は Mobile IPv6 を拡張したネットワークモビリティプロトコルであり、Mobile IPv6 と同様、データ配送時の冗長経路・トンネリングによるヘッダオーバーヘッド・耐故障性の問題をかかえている。LIN6-NEMO では LIN6 のアドレス構造を応用し、これらの問題を解決する。また、Domain という概念を導入し、大量の制御メッセージが外部ネットワークに流出することを防止する。本論文では LIN6-NEMO を NetBSD 1.6.1 Release 上に実装し、性能評価を行った。その結果から、LIN6-NEMO はわずかなオーバーヘッドで移動ネットワークに移動透過性を保証可能なプロトコルであることを実証する。

Design and Implementation of a Network Mobility Protocol Based on LIN6

AYUMI BANNO,[†] TAKUMA OIWA,[†] MITSUNOBU KUNISHI^{††}
and FUMIO TERAOKA[†]

In this paper, we propose LIN6-NEMO which is a network mobility protocol based on LIN6, and perform design, implementation, and evaluation. NEMO Basic Support Protocol which is being discussed in IETF is a network mobility protocol based on Mobile IPv6. NEMO Basic Support Protocol as well as Mobile IPv6, however, has some fundamental problems in its communication procedures, such as redundant routing, header overhead due to tunneling, and single point of failure. LIN6-NEMO applies the address structure of LIN6, and solves these problems. Moreover, LIN6-NEMO introduces the concept of Domain and it prevents that a lot of control messages flow into the Internet. We implemented LIN6-NEMO on NetBSD 1.6.1 Release, and performed evaluation. According to the result, we prove that LIN6-NEMO achieve providing migration transparency for a mobile network with a slight overhead.

1. はじめに

Mobile IPv6¹⁾ や LIN6²⁾ のようなホストモビリティプロトコルは単一のノードに移動透過性を提供可能である。しかし、今後は LAN (Local Area Network) が構築された乗物内で乗客がパソコンを接続して通信しながら移動したり、個人の携帯端末機器が PAN (Personal Area Network) を形成して移動したりする、ネットワークモビリティが重要になると考えられる。乗物に構築された LAN や PAN のような移動ネットワークに対し、既存のホストモビリティプロトコルを用いるとネットワークに不必要な制御パケッ

トが大量に流出し、遅延をもたらす可能性がある。したがって、ネットワークが移動する際は、ネットワーク全体として位置管理を行い、移動ネットワークに対して移動透過性を提供する効率的なモビリティプロトコルが必要になる。このようなプロトコルはネットワークモビリティプロトコルと呼ばれる。ネットワークモビリティプロトコルはホストモビリティプロトコルをできるだけシンプルに拡張して実現されるのが望ましい。

現在、IETF (Internet Engineering Task Force) では Network MObility (NEMO) Basic Support Protocol³⁾ の標準化が進められている。しかし、NEMO Basic Support Protocol は、Mobile IPv6 に基づいているため、データ配送時の冗長経路やトンネリングによるヘッダオーバーヘッドの問題を継承している。また、移動ネットワークがネストするとこれらの問題はより深刻になる。

[†] 慶應義塾大学大学院理工学研究科

Graduate School of Science and Technology, Keio University

^{††} メディアエクスチェンジ株式会社

Media Exchange, Inc.

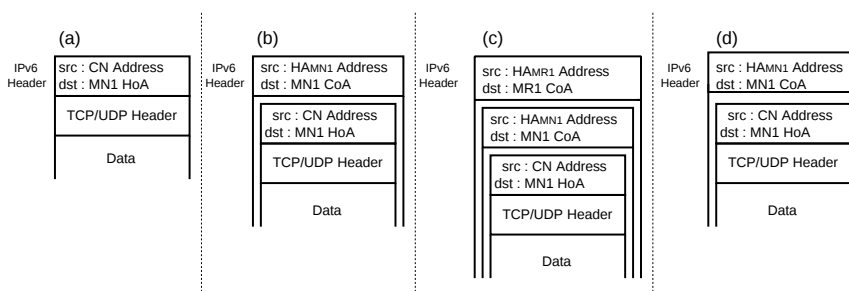


図 2 CN から MN1 への通信時のヘッダ構成

Fig. 2 Header structure of CN to MN1.

そこで、本論文ではデータ配送時のヘッダオーバーヘッドがなく、つねに最適経路で配送できる LIN6 に基づいて、移動ネットワークに対して移動透過性を保証するネットワークモビリティプロトコルである LIN6-NEMO を提案する。

2. NEMO Basic Support Protocol の概要と問題点

2.1 MONET とは

インターネット上のあるサブネットからなるネットワークがその内部のトポロジを変えることなく全体として移動し、インターネット上の接続点を動的に変更するネットワークのことを移動ネットワーク (MONET) と呼ぶ。MONET は外部インターネットとの境界ルータである Access Router (AR) に Mobile Router (MR) を通して接続する。図 1 のように、MONET は MR と Mobile Node (MN) で構成され、ネスト可能である。ネストとは、図 1 の MR3 のように、ある移動ネットワークが、他の移動ネットワークのサブネットに接続する状態のことである。

2.2 動作概要

NEMO Basic Support Protocol の動作概要を図 1、図 2 を用いて説明する。図 1 の (a) から (d) は、図 2 の (a) から (d) に対応している。また、図中の HA_{MN1} や CN_{MN1} などの表記は、それぞれ MN1 の位置情報を管理する Home Agent (HA)、MN1 の通信相手であるノード Correspondent Node (CN) を表している。

NEMO Basic Support Protocol では、MN や MR は移動によって変化しないノード識別子としてのアドレスである Home Address (HoA) と、移動先リンクで一時的に割り当てられる位置指示子としてのアドレスである Care-of Address (CoA) を保持する。HoA と CoA の対応関係は、MN や MR のホームリンクに設置される HA と呼ばれるノードが管理する。MR は

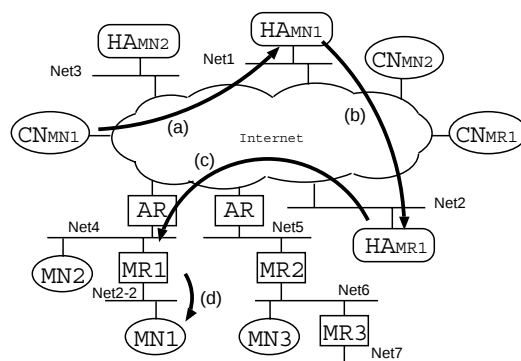


図 1 CN から MN1 への通信におけるパケットの流れ

Fig. 1 Packet flow of CN to MN1.

1 つ以上の HoA を持つことができる。

MR は AR に接続すると CoA を取得し、MR の位置を管理する HA_{MR} に Binding Update を送信する。Binding Update は、MN や MR の移動時に HA へ CoA を登録する際や、MN が通信ノード (CN_{MN}) に CoA と HoA の対応づけを通知するために用いられる。CoA と HoA の対応づけを binding という。 HA_{MR} は Binding Update を受信すると MR の HoA に CoA を対応づけ、位置登録エントリを生成する。

MR が MONET 内のノードに接続性を提供するかどうかは、HA への Binding Update 中の R フラグによって判断される。HA は R フラグが設定されている MR から Binding Update を受信した場合、MR の下位サブネットのネットワークプレフィックスに対し転送経路を設定する。また、HA は Binding Update の確認応答として MR へ Binding Acknowledgement を送信する。

通信ノード (CN_{MN1}) から MONET 内のノード (MN1) への通信について説明する。 CN_{MN1} は MN1 の HoA 宛にパケットを送信する (図 1 (a))。HoA 宛に送信されたパケットはまず HA_{MN1} に到達し、 HA_{MN1} によって MN1 の CoA へトンネリングされ、 HA_{MR1}

に転送される(図1(b)). トンネリングとは、パケットを別のIPヘッダでカプセル化して送信する手法である。 HA_{MR1} へ転送されたパケットはMR1のCoAへトンネリングされ、MR1に到達する(図1(c)). パケットがMR1に到達すると、カプセル化により付加されたIPヘッダを除去し、 HA_{MN1} が作成したパケットを取り出してMN1に転送する(図1(d)). このように、複数のHAを経由してパケットを配送することをPin-ball routingという。

また、MONET内のノード(MN1)から通信ノード(CN_{MN1})への通信では逆の手順を踏む。そうすることで双方向トンネルが機能し、MONET内のノードに移動透過な通信が提供される。

2.3 問題点

2.2節の説明から分かるように、NEMO Basic Support Protocolには冗長経路、ヘッダオーバーヘッド、耐故障性の3つの問題点がある。NEMO Basic Support Protocolでは複数のHAを経由してパケットが配送されるため、通信ノードからMONET内のノードへの経路は冗長経路となる。また、HAからMONET内のノード宛のパケットはトンネリングされて転送されるので、ヘッダオーバーヘッドが生じる。ネストしているMONET内のノードと通信する場合は、複数のHAを経由してパケットが配送されるため、複数回のトンネリングが行われてヘッダオーバーヘッドの増大と、Pin-ball routing問題により深刻なパケット遅延を引き起こす可能性がある。また、NEMO Basic Support Protocolでは、送信されるパケットは必ずHAを経由するため、HAが接続されているリンクに障害が発生すると通信を継続することができない。

これらは、NEMO Basic Support ProtocolがMobile IPv6に基づいて設計されていることに起因する問題である。これらの問題点を解決しようとする試みもあるが、Mobile IPv6に基づいているため本質的な解決にはなっていない^{(4)~(6)}。

また、冗長経路の解決方法としてMobile IPv6の階層化技術⁽⁷⁾に基づいたネットワークモビリティプロトコル⁽⁸⁾も提案されているが、ヘッダオーバーヘッドの問題は解決されていない。また、MONET内のMN

がCNやroot-MRにBinding Updateを送信する際のCNにおけるMNの認証やroot-MRにおけるMNの認証が具体的には考察されていない。

そこで本論文では、冗長経路、ヘッダオーバーヘッド、耐故障性の3つの問題点を克服し、認証機構を備えたネットワークモビリティプロトコルであるLIN6-NEMOを提案する。

3. LIN6-NEMO の設計

3.1 設計指針

LIN6-NEMO⁽⁹⁾は、データ配送時のヘッダオーバーヘッドがなく、つねに最適経路で配送できるIPv6でのモビリティプロトコルであるLIN6に基づいたネットワークモビリティプロトコルである。LIN6-NEMOはNEMO Basic Support Protocolと同様、MNがMONET内外に自由に移動でき、任意数のMONETのネストを可能とする。設計にあたり、前章までで述べたNEMO Basic Support Protocolの問題点をふまえ、以下の項目を設計指針とする。

- (1) MONET単位での局所的な移動の際に、制御パケットが不必要に外部のネットワーク上へ送信されることを回避する。
- (2) LIN6の特性を活かし、MONETのネストの階数によらずつねに最適経路で、かつヘッダオーバーヘッドなしに通信することを可能とする。

3.2 LIN6 アドレスの利用

LIN6で用いられるアドレスを図3に示す。図3(a)はLIN6アドレスと呼ばれ、LIN6で実際にパケット配送に用いられるアドレスである。これは、上位64bitにはIPv6アドレスと同様のネットワークプレフィックスを格納してノードの接続点を示し、下位64bitにはインターネット上で一意となるノード識別子のLIN6 IDを格納したものである。図3(b)のLIN6汎用識別子は64bitのLIN6 IDを128bitに拡張したもので、上位64bitのLIN6プレフィックスと呼ばれるあらかじめ定められた固定値と下位64bitのLIN6 IDからなる。これは位置に依存しないIPv6アドレスである。と見なすことができる。トランスポート層以上では、このLIN6汎用識別子を用いて位置に依存しないコネ

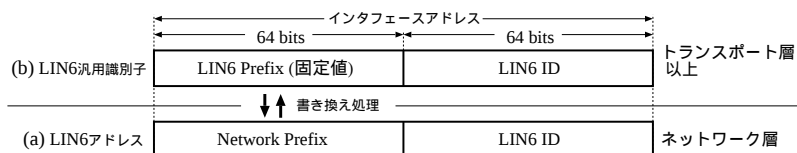


図3 LIN6のアドレス構造

Fig. 3 Address structure of LIN6.

クションを確立し、ネットワーク層では LIN6 アドレスを用いて経路制御をすることで移動透過性を保証できる。LIN6 アドレスの上位 64 bit をパケット転送中に書き換えてもトランスポート層以上には影響がない。LIN6-NEMO ではこの性質を利用し設計指針 (2) を実現する。

LIN6 では、ノードの LIN6 ID と現在のネットワークプレフィックスとの対応関係をマッピングと呼び、このマッピングを保持する機構として Mapping Agent (MA) を設置する。MA は任意の場所に複数個設置可能である。

3.3 Domain の導入

HLIN6 (Hierarchical LIN6)¹⁰⁾ と同様に、LIN6-NEMO では MN や MONET の移動を外に隠蔽する範囲を規定する Domain という概念を導入する。MONET 内のノードや MN が Domain 内を移動する場合、Domain はその移動を Domain 外部に対し隠蔽する。Domain 内移動を外部に対し隠蔽するとは、以下の 2 つの意味を持つ。

- (1) MONET 内のノードや MN の移動処理の制御パケットが、外部ネットワークに流出しない。
- (2) Domain 外部のノードが、Domain 内部の MONET 内のノードや MN の移動による影響を受けない。

つまり、Domain という概念を導入することで設計指針の (1) を実現できる。

LIN6-NEMO では、ノードの位置を表す 2 種類の Locator を導入する。Regional Locator (RLoc) と Local Locator (LLoc) である。ある Domain 内に接続しているノードの位置を Domain 外では RLoc を用いて表し、Domain 内では LLoc を用いて表す。RLoc と LLoc の対応関係は Domain を管理する RRR (Regional Root Router) で保持される。図 4 は LIN6-NEMO のネットワーク構成の概要であり、網かけ部分は MONET を表している。図 4 の move (A) のように MONET が移動すると、MR や MN の RLoc と LLoc はともに変化する。このような移動は Domain 間移動と呼ばれる。また、図 4 の move (B) のように MONET が移動すると、MR や MN の LLoc は変化するが RLoc は変化しない。このような移動は Domain 内移動と呼ばれる。

3.4 多段ネストのサポート

LIN6-NEMO では MONET の任意数のネストをサポートする。そのため、AR は下位サブネットに接続した MR に Prefix Delegation^{11)~13)} で動的にネットワークプレフィックスを割り当て、拡張 Router Ad-

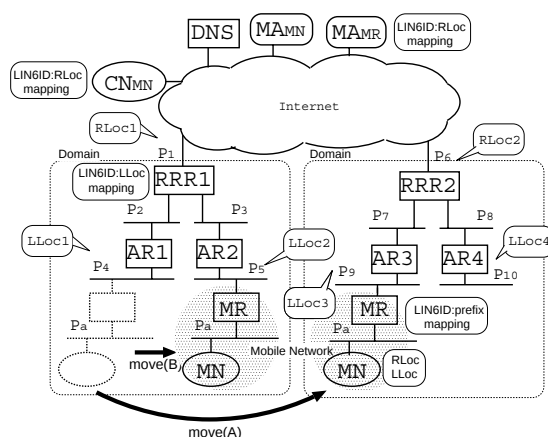


図 4 LIN6-NEMO のネットワーク構成

Fig. 4 Network structure of LIN6-NEMO.

vertisement (RA) で RLoc を通知する機能を持つ。また、MR は拡張 RA により RLoc と自身の LIN6 ID (MRID) を下位サブネットに接続したノードに通知する機能を持つ。ただし、AR による Prefix Delegation は、あくまでも MONET の最上位ルータにネットワークプレフィックスを割り当てるだけであり、MONET 内の全ネットワークプレフィックスを再設定するわけではない。

LIN6-NEMO では、ネストした MONET の最上位ルータを root-MR、root-MR が移動先ネットワークで動的に割り当てられるネットワークプレフィックスを root MRLoc と呼ぶ。root-MR は MONET 内に接続している全ノードのネットワークプレフィックスと LIN6 ID の対応関係を Mapping Table に保持する。しかし、本論文では説明の簡略化のためネストしたケースを扱わないので、これ以降の MR は root-MR と同等の働きをする。

3.5 想定される運用例

本節では LIN6-NEMO の想定される運用例を示す。LIN6-NEMO を東海道新幹線に適用した場合、東京から新大阪までの領域が RRR で管理される Domain であり、線路に沿って AR が多数設置されている。そして、新幹線の車両内に敷設された LAN が MONET となり、root-MR によって AR に接続している。乗客の持つ PDA などの端末機器が MN または MR である。このように規定した場合、新幹線が東京-新大阪間を移動する際はすべて Domain 内移動である。

root-MR は MONET 内の全ノードを管理するので、新幹線の最大乗客数がおよそ 1,300 人で 1 人 1 台程度の端末機器を使用すると想定すると、root-MR は 1,300 ノードを管理する必要がある。また、RRR は

Domain 内の全 MONET を管理するので、数万のノードを管理可能でなければならない。

4. 移動時の処理

本章では、説明の簡略化のためネストしないネットワーク構成を例に LIN6-NEMO の移動時の処理を説明する。移動時の処理には、なりすましなどを防止するために認証機構が必須である。LIN6-NEMO では RRR, MR, CN への位置登録処理に LIN6 および HLIN6 の位置登録における認証機構¹⁴⁾を利用する。この認証機構は Mobile IPv6 の Return Rountability と同様にクッキーの交換を利用しており、Return Rountability よりも頑強である。また、MA への位置登録処理には IPsec を用いる。

4.1 MN が MONET 内に移動した場合

MN が MONET 内に移動した場合、MN は接続した MR の RA から現在の RLoc, MRID とネットワークプレフィックスを獲得し、ネットワークプレフィックスを MR へ、MRID を RRR と MA_{MN} へ登録するための位置登録メッセージを送信する。また、MN は通信中の全ノード (CN_{MN}) に位置登録メッセージで RLoc を通知する。

4.2 MONET の Domain 内移動

MONET の MR が図 4 の move (B) のように Domain 内移動して AR2 に接続した場合、MR は接続した AR2 の RA から現在の RLoc と LLoc を獲得する。そして、MR は現在のネットワークプレフィックスとして LLoc2 を RRR1 へ登録するための位置登録メッセージを送信する。

4.3 MONET の Domain 間移動

MONET の MR が図 4 の move (A) のように Domain 間移動して AR3 に接続した場合、MR は接続した AR3 の RA から現在の RLoc と LLoc を獲得し、現在の MR のネットワークプレフィックスとして LLoc3 を RRR2 へ登録するための位置登録メッセージを送信する。次に、MR は RLoc を MA_{MR} に登録するための位置登録メッセージを送信する。上記のような位置登録処理が行われた後、MR は下位サブネットへ RLoc を通知する。MONET 内の MN はこれを受信することにより、現在接続している位置の RLoc を知ることができる。通信中の全 MN は位置登録メッセージで RLoc を CN_{MN} へ通知する。

5. 通信手順

本章では、説明の簡略化のためネストしないネットワーク構成である図 4 (move (B) の後) を例に、図 5 を用いて LIN6-NEMO の通信手順を説明する。

5.1 CN_{MN} から MN への通信手順

CN_{MN} は DNS に問い合わせた MA_{MN} の位置情報を得た後 (図 5 (1)), MA_{MN} に問い合わせた MRID を取得し (図 5 (2)), さらに MA_{MR} に問い合わせた RLoc1 を得る (図 5 (3))。 CN_{MN} はその RLoc1 を宛先 IP アドレスの Locator とし、データパケットを送信する。 CN_{MN} から MN へのデータパケットは、RRR1 までは LIN6 の通常の通信手順に従い配送される (図 5 (4))。 MN へのデータパケットが RRR1 に到着すると、RRR1 では宛先 IP アドレスの下位 64 bit から MN の LIN6 ID を取得する。そして RRR1 は

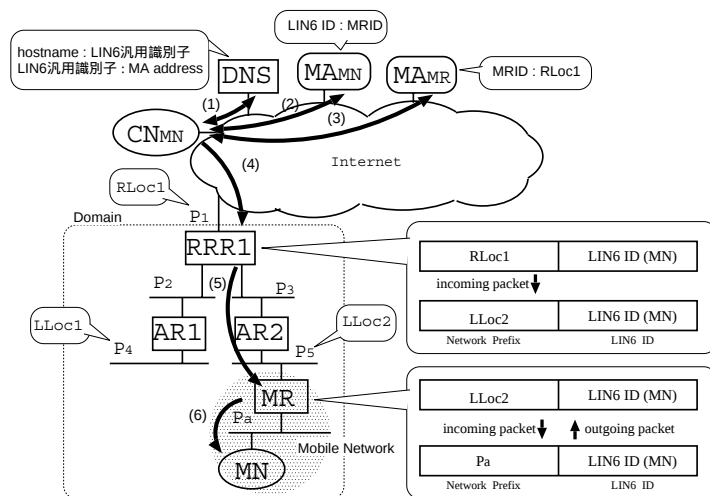


図 5 LIN6-NEMO の通信手順

Fig. 5 Communication procedure of LIN6-NEMO.

MN の LIN6 ID をもとに現在 MN が接続している MONET のサブネットのネットワークプレフィックスである $LLoc2(P_5)$ を得る．そして、宛先 IP アドレスの上位 64 bit を $LLoc2$ に書き換え、データパケットを転送する (図 5 (5))．MN へのデータパケットが MR に到着すると、MR では宛先 IP アドレスの下位 64 bit から MN の LIN6 ID を取得し、これをもとに MN が接続しているネットワークプレフィックスである P_a を得る．そして、宛先アドレスの上位 64 bit を $LLoc2$ から P_a に書き換え、MN に向けて再び転送する (図 5 (6))．

5.2 MN から CN_{MN} への通信手順

MN は MA へ問合せを行い、 CN_{MN} の IP アドレスを宛先 IP アドレスの Locator とし、データパケットを送信する．送信されたデータパケットの送信元 IP アドレスの Locator には P_a が入っている． CN_{MN} へのデータパケットが MR へ到着すると、MR では IP ヘッダから MN の LIN6 ID を取得する．MR は MN の LIN6 ID が Mapping Table 内にあれば、IP アドレスの P_a を $LLoc2$ に書き換え、データパケットを CN_{MN} へ転送する．

6. NEMO Basic Support Protocol との比較

本章では、NEMO Basic Support Protocol と LIN6-NEMO の定性的な性能の比較、検討を行う．

6.1 パケットの配送経路

NEMO Basic Support Protocol における通信中のノード間のパケット配送経路は、複数の HA を経由し、Pin-ball routing 問題が発生する．また、MONET がネストした場合、Pin-ball routing 問題はさらに深刻になる．これに対し本提案プロトコルである LIN6-NEMO では、通信開始時に MONET 内でネストしている MR や MN の数だけ MA に位置情報を要求する必要があるだけで、位置情報を獲得した後は MONET のネストの段数にかかわらず、つねに最適経路で MONET 内のノードと通信をすることが可能である．

6.2 ヘッダオーバーヘッド

NEMO Basic Support Protocol において経路最適化処理がなされていない場合、HA を経由するのでトンネリングにより 40 byte のヘッダが付加される．MONET がネストしている場合、HA の数 n だけトンネリングによって 40 byte の IPv6 ヘッダが付加されることになるため、全体で $40n$ byte のヘッダオーバーヘッドになる．これに対して LIN6-NEMO では、ト

ンネリングせず RRR と root-MR でネットワークプレフィックスの書き換え処理を行うため、つねにヘッダオーバーヘッドなしに通信することが可能である．

6.3 耐故障性

NEMO Basic Support Protocol では各 MN、MR の HA は複数設置可能であるが、ホームリンクのみにしか設置することができない．したがって、ホームリンクに障害が発生したときには、各 HA が管理する移動ノードと MONET 内のノードは通信不可能になる．LIN6-NEMO では、LIN6 と同様、MA の設置可能数や設置場所に制限はないので、1 つの MA に障害が起きたり、1 つの MA が接続しているサブネットに障害が発生したりしても、別の MA を利用して通信することが可能である．

6.4 既存ノードやルータへの変更点

NEMO Basic Support Protocol は、MN、MR、HA の機能によって実現されるので、このプロトコルと無関係の、通常のノードやルータに対して変更をする必要はない．これに対し、LIN6-NEMO では、root-MR が移動先のサブネットでネットワークプレフィックスを取得するために Prefix Delegation¹³⁾ を利用するため、MONET が接続する AR は、Prefix Delegation 機能を持たなくてはならない．AR が Prefix Delegation 機能を持たない場合は、通信不可能である．また、LIN6-NEMO では Domain を定義し、Domain 内の移動を外部ネットワークに隠蔽するために RRR を設置する必要がある．

ここまで行った、NEMO Basic Support Protocol と本提案プロトコルの定性的な性能の比較の結果を表 1 に示す．

6.5 位置登録メッセージ数

表 2 は位置登録メッセージ数を比較し、まとめたものである．表 2 における“MN 移動”、“MONET 移動”はそれぞれ、MN が MONET 内に移動した場合、MONET が移動した場合の意味である．また、“外”、“内”は制御メッセージが送出される領域が Domain 外か Domain 内かを示す．“ CN_{MN} ”は MN と通信中の CN の台数であり、“MN”は MONET 内の MN の台数である．表 2 より、LIN6-NEMO は NEMO Basic Support Protocol よりも位置登録メッセージ数が多いことが分かる．しかし、MONET が Domain 内移動する場合は、近距離に存在する RRR に対してのみ位置登録メッセージを送信するだけで済むため、処理

Prefix Delegation を規定している RFC3633 は現在のところ Proposed Standard である．

表 1 NEMO Basic Support Protocol と LIN6-NEMO の定性的性能の比較
Table 1 Comparison of NEMO Basic Support Protocol and LIN6-NEMO.

	NEMO Basic Support Protocol	LIN6-NEMO
基本となる ホストモビリティプロトコル	Mobile IPv6	LIN6
パケットの配送経路	ネスト数だけ HA を経由	つねに最適経路
ヘッダオーバーヘッド	40 バイト × 経由する HA	なし
位置管理エージェントの 設置点と設置可能数	ホームリンクのみ可能 1 つ以上	どこでも可能 1 つ以上
既存ノードや ルータへの変更点	なし	AR は Prefix Delegation 機能を持つ RRR を設置

表 2 位置登録メッセージ数の比較 (単位: 個)
Table 2 Comparison of number of location registration messages.

	NEMO Basic Support Protocol		LIN6-NEMO	
	Binding Cache 有	Binding Cache 無	外	内
MN 移動	$1 + CN_{MN}$	1	外	$1 + CN_{MN}$
			内	2
MONET 移動	1		Domain 内移動	Domain 間移動
			外	0
			内	$1 + MN (1 + CN_{MN})$
				1

時間は短く、外部ネットワーク上に位置登録メッセージは流出しない。これに対し、NEMO Basic Support Protocol は比較的遠距離に存在する HA に対して位置登録メッセージを送信するため、処理に時間がかかり、外部ネットワーク上に位置登録メッセージが流出する。

7. 実装と評価

7.1 実 装

LIN6-NEMO は、KAME プロジェクトから配布されている IPv6 スタックと本研究室で開発されている LIN6 を利用し、NetBSD 1.6.1 Release 上に実装を行った。LIN6-NEMO は LIN6 における kernel 部分および Mapping Resolver Daemon と Mapping Agent Daemon という userland の 2 つのデーモンへ LIN6-NEMO に必要な機能を追加し、LIN6 を拡張した形で実現した。ただし、認証機構は未実装である。

図 6 に実験ネットワークを示す。実験では、有線環境で Ethernet ケーブルを抜き差しすることにより移動をエミュレートし、様々な移動パターンでの LIN6-NEMO の動作を確認した。

7.2 移動時の処理時間

移動先のネットワーク構成により多少異なるが、MN や MONET は移動した場合、MR, RRR, MA, CN のうち必要なものに対してのみ位置登録処理を行う。この位置登録処理は通常の通信にはない処理であるため、LIN6-NEMO におけるオーバーヘッドになる。

そこで、図 6 のネットワークにおいて、

- (1) MN1 が MR1 を root-MR とする MONET 内に移動した場合、
 - (2) MONET が Domain 内移動をした場合、
 - (3) MONET が Domain 間移動をした場合、
- の 3 つの場合の位置登録処理時間を tcpdump と ping6 コマンドを利用して測定した。

LIN6-NEMO では MR への位置登録処理が終了する以前に、RRR, MA, CN への位置登録処理を開始することが可能であるため、実際の位置登録処理に必要な時間は、マシンでの処理時間と MN から最も遠い位置登録先 (RRR, MA, CN のどれか) と MN の間の RTT (Round Trip Time) の合計である。(1) のとき、MN から最も遠いノードが MA であると仮定すると、移動処理時間は、測定結果から得られた MA での位置登録処理時間 $687 \mu\text{sec}$ + MN-MA 間の RTT である。(2) のときの移動処理時間は、測定結果から得られた RRR での処理時間 $796 \mu\text{sec}$ + MN-RRR 間の RTT である。(3) のとき、MN から最も遠いノードが MA であると仮定すると、移動処理時間は、測定結果から得られた MA での位置登録処理時間 $934 \mu\text{sec}$ + MN-MA 間の RTT である。

実際の運用環境では、RTT に比べると本測定で得られた位置登録処理時間は非常に小さな値であるため、移動処理時間全体と RTT はほぼ等しくなる。つまり、移動処理時間は MN から最も遠い位置登録先との RTT に依存するといえる。

しかし (3) の場合は、4.3 節で述べたように、MR が下位サブネットに RLoc を通知する処理も移動時に

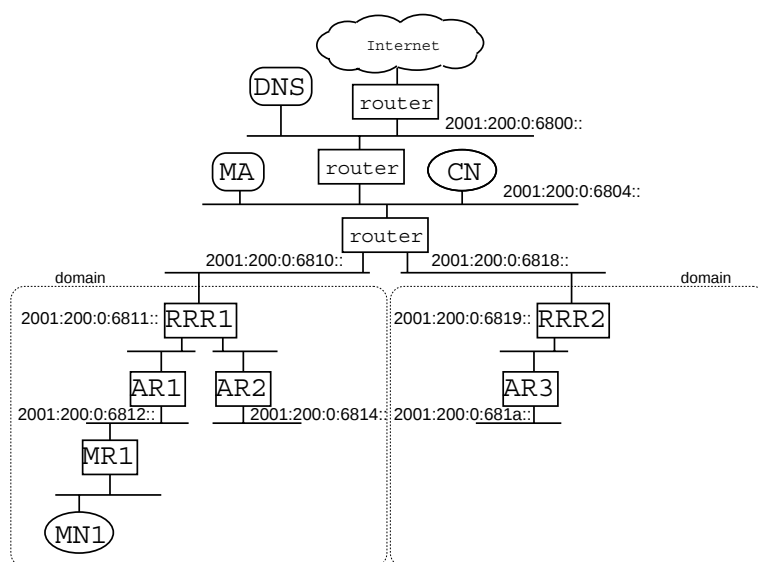


図 6 実験ネットワーク

Fig. 6 Experimental network.

必要な処理であるため、移動処理時間は RTT と RLoc の通知処理時間に依存する。ただし、Domain を適切な大きさに定義することにより、Domain 間移動がなるべく発生しないようにすることは可能である。(2) のような Domain 内移動の場合は、MONET 内に通信中の多数の MN や MR が接続している場合でも、RRR への位置登録処理のみだけでよく、外部ネットワークに位置登録メッセージが流出することはない。したがって、LIN6-NEMO では Domain 間移動の発生頻度を低く抑えることが重要である。

NEMO Basic Support Protocol の場合、移動処理時間は MR-HA 間の RTT とほぼ等しくなる。これは、LIN6-NEMO における移動処理時間とほぼ同等である。

7.3 CN_{MN} から MONET 内の MN への通信におけるオーバーヘッド

5.1 節で述べたように、 CN_{MN1} は通信開始時に MA_{MN1} 、 MA_{MR1} と複数の MA へ反復的に問合せを行い、通信相手である MN の位置情報として、MN が属する MONET の RLoc を取得する。通常の通信では、DNS への問合せ後、すぐに通信を開始することができるので、MA への反復的な位置情報要求処理は LIN6-NEMO におけるオーバーヘッドになる。

しかし、通信開始時のみ MA への位置情報要求処理が必要であり、それ以降はすでに獲得した位置情報を利用して通信を行うことが可能であるため、このオーバーヘッドがない。通信開始時の位置情報要求処理に要

する時間自体は、(MA での処理時間 + CN_{MN} - MA 間の RTT) $\times$ MA への問合せ回数である。しかし、これは通信開始時のみ必要な処理時間である。

また、LIN6-NEMO では通信時に RRR と root-MR においてアドレスの書き換え処理を行う。この処理も通常の通信にはないため、LIN6-NEMO におけるオーバーヘッドになる。そこで、root-MR の Mapping Table に登録されているエントリの数を 1, 10, 100, 1,000, 10,000, 100,000, 250,000 個に変化させ、各場合における宛先アドレスの書き換え処理時間を測定した。

測定では、指定数のエントリを追加した後に CN_{MN1} が MN1 へ ping6 コマンドを実行する。宛先アドレスの書き換え処理時間を測定する場合は、その ICMP echo request が root-MR で書き換えられる際の処理時間を、送信元アドレスの書き換え処理時間を測定する場合は、その ICMP echo reply が root-MR で書き換えられる際の処理時間を測定した。最悪値を測定するため、エントリを Mapping Table に登録する際には、MN1 のエントリが Mapping Table のハッシュチェーンの末尾になるように、MN1 のエントリを最後に追加した。処理時間の測定には microtime() 関数を用いている。

図 7 は測定結果をグラフにしたものである。図 7 の縦軸は宛先アドレスの書き換え処理時間 (μsec) であり、横軸は root-MR の Mapping Table のエントリ数 (個) である。横軸は対数表示である。

図 7 が示すように、エントリ数が 100 程度までの

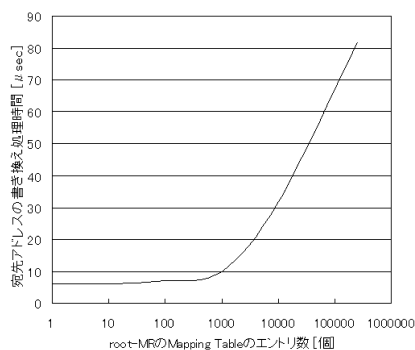


図 7 root-MR での宛先アドレス書き換え処理時間

Fig. 7 Overhead of overwriting destination addresses on root-MR.

アドレスの書き換え処理時間は $6 \mu\text{sec}$ で、ほぼ一定であることが分かる。これらは、通信ノード間の RTT と比較しても十分小さい値であり、処理オーバーヘッドとしては無視できる値である。

しかし、エントリ数が 1,000 に近くなるあたりから処理時間は次第に増加している。これは、Mapping Table がハッシュテーブルで管理されているためであると考えられる。ハッシュテーブルではハッシュ値が衝突すると、同一のハッシュ値を持つエントリはリストとして管理され、検索の際にはそのリストを線形に探索する。本実装では Mapping Table のハッシュテーブルのサイズは 64 であるので、100 程度まではハッシュ値の衝突は少なく、エントリの検索は迅速に行われる。しかし、それ以降はハッシュ値の衝突が頻繁に発生するため、エントリのリストが長くなり、線形探索によるエントリの検索に時間がかかるのである。

本測定では MN1 のエントリがハッシュチェーンの末尾になるよう登録を行ったので、得られた測定結果はあるエントリ数の場合における最大処理時間である。実際には、エントリがハッシュチェーンのどのあたりに登録されているかはランダムであるので、全体で平均すると本測定結果の半分程度の処理時間になると予測される。ただし、 CN_{MN1} から MN1 の通信の場合、アドレスの書き換え処理は root-MR と RRR で行われるため、2 倍の時間が必要となる。

実際の通信では通信ノード間の RTT は数 msec から数十 msec のオーダーになると考えられるため、アドレス書き換え処理に必要な処理時間は相対的に無視できる程度の値になると考えられる。また、3.5 節のように現実の利用形態を考えると、1 つの MONET 内に 10,000 を超えるノードが接続したり、Domain 内に 100,000 を超えるノードが接続する可能性は稀であると考えられる。MONET は最大でも数千のオーダー、

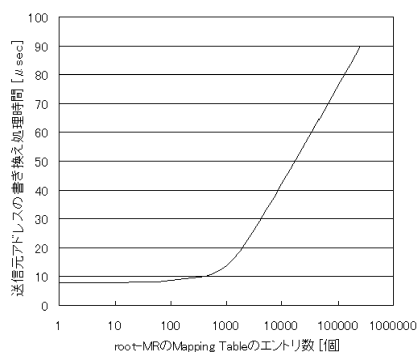


図 8 root-MR での送信元アドレス書き換え処理時間

Fig. 8 Overhead of overwriting source addresses on root-MR.

Domain の場合は最大で数万のオーダーであると予想される。このとき、RTT に対するアドレス書き換え処理時間は非常に小さな値であるといえるため、アドレス書き換え処理におけるオーバーヘッドは無視できると考えられる。

7.4 MONET 内の MN から CN_{MN} への通信におけるオーバーヘッド

MONET 内の MN から CN_{MN} への通信におけるオーバーヘッドは、7.3 節で述べた CN_{MN} から MONET 内の MN への通信におけるオーバーヘッドとほぼ同等である。異なるのは、MONET 内の MN から CN への通信では RRR におけるアドレスの書き換え処理がなく、root-MR で送信元アドレスの書き換え処理を行っている点である。

そこで、7.3 節と同様の実験方法で root-MR における送信元アドレスの書き換え処理時間を測定した。図 8 は測定結果をグラフにしたものである。図 8 の縦軸は送信元アドレスの書き換え処理時間 (μsec) であり、横軸は root-MR の Mapping Table のエントリ数 (個) である。横軸は対数表示である。

図 8 は、多少値は大きいものの、図 7 とほぼ同様の結果であるといえる。送信元アドレスの書き換え処理時間のほうが宛先アドレス書き換え処理時間より長い理由は、送信元アドレスの書き換え処理では Mapping Table の検索を 2 回行うため、ハッシュリストの線形探索によるエントリの検索で余計に時間がかかりやすいためである。送信元アドレス書き換え処理時間も、宛先アドレス書き換え処理時間と同様にそのオーバーヘッドを無視することができるといえる。

7.5 Domain の大きさに関する考察

LIN6-NEMO では NEMO Basic Support Protocol にはない Domain を定義する必要がある。Domain の大きさと LIN6-NEMO における処理のオーバーヘッ

ドの関係を 3.5 節の例を用いて考察する。

3.5 節の例において、東京-新大阪間が C 個の無線セルで構成され、 D 個の Domain に分割されていると仮定する。そこを 1 つの MONET に相当する新幹線が移動する場合、Domain 間移動は $D-1$ 回、Domain 内移動は $C-D$ 回発生する。このとき MONET 内の MN の台数を " MN "、MN と通信中の CN の台数を " CN_{MN} " とすると、表 2 より、Domain 内には $C-1$ 個、Domain 外には $(D-1)(1+MN(1+C_{N_{MN}}))$ 個の位置登録メッセージが送信される。つまり、MONET である新幹線が東京から新大阪へ移動する場合に、最大 $92 \times (D-1)(1+MN(1+C_{N_{MN}}))$ byte の制御メッセージが送信され、外部ネットワークに負荷をかけてしまう。よって、外部ネットワークへの負荷を抑えるためには、Domain を大きく定義する必要がある。

また、 M 個の MONET が無線セル数 C の範囲に均等に分散していると仮定すると、1 Domain 内に存在する MONET 数 m は $m = M/D$ である。 D が大きければ、 m は小さくなるので RRR が保持すべき MN の情報は少なくなり、アドレス書き換え処理時間が短くなる。また、 D が小さければ、 m は大きくなるので RRR におけるアドレス書き換え処理時間が長くなる。つまり、RRR におけるアドレス書き換え処理時間を短くするためには、Domain を小さく定義する必要がある。

したがって、Domain が小さい場合は外部ネットワークへの負荷が高くなるが RRR におけるアドレス書き換え処理時間が短くなり、Domain が大きい場合は外部ネットワークへの負荷が低くなるが RRR におけるアドレス書き換え処理時間が長くなるという、トレードオフの関係があるといえる。

8. ま と め

本論文では、データ配送時につねに最適経路でヘッダオーバーヘッドがなく、耐故障性にも優れている LIN6 に基づき、移動ネットワークに移動透過性を提供するネットワークモビリティプロトコルである LIN6-NEMO の提案・設計・実装を行った。

LIN6-NEMO では、MA の位置管理機能と MN の位置登録処理機能を MR が持つことで、移動ネットワークに移動透過性を提供した。また、Domain とい

う概念を導入し、RRR が Domain 内部における移動を隠蔽することにより、局所的な移動の際に位置登録などの制御パケットが外部ネットワークに大量に流出するのを防止した。

そして、本提案プロトコルを NetBSD 1.6.1 Release 上で実装し、実験ネットワークで行った測定結果をもとに定性的性能と定量的性能について評価した。定性的な性能では、NEMO Basic Support Protocol と比較し、LIN6-NEMO は通信経路・ヘッダオーバーヘッド・耐故障性の点において優れていることが分かった。また、Domain 内移動の場合は外部ネットワークに位置登録メッセージを流出させずに移動可能であるが、Domain 間移動の場合は多数の位置登録メッセージが必要になることが分かった。

定量的な性能では、データ送信時のオーバーヘッドは無視できることが分かった。したがって、本論文で提案したプロトコルは、効率的に移動ネットワークに移動透過性を提供することが可能なプロトコルであることを確認できた。

今後の課題は、認証機構の実装とマルチホームへの対応である。

参 考 文 献

- 1) Jhonson, D., Perkins, C. and Arkko, J.: Mobility Support in IPv6, Internet RFC3775, *IETF* (2004).
- 2) Kunishi, M., Ishiyama, M., Uehara, K., Esaki, H. and Teraoka, F.: LIN6: A New Approach to Mobility Support in IPv6, *Proc. 3rd International Symposium on Wireless Personal Multimedia Communications* (2000).
- 3) Devarapalli, V., Wakikawa, R., Petrescu, A. and Thubert, P.: Network Mobility (NEMO) Basic Support Protocol, Internet Draft, *IETF* (2004).
- 4) Cho, H., Paik, E.K., Choi, Y.: R-BU: Recursive Binding Update for Route Optimization in Nested Mobile Networks, *IEEE VTC2003 Fall, IP Mobility* (2003).
- 5) Wakikawa, R., Koshiba, S., Uehara, K. and Murai, J.: ORC: Optimized Route Cache Management Protocol for Network Mobility, *Proc. IEEE 10th International Conference on Telecommunication (ICT)* (2003).
- 6) Lee, K., Park, J. and Kim, H.: Route Optimization for Mobile Nodes in Mobile Network based on Prefix Delegation, *IEEE VTC2003 Fall, IP Mobility* (2003).
- 7) Soliman, H., Catelluccia, C., Malki, K.E. and Bellier, L.: Hierarchical Mobile IPv6 mobility

セルは、IP サブネットに相当するものとする。

位置登録メッセージの大きさは 92 byte である (IPv6 Base Header (40 byte) + UDP Header (8 byte) + Registration Message (44 byte) = 92 byte)。

management (HMIPv6), Internet Draft, *IETF* (2004).

- 8) Takagi, Y., Ohnishi, H., Sakitani, K., Baba, K. and Shimojo, S.: Route Optimization Methods for Network Mobility with Mobile IPv6, *IEICE Trans. Commun.*, Vol.E87-B, No.3, pp.480-489 (2004).
- 9) Oiwa, T., Kunishi, M., Ishiyama, M., Kohno, M. and Teraoka, F.: A Network Mobility Protocol based on LIN6, *IEEE VTC2003 Fall, IP Mobility* (2003).
- 10) 原田友紀子, 國司光宣, 寺岡文男: LIN6 のマイクロモビリティサポート, 情報処理学会論文誌, Vol.43, No.12, pp.3878-3888 (2002).
- 11) Narten, T., Nordmark, E. and Simpson, W.: Neighbor Discovery for IP Version 6 (IPv6), Internet RFC2461, *IETF* (1998).
- 12) Droms, R., Bound, J., Volz, B., Lemon, T., Perkins, C. and Carney, M.: Dynamic Host Configuration Protocol for IPv6 (DHCPv6), Internet RFC3315, *IETF* (2003).
- 13) Troan, O. and Droms, R.: IPv6 Prefix Options for Dynamic Host Configuration Protocol (DHCP) version 6, Internet RFC3633, *IETF* (2003).
- 14) 田中康之, 國司光宣, 石山政浩, 寺岡文男: LIN6 および HLIN6 における認証機構, 電子情報通信学会論文誌, Vol.J87-D-I, No.5, pp.497-507 (2004).

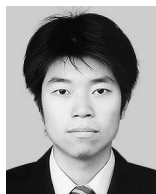
(平成 16 年 4 月 5 日受付)

(平成 16 年 10 月 4 日採録)



坂野あゆみ

1981 年生. 2004 年慶應義塾大学理工学部情報工学科卒業. 現在, 同大学大学院理工学研究科前期博士課程に在学中. ネットワークモビリティを中心とした移動体通信サポートに興味を持ち, 主な研究テーマとしている.



大岩 拓馬

1979 年生. 2003 年慶應義塾大学理工学部管理工学科卒業. 現在, 同大学大学院理工学研究科前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に興味を持ち, LIN6 に基づくネットワークモビリティプロトコルの研究に携わる.



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. 2004 年 9 月同研究科後期博士課程修了. 同年 10 月メディアエクスチェンジ株式会社入社. ネットワーク層での移動体通信サポートを主なテーマとし, LIN6 の基本プロトコル設計, 実装に携わる. 現在は IX のオペレーションについて勉強しながら研究テーマについて熟考中.



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. 特に移動透過性を保証するプロトコル 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), 『Wireless IP and Building the Mobile Internet』(共著, Artech House Publishers). 監訳に『詳細 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, 電子通信学会各会員.