

## 移動体通信プロトコル LIN6 における後方互換性拡張の方式

石山 政浩<sup>†</sup> 國司 光宣<sup>††</sup> 河野 通宗<sup>†††</sup> 寺岡 文男<sup>††††</sup>

<sup>†</sup> (株) 東芝 研究開発センター 通信プラットフォームラボラトリー

<sup>††</sup> 慶應義塾大学 大学院 理工学研究科

<sup>†††</sup> (株) ソニーコンピュータサイエンス研究所

<sup>††††</sup> 慶應義塾大学理工学部

E-mail: <sup>†</sup>masahiro@isl.rdc.toshiba.co.jp, <sup>††</sup>kunishi@tokoro-lab.org, <sup>†††</sup>mkohno@csl.sony.co.jp,

<sup>††††</sup>tera@tera.ics.keio.ac.jp

あらまし 本稿では、移動体通信プロトコル LIN6 における後方互換性拡張の方式を提案する。従来の LIN6 では、既存の IPv6 ノードとの通信の際には通常の IPv6 ノードとしてのみしか通信できず、移動透過性保証を提供できなかった。本稿では、LIN6 の位置管理エージェントである MA を拡張することにより、既存の IPv6 ノードとの移動透過性を提供する方式について検討する。

キーワード 移動透過性保証プロトコル, LIN6, IPv6

## Enhancing the LIN6 Protocol to Provide Mobility Support for Traditional IPv6 Nodes

Masahiro ISHIYAMA<sup>†</sup>, Mitsunobu KUNISHI<sup>††</sup>, Michimune KOHNO<sup>†††</sup>, and Fumio

TERAOKA<sup>††††</sup>

<sup>†</sup> Communication Platform Laboratory, R&D Center, Toshiba Corporation.

<sup>††</sup> Graduate School of Science and Technology, Keio University.

<sup>†††</sup> Sony Computer Science Laboratories, Inc.

<sup>††††</sup> Faculty of Science and Technology, Keio University.

E-mail: <sup>†</sup>masahiro@isl.rdc.toshiba.co.jp, <sup>††</sup>kunishi@tokoro-lab.org, <sup>†††</sup>mkohno@csl.sony.co.jp,

<sup>††††</sup>tera@tera.ics.keio.ac.jp

**Abstract** In this paper we propose a method to enhance the LIN6 protocol to provide mobility support for traditional IPv6 nodes. The current LIN6 protocol does not provide mobility when a LIN6 node communicates with standard IPv6 nodes. We discuss a method to support mobility for IPv6 nodes by extending a function of the Mapping Agent—the entity that manages location of mobile nodes in LIN6.

**Key words** Mobility Protocol, LIN6, IPv6

### 1. はじめに

携帯端末の性能の向上と、移動通信機器の普及に伴い、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、加えて第三世代携帯電話のような高速かつ広域の無線通信のサービスも始まっていることから、移動透過性を保証する通信プロトコルへの期待が高まっている。また現在の IPv4 の限界を無くす、広大なアドレス空間を持つ

IPv6 に対する期待も大きい。

現在提案されている移動透過性保証プロトコルの一つに Location Independent Networking for IPv6 (LIN6) [4] がある。LIN6 は、LIN6ID と呼ばれる単一の ID 空間における識別子を利用して移動透過性保証を提供するプロトコルである。LIN6 は現在 IETF で標準化が行なわれている Mobile IPv6 [5] が持ついくつかの問題点を解決しているが、一方で LIN6 が実装されていない既存ノードとの通信の際には移動透過性が保証できないという問題があった。加えて、移動ノードの LIN6ID に関連する Domain Name System (DNS) サーバに対して、LIN6

で使用する新しいレコードを理解するように変更を加える必要があり、LIN6 の導入を難しくしていた。

本稿では、LIN6 における位置管理エージェントである Mapping Agent (MA) の機能を拡張することにより、既存の LIN6 を理解しないノードとの通信においても移動透過性保証を提供可能にし、かつ、DNS サーバに対して変更を加えなくとも LIN6 を導入可能とする方式を提案する。

## 2. 現在の LIN6 の通信方式の概略

本章では、LIN6 の通信方式について概観する。詳しい概念の説明については [4] を参照されたい。

LIN6 は、現在の移動透過性が困難となっている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の再考を行なって得られた概念である LINA を基底に構築された IPv6 上の移動体プロトコルである。LIN6 では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。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により移動透過性を保証する。

### 2.1 縮退アドレスモデル

LINA では縮退アドレスモデルと呼ばれるアドレスモデルを導入している。縮退アドレスモデルでは、ノード識別子を位置指示子の中に縮退させるという構造を取る。ノード識別子が縮退された位置指示子を縮退アドレスと呼ぶ。縮退アドレスモデルにより、概念的に識別子を扱う層 (識別副層) と位置指示子を扱う層 (配送副層) の 2 層に分離されたヘッダを 1 つのヘッダに統合することが可能となる。また、ネットワーク層より上位層で用いられる識別子は汎用識別子と呼ばれる。汎用識別子は、固有位置指示子とノード識別子とを縮退したものである。固有位置指示子はあらかじめ定められた固定値であり、位置に依存しない。すなわち、移動により変化しない。また、縮退アドレスあるいは汎用識別子からノード識別子を得る操作を抽出と呼ぶ。

LIN6 では、この縮退アドレスモデルを以下のように実現している。まず、LIN6 では 64bit のノード識別子を導入する。これを LIN6ID と呼ぶ。

現在、IPv6 の通信で主に使用されている Aggregatable Global Unicast Address (AGUA) [2] は、上位 64bit がネットワークプレフィクス、下位 64bit がインターフェース識別子という構造である (図 1 (a))。LIN6 のアドレスモデルは、この構造を利用し、アドレス構造の 128bit 全体を位置指示子とし、この位置指示子の下位 64bit にノード識別子である LIN6ID を縮退させる。この 128bit の位置指示子を LIN6 アドレスと呼ぶ (図 1 (b))。

図 1 に示すように、LIN6 アドレスは従来の AGUA 形式と互換性を保ちながら、LINA における縮退アドレスと同様に位置指示子とノード識別子という分離された 2 つの情報を保持したアドレスとなる。

LIN6 での汎用識別子の導出は次のようになる。まず、LINA

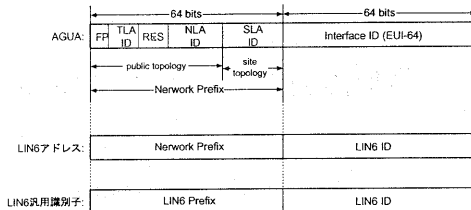


図 1 AGUA, LIN6 アドレスおよび LIN6 汎用識別子: 既存の IPv6 アドレスである AGUA では上位 64bit はサブネットの位置を示し、下位 64bit はインターフェース識別子である。LIN6 アドレスでは上位 64bit は AGUA と同様にノードの現在位置のサブネットの位置を示すが、下位 64bit はノード識別子である LIN6ID である。LIN6 汎用識別子は、上位 64bit は位置に依存しない LIN6 プレフィクスである

における固有位置指示子に対応する上位 64bit の固定値を導入する。これを LIN6 プレフィクスと呼ぶ。そして、この固有プレフィクスに LIN6ID を縮退させる。この縮退されたアドレスが LIN6 における汎用識別子となり、これを LIN6 汎用識別子と呼ぶ (図 1 (c))。

### 2.2 LIN6ID と LIN6 アドレスとの対応づけ

LIN6 では、移動ノードと通信する際に、LIN6ID とそのノードの現在の位置指示子である LIN6 アドレスとの対応関係を得る必要がある。この対応関係を Mapping と呼ぶ。LIN6 では、Mapping を管理する Mapping Agent (MA) と呼ばれるノードを導入する。ノードは移動した際には現在の位置指示子を MA に通知する。MA は LIN6ID と位置指示子の関係を保持し、通信ノードから要求があった場合、指定された LIN6ID に対する位置指示子を知覚する役割を担う。あるノード A の Mapping を管理する MA を、ノード A の Designated Mapping Agent と表現する。あるノードの Designated MA は複数存在しても良い。

### 2.3 LIN6 アドレスと通常の IPv6 アドレスとの識別

LIN6 アドレスは、AGUA の形式と構造上同じであるため、アドレス構造だけでは与えられたアドレスが LIN6 アドレスかどうか識別することはできない。しかし LIN6 では、受信時にアドレスから汎用識別子に変換するかどうかを判断しなければならず、このため、アドレスからそれが LIN6 アドレスかどうか識別する機構が必要となる。

通常、AGUA のインターフェイス ID 部は EUI-64 [3] の構造に従っている。EUI-64 の構造は、IEEE から割当てられる先頭 24bit の Organizationally Unique Identifier (OUI) と OUI の管理者が割り付ける 40bit からなる。この構造を利用し、64bit ノード識別子の先頭 24bit には、LIN6 固有の OUI を付加することとする。これにより、アドレスから抽出したノード識別子がこの特定の OUI から始まるかどうかを確認することにより、LIN6 アドレスと通常の IPv6 アドレスを識別できる。

### 2.4 LIN6 の通信モデル

LIN6 の通信モデルを図 2 に示す。

送信時に上位層から LIN6 汎用識別子が指定された場合 (図

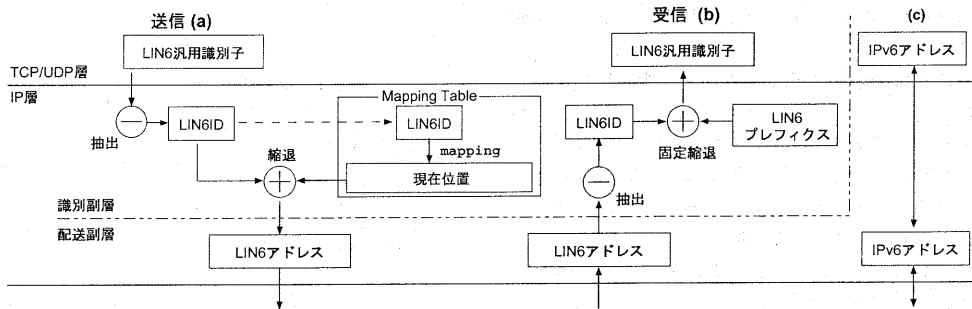


図2 LIN6における通信モデル: LIN6 汎用識別子は送信時にノード識別子を基に得られた現在の位置指示子から LIN6 アドレスに変換される。LIN6 アドレスは受信時に LIN6 汎用識別子に変換される。LIN6 汎用識別子は上位層における識別子として利用され、LIN6 アドレスはパケットの経路制御のために利用される。上位層から直接 IPv6 アドレスを指定された場合には変換操作はおこなわない。

2 (a), まず識別副層で LIN6ID を抽出する。次に MA からこの LIN6ID に対する Mapping を取得して現在の位置指示子を導く。得られた位置指示子に LIN6ID を縮退させ、LIN6 アドレスに変換する。LIN6 アドレスは配送副層に渡され、送信パケットはそのアドレスを基に経路制御される。

受信時 (図 2 (b)) にはデータリンク層からネットワーク層に渡されたパケットは、まず配送副層に到達する。このパケットから得られた LIN6 アドレスは識別副層に渡され LIN6ID が抽出される。そして LIN6 プレフィクスに LIN6ID を縮退させ LIN6 汎用識別子に変換する。得られた LIN6 汎用識別子は上位層に渡される。

アプリケーションから直接位置指示子を指定された場合 (図 2 (c)) には、LIN6 のネットワーク層は従来のネットワーク層と同様の処理を行う。

### 2.5 Mapping Agent の発見

LIN6 では、あるノードの Designated MA の発見に DNS を用いる。LIN6 では、DNS のレコード群に新たに MA のアドレスを表す MA レコードを導入する。LIN6ID に対する MA レコードを DNS に問い合わせることにより、対象の LIN6ID を持つノードの Designated MA のアドレスを得ることができる。この問い合わせの際、DNS サーバには、LIN6ID を 4bit づつ逆順に並べた文字列に、lin6.net を連結した形式で問い合わせる。例えば、LIN6ID として 0001:4afe:dcba:9876 を持つノードの Designated MA のアドレスを得る場合には、6.7.8.9.a.b.c.d.e.f.a.4.1.0.0.lin6.net の MA のレコードを問い合わせれば良い。以後、この LIN6ID を逆順に並べたものを  $rev(LIN6ID)$  と表現する。すなわち、

$$rev(0001:4afe:dcba:9876) = 6.7.8.9.a.b.c.d.e.f.a.4.1.0.0$$

となる。

## 3. 提案方式

### 3.1 Mapping Agent の拡張

従来方式では、MA はすべて通常のホストでも運用可能で

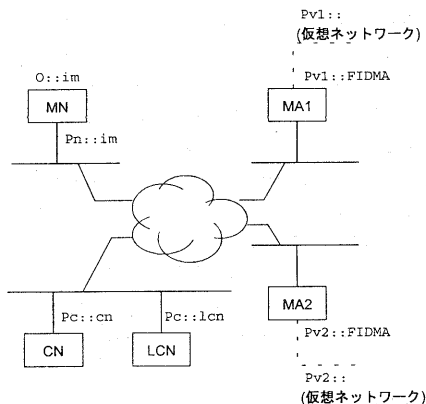


図3 提案方式におけるネットワーク例: MN, CN はそれぞれ LIN6 ノードであり、LCN は LIN6 を理解しない従来の IPv6 ノードである。MA1, MA2 はともに MN の Designated MA である。

あったが、提案方式では MA を拡張し、MA は必ず LIN6 を理解するルータであることとする。MA は必ず一つ以上の仮想ネットワークを持つ。仮想ネットワークとは、ここでは物理的な存在を伴わない論理的に存在するネットワークを指す。また、この仮想ネットワークへの経路はインターネットに広告されているものとする。加えてすべての MA は、仮想ネットワーク上のインターフェイス ID 部があらかじめ定められた固定値となるアドレスを自己のアドレスとして割り付ける。すべての LIN6 ノードはこの固定値をあらかじめ知っているものとする。以後、この固定値を **FIDMA** と表現する。例えば、ある MA が持つ仮想ネットワークのプレフィクスが  $Pv::$  である場合、 $Pv::FIDMA$  はその MA が持つアドレスである。

提案方式を図 3 を例に説明する。MN, CN は LIN6 を理解するノードであり、MN の LIN6ID を  $im$  とする。LCN は LIN6 を理解しない従来の IPv6 ノードである。Pn, Pc, Pv1, Pv2 はそれぞれネットワークプレフィクスを示す。O は LIN6 プレ

フィクスを表す。MA1, MA2 はそれぞれ MN の Designated MA である。

### 3.2 MA レコードを利用しない MA の発見方式

2.5 章で述べたように、従来方式では、ある移動ノードの Designated MA のアドレスを知るためには、LIN6 独自のレコードである MA レコードが必要である。すなわち、MA の位置を管理する DNS サーバには、この新しい MA レコードを理解できるように変更を加える必要があった。これは現在運用されているすべての DNS サーバが MA レコードを理解する必要があるということではなく、一部の DNS サーバのみの変更であるため導入の障壁としては低いといえるが、LIN6 の普及を阻害する一要因となりうることは否めない。

そのため我々は、あるノードの Designated MA のアドレスを既存の AAAA レコード [8] を利用して表現する方式を提案する。AAAA レコードは、現在 IPv6 アドレスを表現するために利用されているレコードである。

提案方式では、移動ノードの AAAA に対して、Designated MA の仮想ネットワークプレフィクスに、自分の LIN6ID を連結した結果を登録する。この例では、mn.lin6.net の AAAA レコードの値は、Pv1::im および Pv2::im となる。以後、移動ノードのこの仮想ネットワーク上のアドレスを後方互換アドレスと呼ぶ。同様に、従来 MA レコードを引くために利用していた rev(LIN6ID).lin6.net には PTR レコード [7] を設定し、そのノードの FQDN を定義しておく。PTR レコードは現在逆引き、すなわちアドレスから FQDN を取得するために利用される既存のレコードである。本例においては、rev(im).lin6.net の PTR レコードには、mn.lin6.net が定義される。

これにより、あるノードが Designated MA のアドレスを取得する手順は次のようになる。

- rev(LIN6ID).lin6.net の PTR レコードを問い合わせ、対象ノードの FQDN を得る
- 対象ノードの FQDN に対応した AAAA を問い合わせる。
- 得られた IP アドレスのうち、下位 64bit を FIDMA にすることによって MA のアドレスが得られる。

図 3 を例にすると、MN の Mapping の登録処理は図 4 のようになる。

- MN はまず自分の LIN6ID から FQDN を取得する (1, 2)。
- 次に、FQDN から AAAA を取得し、下位 64bit を FIDMA に入れ換えることで MN の Designated MA のアドレスを得る (3, 4)。
- MN は得られたアドレスのうち、任意の一つを選び、その MA に対して登録要求を行なう (5, 7)。
- MN から登録要求を受けた MA1 は、他の MA すなわち MA2 にこの登録要求を転送する (6)。

### 3.3 従来ノードとの通信手順

次に、従来ノードとの通信手順について述べる。従来ノードは、通信相手となる LIN6 ノードの FQDN から AAAA を引き、そこで示されたアドレスに対してパケットを送出する。す

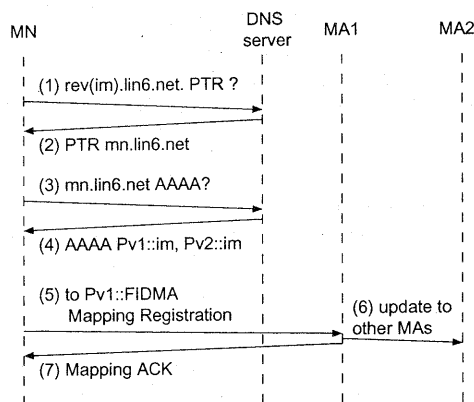


図 4 提案方式における MN の登録処理: MN は自己の FQDN から後方互換アドレスを求め、そのネットワーク上の MA に登録要求を送信する。

なわち、従来ノードが提案方式の LIN6 ノードにパケットを送信する場合には、すべての対象ノードのいずれかの designated MA の仮想ネットワークへと配送されることになる。提案方式では、この仮想ネットワークへと配送されるパケットを MN の現在位置にトンネリングによって配送を行なうことによって移動透過性を提供する。

図 3 を例にすると、MN と従来ノードである LCN の通信手順は図 5 のようになる。

- LCN は MN の FQDN から AAAA レコードを使ってアドレスを解決する (1, 2)。
- LCN は得られたアドレスから任意の一つを選び、パケットを送信する (3)。ここでは MN は MA1 のアドレスを選択したとする。
- パケットは MA1 が持つ仮想ネットワークへと配送される。MA1 は、対象となるパケットの宛先が、自分が Mapping を持つノードへのパケットであるかを宛先アドレスの下位 64bit を見て判断する。この例では MA1 は MN の Mapping を持ったため、MN への IPv6-in-IPv6 のトンネリングを利用してパケットを配送する。MN は、後方互換アドレス宛のパケットを自分宛のパケットとして受信する。
- MN は、LCN に対する応答パケットを MA1 へトンネリングして送出する (5)。
- MA1 は、MN からトンネルされたパケットを decapsulation し、LCN へパケットを転送する。

この結果、MN が移動しても、LCN の上位層が通信相手である MN を認識するアドレスは MA1 が持つ後方互換アドレスであるため変化せず、また、MN へのパケットも MA1 が MN へのパケットを適切な現在位置へと転送するため、従来ノードとの通信においても移動透過性が保証されることになる。

### 3.4 LIN6 ノード間の通信手順

提案方式では MA の発見方式を変更しているため、従来の LIN6 ノード間の通信手順は利用できない。提案方式では、リ

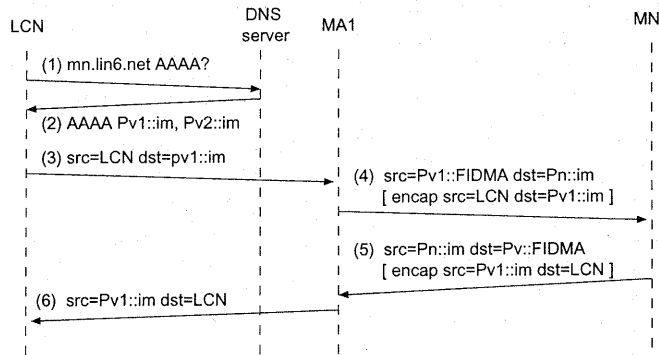


図5 提案方式における通常ノードとの通信手順: LCN は MN の後方互換アドレスに向けてパケットを送信する。MA はこのパケットを MN との IPv6-in-IPv6 のトンネルを利用して転送する。

ゾルバに変更を加えた手法を提案する。

一般的に、アプリケーションは FQDN によって通信相手を指定された場合、これを IPv6 アドレスに変換するために DNS に問い合わせを行なう。この処理はアプリケーション非依存であるので、OS が提供する共通の関数群が利用されることが多い。この関数群はしばしばリゾルバとよばれる。以後本稿においてもこの処理を行なう関数群をリゾルバと呼ぶ。

提案方式では、AAAA レコードを問い合わせた結果が、LIN6ID を含むアドレスであった場合、リゾルバでこのアドレスを LIN6 汎用識別子に変換してアプリケーションに渡すようにする。以下、提案方式における LIN6 ノード間の通信手順を、図3において CN が MN に発呼する例を用いて説明する。

- ユーザあるいはアプリケーションは FQDN 形式、すなわち mn.lin6.net で通信相手を指定する。
- アプリケーションはリゾルバに対して、mn.lin6.net を IPv6 アドレスに変換するように要求する。
- リゾルバは、与えられた FQDN から IPv6 アドレスへの変換を行なうために DNS サーバに AAAA レコードを問い合わせる。この結果、Pv1::im および Pv2::im が得られる。
- リゾルバは、得られた IPv6 のアドレスが LIN6ID を含んでいるため、これを LIN6 汎用識別子に変換する。すなわち、この2つのアドレスは O::im に変換され、アプリケーションに渡される。
- アプリケーションは O::im に対して発呼を行なう。
- O::im のマッピングを得るために、LIN6ID が im であるノードの Designated MA のアドレス、Pv1::FIDMA、Pv2::FIDMA を得る (3.2 章参照)。
- Pv1::FIDMA (あるいは Pv2::FIDMA) に対して Mapping の問い合わせを行なう。

## 4. 考 察

### 4.1 移動ノードの設定の自律性について

まず移動ノードは、通常ノードと通信する際に自分の designated MA が管理する後方互換アドレスをソースアドレスとす

るため、それらのアドレスリストを保持する必要がある。これらは、自分の LIN6ID から、自分に割り当てられた FQDN を求め、その FQDN の AAAA レコードとして定義されている値が後方互換アドレスであるため、ユーザが静的に設定する必要はない。

また、複数の MA を利用する場合、トンネリング先も複数個存在することになる。MN は、各後方互換アドレスについて、トンネルのエンドポイントとして適切な MA を選択する必要がある。しかし、提案方式では、トンネルのエンドポイントは MA が持つ仮想ネットワーク上の既知のインターフェイス ID であるので、トンネルのエンドポイントは、トンネルするパケットの送出時に、ソースアドレスから決定可能である。このため、トンネルのエンドポイントのための特別なテーブル等は必要とならない。

よって、本提案方式によってあらたにノード側で静的に設定する情報はなく、ノードの設定の自律性は高いといえる。

### 4.2 従来ノードとの通信時におけるソースアドレスの選択

LIN6 ノードがパケットを送出する際においてソースアドレスを選択する場合について考える。従来ノードとの通信は、基本的に後方互換アドレスを選択することになる。しかし、後方互換アドレスを選択する場合、すべての通信はその後方互換アドレスに対応する MA を経由する通信となるため、通信路のオーバーヘッドが大きい。とくに短いトランザクションとなることが予想される通信においては、このオーバーヘッドは好ましくない。例えば、DNS への問い合わせパケットなどがこのような通信に適合する。

この問題は、ソースアドレスの選択時に、ユーザなどが指定可能なフィルタを用いて、あるパケットが指定された条件に合致した場合には後方互換アドレスではなく現在移動ノードが利用可能な通常の IPv6 アドレスを選択可能にするようなメカニズムによって回避可能である。

### 4.3 Designated MA の発見処理のオーバーヘッド

従来方式において、Designated MA の発見は、ある LIN6ID に対して、rev(id).lin6.net の MA レコードを問い合わせる

だけで可能であった, すなわち DNS サーバへの問い合わせ回数は 1 回であった. 一方, 提案方式では, rev(id).lin6.net の PTR レコードを問い合わせ, その結果得られた FQDN に対して AAAA を問い合わせる, すなわち 2 度 DNS サーバへ問い合わせる必要があり, オーバヘッドが増加している. しかし, rev(id).lin6.net の PTR レコードを問い合わせた結果の FQDN は, 高い確立でユーザあるいはアプリケーションが指定した通信相手の FQDN になると考えられる. そのため, この FQDN に対する問い合わせの結果は MA の発見の前にすでに行なわれており, ノードが問い合わせに利用する DNS サーバにキャッシュされている可能性が高く, オーバヘッドの増加はそれほど高くならないと考えられる.

#### 4.4 セキュリティ

提案方式においては, 従来の LIN6 に加え, MA とのトンネリングが新たに加わっており, この通信のセキュリティが問題となる. 提案方式では, MA が仮想ネットワーク上に持つ既知のアドレスを, LIN6 の stationary address [6] とすることにより, 移動ノードとのトンネリングに対しても LIN6 を利用し, 移動透過性を得ることができる. このため, このトンネリングに対して IP security (IPsec) [1] を適用することが可能である. すなわち, MA と移動ノードとの間で利用される Security Association (SA) はノードの位置に関わらず一組の SA を継続的に利用可能である. よって, トンネリングのパケットは IPsec によってセキュリティを確保できる.

#### 5. おわりに

本稿では, 移動等価性保証プロトコルである LIN6 を拡張し, 既存の IPv6 ノードとの通信時においても移動等価性保証を提供できる LIN6 の機能拡張方式について議論した. 提案方式は, MA に仮想的なネットワークを配置し, そのアドレスに移動ノードが存在するように見せることで MA へパケットを誘導し, そのパケットをトンネリングによって移動ノードにパケットを配送することで, 既存のノードとの通信においても移動等価性保証を提供可能とした. 同時に, MA レコードを利用せず AAAA レコードに MA の情報を埋め込むことによって, 既存の DNS サーバを変更することなく LIN6 を使用することができるようになり, より LIN6 への移行が容易となった.

今後は本提案の実装を行ない, 実際の性能評価と運用実験を行い, 提案方式が現実のインターネット上で動作することを証明していきたい.

#### 謝 辞

本研究を進めるにあたり貴重な議論をいただいたモバイルインターネットサービス株式会社の太田昌孝氏, 藤川賢治氏, 平原正樹氏およびその関係者の皆様, ならびに奈良先端科学技術大学院大学の Jonathan Khoo 氏に感謝致します.

#### 文 献

- [1] S. Deering and R. Hinden. *Internet Protocol, Version 6 (IPv6) Specification*, December 1998. RFC 2460.
- [2] R. Hinden, M. O'Dell, and S. Deering. *An IPv6 Aggregatable Global Unicast Address Format*, July 1998. RFC 2374.

- [3] IEEE. *Guidelines for 64-bit Global Identifier (EUI-64) Registration Authority*, 1997.
- [4] Masahiro Ishiyama, Mitsunobu Kunishi, Keisuke Uehara, Hiroshi Esaki, and Fumio Teraoka. LINA: A New Approach to Mobility Support in Wide Area Networks. *IEICE Transactions on Communications*, Vol. E84-B, No. 8, Aug 2001.
- [5] David B. Johnson, Charles Perkins, and Jari Arkko. *Mobility Support in IPv6*, June 2002. Internet-draft.
- [6] 國司光宣, 石山政浩, 寺岡文男. 固定ノードと IPv4 ノードを考慮した LIN6 の改良. In *Internet Conference*, November 2001.
- [7] P.V. Mockapetris. *Domain names - concepts and facilities*, November 1987. RFC 1034.
- [8] S. Thomson and C. Huitema. *DNS Extensions to support IP version 6*, December 1995. RFC 1886.