2003－MBL－25 (4)
2003／7／4

多様な名前空間をサポートする 移動透過性保証プロトコルの実装と評価

安間 健介[†] 石山 政浩[††]
國司 光宣[†] 寺岡 文男[†]

本研究では、多様な名前空間をサポートする移動透過性保証プロトコル T40 を設計し、その実装と性能評価を述べる。T40 は Location Independent Network Architecture (LINA) と呼ばれるネットワークアーキテクチャに基づき、複数の識別子空間の識別子と動的に決定する汎用識別子を結びつけ、IPv6 上での移動透過性を実現する。本研究では、T40 を FreeBSD4.7-Release 上に実装し、評価した。T40 の機能を API という形でアプリケーションに提供する。T40 独自の処理に費やす時間は、RTT に比べ非常に小さく無視できることがわかった。2 ノード間では、悪意のある第三者による盗聴に対しても匿名性の保証ができることが分かった。

Implementation and Evaluation of A Mobility Protocol Framework to Support Multiple Namespaces

Kensuke Yasuma,[†] Masahiro Ishiyama,[††] Kunishi Mitsunobu[†] and Fumio Teraoka[†]

This paper presents an implementation and evaluation of a new mobility protocol that supports multiple namespaces in an IPv6 network. The protocol framework has already been proposed. This framework is based on Location Independent Network Architecture(LINA). We show that this protocol can support multiple namespaces with low protocol overhead and that this protocol can assure anonymousness between two nodes.

1. はじめに

IP Version 6 (IPv6)[3] のアドレスは 128bit の長さを持ち、16 進数で表記されるため、ユーザやアプリケーションがその文字列を入力することは現実的でない。そこで、ユーザやアプリケーションが IPv6 のアドレスをより扱い易くするために、ホストに名前をつけ使用したいという要求がある。現在のインターネットでは、Domain Name System (DNS) を使用した Fully Qualified Domain Name (FQDN) による名前解決が一般的である。しかし、DNS の情報は基本的にインターネット上すべてのノードに公開されるため、DNS 上で外部に公開したくない名前を扱うことは難しい。このため、グローバルなネットワークで利用できるプライバシを守った名前空間が求められる。また、zeroconf[9] などで解決可能なローカルな名前の利用も考えられる。Auto-ID[2] は、バーコードの次世代版として個別の識別番号 (id) のみを持つ無線タグを商品につけ、流通の過程でその識別番号を読み管理する手法である。この Auto-ID は、新しい識別子空間の提案である。IP phone の電話番号といった名前空間の例も考えられる。このように、FQDN 以外の名前空間として Auto-id、電話番号、地理位置情報といった名前空間が考えられ、多様な名前空間を扱いたいという要望は大きい。

一方、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。移動透過性をインターネットにおいて実現するために、Mobile IPv6[1]、LIN6[6),12] 等の様々なプロトコルが研究されている。これらのプロトコルでは、FQDN しか扱えない。また、これまでの移動透過性保証プロトコルでは、通信パケットから通信者を特定できるため匿名性を持った通信ができなかった。しかし、信頼がないホストに対しては匿名で通信を行いたいという要求は大きいと考えられる。同様に、ネットワークの盗聴によって自ノードがどのノードと通信しているのかを特定されたくないという要求もある。

我々は、既に LINA (Location Independent Network Architecture)[10] のアーキテクチャを用い多様な名前空間をサポートする移動透過保証プロトコルを提案をしている[11]。本研究では、この多様な名前空間

† 慶應義塾大学大学院 理工学研究科
Graduate School of Science and Technology, Keio University.
†† 東芝研究開発センタ
Corporate Research and Development Center.

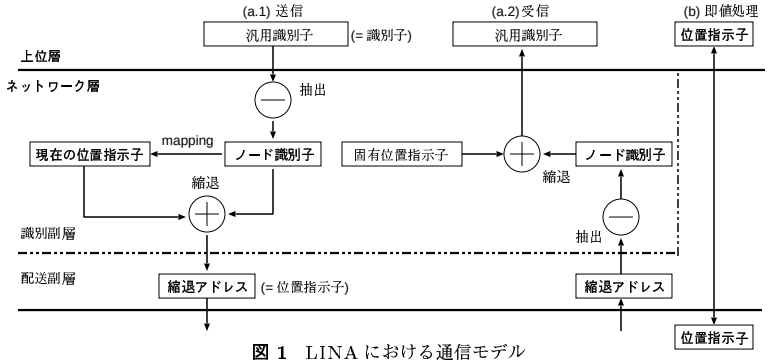

図 1 LINA における通信モデル

をサポートする移動透過性保証プロトコルを設計、実装をし、基本性能評価を行う。

2. LINA の概要

従来のネットワークアーキテクチャでは、ネットワークアドレスが位置指示子とノード識別子という二つの意味を持ち、分離不可能である。また、アドレスの処理はこれら二つの識別子を概念的に分離せずに全てネットワーク層で行われる。LINA ではノード識別子と位置指示子の概念を分離するため、ネットワーク層を概念上２つの副層に分離する。２つの副層でそれぞれノード識別子と位置指示子のための機能が提供される。縮退アドレスモデルを導入し、概念的に２層に分離されたヘッダを一つのヘッダに統合することを実現している。

LINA の通信モデルを図 1 に示す。LINA では、縮退アドレスモデルを用いてノード識別子を位置指示子に埋め込むので、ノード識別子は位置指示子よりも短い。既存のインターネットでは、各層で位置指示子と同様の構造をしたアドレスを扱うが、LINA では位置指示子よりもノード識別子が短いため、そのまま扱うことは望ましくない。そこで、位置指示子と同じ構造の識別子を実現するために固有位置指示子を導入する。上位層では、位置指示子と同一の構造の汎用識別子 (識別子) を用いる。送信時の処理 (図 1(a.1)) はネットワーク層でノード識別子を抽出し、これから現在の位置指示子を導出する。受信時の処理 (図 1(a.2)) は現在の位置指示子を縮退し、縮退アドレスを得る。また、上位層で位置指示子が直接指定された場合 (図 1(b))、そのまま配送が行われる。

LINA では、移動ノードと通信する際に、ノード識別子とそのノードの現在の位置指示子との対応づけを得る必要がある。この対応づけを mapping と呼び、mapping の管理には Mapping Agent (MA) を用いる。各ノードが移動した際には現在の位置指示子を MA に通知する。MA はノード識別子と位置指示子の対応関係を保持し、通信ノードから要求があった場合、指定されたノード識別子に対する位置指示子を通知する。

3. T40 の基本概念

3.1 提 案 指 針

LINA のアーキテクチャを適用することで多様な識別子空間を利用した IPv6 上での移動透過性プロトコルを実現する。複数の識別子空間の識別子と、動的に決定する汎用識別子を結びつけることで IPv6 での通信を実現する。パケットの配送機構は現在の IPv6 インフラストラクチャをそのまま利用する。

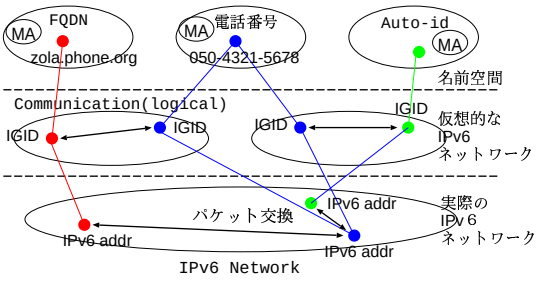

図 2 T40 の通信モデル

T40 の通信モデルを図 2 に示す。FQDN、電話番号、Auto-id といった様々な名前空間の例が考えられる。FQDN の空間の zola.phone.org というノードが電話番号空間の 050-4321-5678 というノードに通信を開始する時、相手の名前空間と名前でノードを指定する。それぞれのノードは、仮想的な IPv6 ネットワーク空間でノード間だけで識別可能な識別子 (IGID) を持つ。IGID は、仮想的に通信するためのアドレスとして使用される。それぞれのノードは、実際の IPv6 ネットワーク空間で実際の IPv6 アドレスを持ち、パケットの配送に使用される。実際の IPv6 アドレスが変わっても IGID は変化しないので、論理的な通信が継続される。IGID と IPv6 ネットワーク上の位置を示す IPv6 アドレスとの関係はノード上で記憶され、パケットの配送に利用される。

3.2 NEN の決定

各名前空間には名前と現在の位置の対応付けをする管理機構である Mapping Agent があり、エンドポイント同士は NEN ネゴシエーションにより通信相手の識別子 (NEN) を決定する。それぞれの識別子は長さ

が違うため、そのまま IPv6 で使用するのは難しい。IPv6 で扱いやすくするために、識別子を Aggregatable Grobal Unicast Address(AGUA)(図 3(a)) の形式にする。Negotiated Endpoint Number(NEN) に Dedicated Prefix(DP) を付与し、仮想的な IPv6 アドレス (仮想汎用識別子, IGID) (図 3(b)) を実現する。上位 64bit は提案方式を実装するノードが知っている既定な固定値である DP であり、続く 64bit は NEN ネゴシエーションによって動的に決定される通信の端点を識別する値である NEN とである。

NEN ネゴシエーションを開始する側が initiator bit を 1 にした値を自分の IGID に使用し、NEN ネゴシエーションを受けた側はこの bit を 0 にした値を自分の IGID として使用する。NEN ネゴシエーションを開始したノードを initiator と呼び、NEN ネゴシエーションを受けたノードを responder と呼ぶ。提案方式においては、アプリケーション及び TCP 等の上位層はこの IGID で通信相手を識別する。この結果、ノードの現在位置が変化しても IGID は変化しないため、移動透過性保証が可能となる。

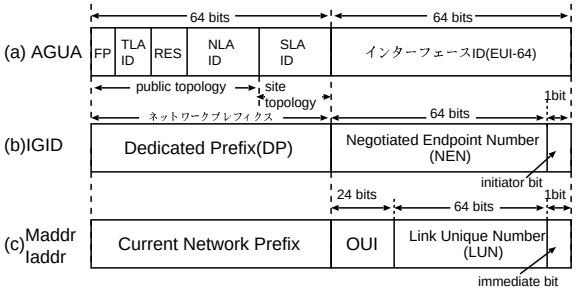

図 3 アドレス形式: IGID, Maddr, Iaddr は IPv6 アドレスと互換性を持つ

3.3 lun の決定

既存の IPv6 ネットワークでのパケット配送は、ノードのインターフェースに割り当てられているアドレス、すなわち位置指示子によって行われる。送信時に上位層によって指定された通信相手が、インターフェースアドレスに変換される必要がある。パケットはインターフェースアドレスによって経路制御されて配送されるため、受信時にはこれを IGID に変換する必要がある。つまり、ノード識別子と位置指示子の相互変換が可能である必要がある。提案方式では、この相互変換を LINA の縮退アドレスを適用し実現する。各ノードは、インターフェースにアドレスを割り当てる際に仮のインターフェース識別子を決定する必要がある。この識別子は AGUA における interface ID(図 3(a)) に相当する値として利用される。仮インターフェース識別子は、図 3(c) に示すような構造を持つ。まず下位 64bit の先頭の 24bit は、提案方式に割り当てられた特定の Organizationally Unique Identifier(OUI) である。これに続く 40bit は、Link Unique Number(LUN) である。immediate bit が 1 である LUN(1) を使い、パケットを配送する場合は抽出等の操作を行わず、そのままデータリンク層に渡される。同様に、immediate bit が 0 である LUN(0) を使い、パケットが配送された場合は縮退等の操作を行わず、そのまま上位層へパケットが渡される。immediate bit とは、LUN の最後の 1bit であり即値処理のための bit である。この bit が 1 であるアドレスを使いパケットが配送された場合は、縮退等の操作を行わずにそのまま上位層へパケットが渡される。各ノードは LUN を決定した後、インターフェースに immediate bit が 0 であるアドレス Maddr と immediate bit が 1 であるアドレス Iaddr の２つのアドレスを割り当てる。自分のインターフェースに割り当てた LUN を local LUN(lLUN) と呼び、相手側のインターフェースに割り当てられた LUN を target LUN(tLUN) と呼ぶ。

3.4 送受信処理

NEN のネゴシエーションの際に、ノードはお互いの名前空間、名前、LUN を通知し合い、NEN を決定する。各名前空間には名前と現在の位置の対応づけ管理機構があり、相手の Maddr は相手の名前から求められる Mapping Agent に問い合わせることで解決する。名前空間、名前、LUN などの情報群を Mapping Entry と呼び、この集合を Mapping Table と呼ぶ。また、NEN と Maddr の対応付けを Mapping と呼ぶ。

この Mapping Table を用いることにより、抽出及び縮退は以下のように実現される。提案方式による送受信処理を図 4 に示す。パケット送信時 (図 4(a.i))、対象となる IGID の下位 64bit から NEN を得る (図 4(1))。次に、送信先の NEN を鍵として Mapping Table を検索することで、その通信で使われている Mapping Entry を発見でき、相手の Maddr が求まる (図 4(2))。パケット受信時 (図 4(a.ii))、送信元 Maddr 及び送信先 Maddr からそれぞれの下位 40bit から LUN を抽出する (図 4(3))。次に、(lLUN, tLUN) を鍵として Mapping Table を検索することで、その通信で使われている Mapping Entry を発見でき、相手の NEN が求まる (図 4(4))。得られたそれぞれの NEN と Dedicated Prefix を連結することで (図 4(5))、IGID が導出される。

OUI の領域を調べることで IP アドレスに Maddr が用いられているか分かるため、既存の IPv6 ノードから送られてきたパケットについては即値処理が適用される (図 4(b))。提案方式は既存の IPv6 との通信では移動透過保証を提供しないが、既存の IPv6 ノードとの通信も可能である。

4. 設　　計

4.1 通信開始時のメッセージフロー

通信開始時のメッセージフローを図 5 に示す。

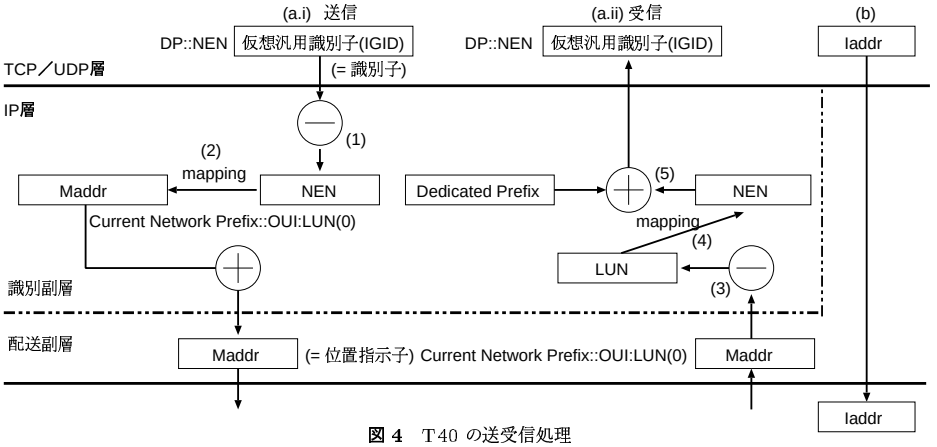

図 4 T40 の送受信処理

図 5 は Node A が Node B へ通信を開始する様子を示す。Node A、Node B の MA はそれぞれ MA1、MA2 である。Node B は、MA2 に現在の Maddr を登録済みである。(a) は MA-Registration メッセージ、(b) は MA-Registration メッセージ Ack、(c) は MA-Request メッセージ、(d) は MA-Replay メッセージ、(e) は NEN ネゴシエーション、(f) は NEN ネゴシエーション Ack をそれぞれ示す。

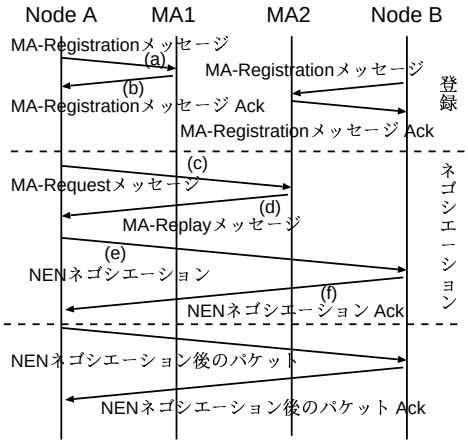

図 5 通信開始時のメッセージフロー

4.2 LUN の決定

通信を開始する前に各ノードはインターフェースの LUN を決定する必要がある。LUN の先頭 39bit は、ランダムな値で決定される。本研究に割り当てられた OUI と下位 40bit のランダムな値は同一リンクで一意である必要がある。LUN の構造を図 6 に示す。

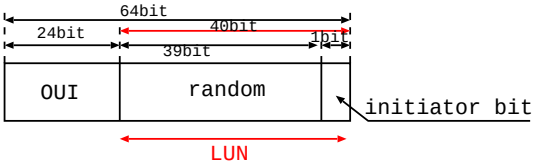

図 6 LUN の構造

4.3 Maddr の登録

各ノードは、LUN を決定した後それぞれのインターフェースに LUN から生成された Maddr 及び Iaddr を付与する。各ノードは自分の名前空間の MA に対して MA-Registration メッセージを送り、現在の Maddr の登録を行う (図 5(a))。また、各ノードは MA のマッピングテーブルが廃棄される前に現在の Maddr を定期的に更新する。

各ノードと MA との情報のやり取りは MA メッセージを使用する。MA メッセージは MA メッセージヘッダを必ずメッセージの先頭に持つ。MA-Registration メッセージは、ノードの Maddr、ノードの名前空間、名前空間での名前を含む。メッセージは名前の長さを含み、任意長の名前を取れる。

MA が MA-Registration メッセージを受理すると MA-Registration メッセージ Ack をノードに返信する (図 5(b))。

各ノードと MA との情報のやり取りは MA メッセージを使用する。MA メッセージは MA メッセージヘッダを必ずメッセージの先頭に持つ。MA メッセージヘッダを図 7 に示す。MA メッセージヘッダの version は、MA メッセージのバージョンを表す。type は、MA メッセージの種類を表し MA-Regist メッセージ等の識別に用いられる。code は、予約された領域で現在は使われていない。auth type は、認証の種類の識別に用いられる。number は、ノードが複数の Maddr を持つ場合、Maddr の番号を表す。reserved は、現在使われてない領域である。

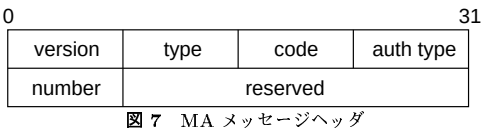

図 7 MA メッセージヘッダ

MA-Registration メッセージを図 8 に示す。MA-Registration メッセージは、ノードの Maddr、ノードの名前空間番号、名前空間での名前を含む。メッセージは名前の長さを含み、任意長の名前を取れる。また、

MA-Registation メッセージ Ack はヘッダのみを持つ。

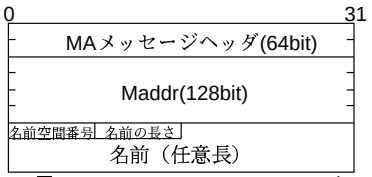

図 8 MA-Registration メッセージ

4.4 Maddr の取得

ノードは MA に MA-Request メッセージを送り、通信相手の名前空間と名前を鍵として Maddr を要求する (図 5(c))。MA-Request メッセージを図 9 に示す。

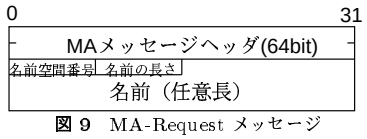

図 9 MA-Request メッセージ

MA-Request メッセージを受信した MA2 は、Maddr を含んだ MA-Replay メッセージを返信する (図 5(d))。MA-Replay メッセージを図 10 に示す。

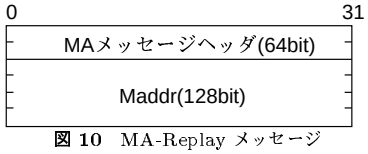

図 10 MA-Replay メッセージ

4.5 NEN ネゴシエーション

各ノード上のユーザやアプリケーションが名前空間を使い通信を開始する場合、NEN ネゴシエーション処理が必要となる。

initiator は initiator 内で一意に決まる NEN の上位 31bit の initiator preferable value を生成し、続く 31bit の responder preferable value をランダムな数値で埋め、structured bit を 1 にする。NEN の構造を図 11 に示す。

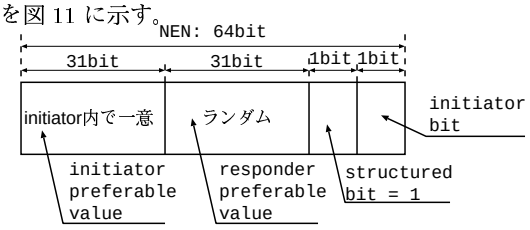

図 11 NEN の構造

initiator が responder に対して NEGOTIATION メッセージを送り、NEN ネゴシエーションが開始される (図 5(e))。NEGOTIATION メッセージを図 12 に示す。NEGOTIATION メッセージ中の Maddr は initiator の Maddr である。また、匿名通信の為に名前空間と名前のフィールドはなくても良い。

NEGOTIATION メッセージを受けとった responder は initiator が指定している NEN が responder 内で一意であるか調べ、NEGOTIATION Ack メッセージを initiator に送る (図 5(f))。responder は NEGOTIATION Ack メッセージを送った後、NEGOTIATION メッセージで受けとった情報を基に responder 内の Mapping Table に新たに Mapping Entry を追加する。また、initiator は NEGOTIATION Ack メッセージの情報を基に initiatorr 内の Mapping Table に新たに Mapping Entry を追加する。現在、衝突処理の実装がなされていないので、NEGOTIATION Ack メッセージはヘッダのみを持つ。この操作をすることでお互いに NEN を持つことが可能となる。

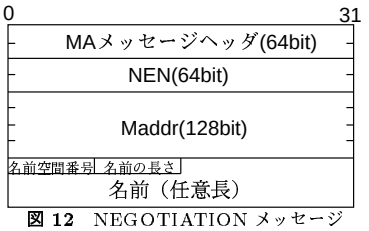

図 12 NEGOTIATION メッセージ

5. 実　　装

5.1 実 装 環 境

本研究では、FreeBSD 4.7-Release のカーネル空間とユーザ空間に実装を行った。T40 は、既存の LIN6 の実装を参考にカーネル内のアドレス変換部を実装した。

5.2 T40 ノードの実装関連図

T40 ノードは、カーネル空間とユーザ空間に実装を行った。T40 ノードの実装関連図を図 13 に示す。

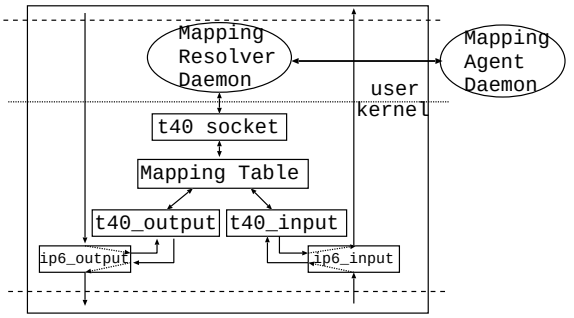

図 13 t40 ノードの実装関連図

アドレス変換の処理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間の情報交換は t40socket を通して行われる。

5.3 カーネル内の実装

カーネル内では、Mapping Table を用いて、IGID と Maddr のアドレス変換を行う。

ip6_output() は、IPv6 パケットが送信時に IP 層で使われる関数である。送信される IPv6 パケットの送信元アドレスと宛先アドレスが IGID であるか、つまり上位 64bit が DP であるかを調べる。IGID であ

る場合は、t40_output() に送られ、アドレスの付け換えを行う。相手の NEN を鍵として Mapping Table を検索し、パケットの送信元アドレスと宛先アドレスをそれぞれの Maddr に書き換える。また、送信される IPv6 パケットの送信元アドレスと宛先アドレスが Iaddr である場合、ip6_output() は即値処理としてアドレスの付け換えを行わず送信する。

ip6_input() は、IPv6 パケットが受信時に IP 層で使われる関数である。in6_input() では、受信された IPv6 パケットの送信元アドレスと宛先アドレスが Maddr であるかを調べ、それぞれが Maddr の場合はアドレスの付け換えを行う。(tlun,llun) の組を鍵として Mapping Table を検索し、ip6_input() はパケットの送信元アドレスと宛先アドレスをそれぞれ IGID に書き換え、上位層へ送る。また、受信された IPv6 パケットの送信元アドレスと宛先アドレスが Iaddr である場合、即値処理としてアドレスの付け換えを行わずに上位層へ送られる。

5.4 Mapping Table

カーネル内の Mapping Table は以下のタプルを持つ。

(local NEN、target NEN、lLUN、tLUN、local Maddr、target Maddr、Lifetime)

local NEN はノード自身の NEN を表し、target NEN は通信先の NEN を表す。local Maddr はノード自身の通信に使用する Maddr を表し、target Maddr は通信先の Maddr を表す。カーネルでは Mapping Table に Lifetime を持ち、t40slowtimo() を使い一定時間毎に Lifetime を減らし、Lifetime が時間切れになったエントリを廃棄する。

5.5 インターフェースアドレス

本実装では現在のインターフェースアドレスと、Maddr、Iaddr を 1 つのインターフェースに付与し、IGID は tforty0 という疑似インターフェースに付与することにより、4 つのアドレスを保持することを実現した。IGID は、カーネル内にキャッシュとして保持することも可能だが、IGID は複数持つことが予想されるので付け換えが容易な疑似インターフェースを用いることにした。また、宛先アドレスが IGID である時、送信元アドレスも対応した IGID を選択する必要があるので、in6_src のソースアドレスセレクションの変更を加えた。

5.6 デーモンの実装

Mapping Resolver Daemon と Mapping Agent Daemon は、デーモンとしてユーザ空間に実装を行った。

Mapping Resolver Daemon は、t40socket を通してカーネルの Mapping Table に様々な情報を操作する機能と、Mapping Agent Daemon と MA メッセージを介し情報交換を行う機能と、他のノードからの NEN ネゴシエーション要求を受信する機能を持つ。また、通信開始前に LUN の生成方法にしたがい、ノード自身の LUN を決定する機能を持つ。Mapping Resolver Daemon は、Mapping Table の Lifetime の半分の時間が経過すると Mapping Agent Daemon に更新メッセージを送信し、Mapping Table を更新する。

Mapping Agent Daemon は MA 上で動作し、MA の Mapping Table を維持する機能と、Mapping Table への登録を受理する機能と、名前空間上の名前で Mapping Table を検索し検出されたエントリ情報を返す機能を持つ。

5.7 ライブラリ

本研究では、T40 のライブラリとして t40lib を用意する。

t40config は、T40 のライブラリを使ったアプリケーションであり、t40resolvd やカーネルへ様々なメッセージを送ることができるツールである。

t40negotiation は、T40 のライブラリを使ったアプリケーションであり、通信相手の名前空間と名前を指定することで NEN ネゴシエーションを行うアプリケーションである。

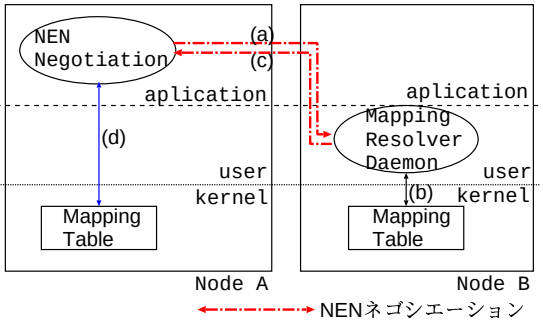

図 14 NEN ネゴシエーションの実装の関連図

NEN ネゴシエーションの実装の関連図を図 14 に示す。図 14 は、Node A が Node B に対して通信を開始する様子である。(図 14(a))。まず、Node A 上の t40negotiation が NEN の生成方法に従い NEN を生成し、Node B に NEN ネゴシエーションメッセージを送信する (図 14(b))。Node B の t40resolvd がその NEN ネゴシエーションメッセージを受信し、Mapping Table にエントリを追加する。Node B は NEN ネゴシエーションメッセージ Ack を Node A に返信する (図 14(c))。Node A は NEN ネゴシエーションメッセージ Ack を受信し、Mapping Table にエントリを追加する (図 14(d))。Node A、Node B はそれぞれの Mapping Table にエントリを追加し、これ以後 NEN による通信が可能となる。

6. 評　　価

6.1 通信開始時のオーバヘッド

図 15 に T40 の通信開始時のオーバヘッドを示す。NEN ネゴシエーションメッセージは、通信開始時に

のみ送信されるメッセージで Mapping Table の検索やインターフェースにアドレスを割り当てるといった操作が行われるため、処理に時間がかかる。LUN が初めて使用される場合、T40 ノード b の Maddr について DAD 処理をしなくてはならない。そのため、LUN が割り当てられ最初のノードとの NEN ネゴシエーションは DAD 処理の時間がオーバヘッドとして付加される。DAD 処理が終った後の NEN ネゴシエーションは、通常の NEN ネゴシエーションで示される処理時間となる。NEN ネゴシエーションは通信開始時のみ一回だけの処理なので、オーバヘッドは小さいと言える。

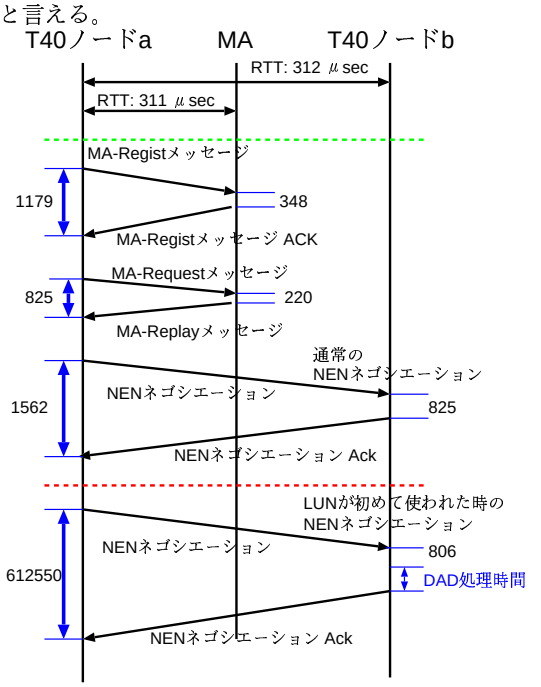

図 15 T40 の通信開始時の評価

6.2 データ送受信時のオーバヘッド

従来の ip6_output()、ip6_input() から t40_output()、t40_input() をそれぞれ呼び出す。t40_output() は宛先アドレスと送信元アドレスをそれぞれの IGID からそれぞれの Maddr に変換する処理を行い、t40_input() は宛先アドレスと送信元アドレスをそれぞれの Maddr からそれぞれの IGID に変換する処理を行う。

この t40_output()、t40_input() が従来の ip6_output() と ip6_input() に対するオーバヘッドとなる。サンプルプログラムとして ping6 を使用し、処理時間の測定にペンティアムカウンタを使用した。測定結果を表 1 に示す。

表 1 output と input のオーバヘッド

t40_output	t40_input
1.44 μsec	0.68 μsec

従来の ip6_output、ip6_input より $2\mu sec$ 以下のオーバヘッドであることが分かった。

また、既存の IPv6 アドレスでの通信と T40 の NEN を用いた通信の比較を行なった。処理時間の測定には ping6 を使用した。比較を表 2 に示す。表 2 から得られる処理時間の差は、$30\mu sec$ になる。既存の IPv6 に比べ、T40 の NEN を用いた通信では、処理時間が 109.6 %になった。

表 2 既存の IPv6 アドレスでの通信と T40 の NEN を用いた通信の比較

既存の IPv6 アドレスでの通信	NEN での通信
312 μsec	342 μsec

実験環境では、トポロジが小さいので非常に RTT が小さい。T40 におけるオーバヘッド 0.030ms($30\mu sec$) は、RTT に比べて非常に小さなオーダであるため誤差の範囲に含まれると言える。

6.3 衝突確率

NEN は図 11 で示すように、2 つの 31bit に分けられた 62bit がランダムで決定される。NEN ネゴシエーションがなされた際にそのランダムで決められた NEN は、ノード内で一意である必要がある。ノード内で一意にできない確率を以下に示す。ノードがすでに保持している NEN の数を n とすると、NEN の衝突確率 p は以下のようになる。

$$p = \frac{n}{2^{62}}$$

既存のインターネットでは、一つのノードが 10000 のコネクションを持つことは稀であり、たとえノードが 10000 のコネクションを持った ($n = 10000$) としても 2^{62} は十分に大きい数であるため、NEN の衝突確率は 2.168×10^{-15} であり、ほぼ無視できる。

NEN は図 6 で示すように、39bit がランダムで決定される。ランダムで決定された LUN は、リンクローカル上で一意である必要がある。ノードがすでに保持している LUN の組を n とすると、LUN の衝突確率 p は以下のようになる。

$$p = \frac{n}{2^{39}}$$

既存のインターネットでは、リンクローカルに 10000 のノードを持つことは稀であり、たとえリンクローカルに 10000 のノードが存在した ($n = 10000$) としても 2^{39} は十分に大きい数であるため、リンクローカル上の LUN の衝突確率は 1.819×10^{-8} であり、ほぼ無視できる。

ノード上での LUN の衝突確率を示す。NEN ネゴシエーションがなされた際に自分の LUN(lLUN) と相手の LUN(tLUN) は、ノード内で一意である必要がある。ノード内で一意にできない確率を以下に示す。ノードがすでに保持している LUN の組の数を n とすると、LUN の組の衝突確率 p は以下のようになる。

$$p = \frac{n}{2^{78}}$$

既存のインターネットでは、一つのノードが 10000

のコネクションを持つことは稀であり、たとえノードが 10000 のコネクションを持つ ($n = 10000$) としても 2^{78} は十分に大きい数であるため、ノード上で LUN の組の衝突確率は 3.309×10^{-20} であり、ほぼ無視できる。

6.4 匿名性の保証

T40 では、ある 2 ノードが通信を行なう時、途中で盗聴されたとしても名前空間上のどのノードとどのノードが通信をしているのかを特定することを不可能にしている。T40 ノードの間の通信は Maddr つまりネットワークプレフィクスとランダムで作成された LUN を使っているので、それぞれの T40 ノードが名前空間上でどのノードであるかを特定することは不可能である。

悪意のある第三者が T40 ノードが MA に対して送信した MA-Registration メッセージを盗聴するとノードの現在の Maddr と名前空間上の名前を得られる。T40 は Maddr の LUN はランダムなので Maddr と名前空間上の名前を盗聴されたとしても次回の通信にはまた違う Maddr が使用されるので匿名性が高いと言える。

今後、各ノードと MA の間の通信には IPsec[7] を利用したパケットの暗号化が必要である。同様に IPsec を利用し NEN ネゴシエーションを暗号化する必要がある。

7. 結　　論

既存のインターネットでは、DNS を使用した FQDN による名前解決が一般的であり、DNS 以外の名前空間をサポートすることはできない。こうした問題を解決するために LINA と呼ばれるネットワークアーキテクチャを用いた多様な名前空間をサポートする移動透過性保証プロトコルが提案されている。そこで本研究では、多様な名前空間をサポートする移動透過性保証プロトコルのプロトタイプである T40 を設計し、その実装と性能評価を行った。

T40 は、MIPv6 などのモビリティサポートプロトコルとの整合性があり、ヘッダオーバヘッドが無い。また、End-to-End の通信を保証し、複数の Mapping Agent を持つことができるので耐障害性、規模拡張性に優れる。T40 の基本性能を評価するために、T40 を FreeBSD 4.7-Release 上に実装を行った。実装では、アドレスの変換部をカーネル空間で処理し、Mapping Agent や NEN ネゴシエーションなどの機構はユーザ空間のプログラムで処理をした。本実装を用い、mapping 情報の登録や取得にかかる時間やパケットの送受信時のオーバヘッドや基本性能について測定したところ、本実装は IPv6 と比較して無視できる程度のオーバヘッドで多様な名前空間をサポートし、移動透過性を付加できることを示した。また、オーバヘッドとなる例外処理の可能性も無視できる程度の確率であることを示した。IP より上位の層は、既存の IPv6 通信と変わらないことを示した。また、悪意のある第三者による盗聴に対して耐性があることを示した。

以上のことから、本研究で設計、実装した T40 は実際のインターネット上で実現可能であることがわかった。T40 は、今後のインターネット上の多様な名前空間を扱うことに有効な手段である。

参 考 文 献

1) D. Johnson C. Perkins and J. Arkko. Mobility Support in IPv6. Internet Draft, work in progress, May 2003.
2) Auto-ID Center. http://www.autoidcenter.org.
3) S. Deering and R. Hinden. Internet Protocol, Version 6 (IPv6) Specification. Internet RFC2460, Dec. 1998.
4) R. Hinden, M. O'Dell, and S. Deering. An IPv6 Aggregatable Global Unicast Address Format. Internet RFC2374, Jul. 1998.
5) IEEE. Guidlines for 64-bits Global Identifier (EUI-64) Registration Authority., 1997. http://standards.ieee.org/regauth/oui/tutorials/EUI64.html.
6) Mitsunobu Kunishi, Masahiro Ishiyama, Keisuke Uehara, Hiroshi Esaki, and Fumio Teraoka. LIN6: A New Approach to Mobility Support in IPv6. In *Proceedings of the Third Inernational Sympsium on Wireless Personal Multimedia Communications*, Nov. 2000.
7) S. Kent, BBN Corp and R. Atkinson Security Architecture for the Internet Protocol Internet RFC2401, Nov. 1998.
8) S. Thomson and T. Narten. IPv6 Stateless Address Autoconfiguration. Internet RFC2462, Dec. 1998.
9) A. Williams. Zeroconf IP Host Requirements. Internet Draft, work in progress, Sep. 2002.
10) Masahiro Ishiyama, Mitsunobu Kunishi, Keisuke Uehara, Hiroshi Esaki, and Fumio Teraoka. LINA: A New Approach to Mobility Support in Wide Area Networks IEICE TRANS. COMMUN., Vol. E84-B, NO.8 Aug. 2001.
11) 石山政浩, 國司光宣, 河野通宗, 寺岡文男. 多様な名前空間をサポートする移動透過性保証プロトコルの 一検討. インターネットコンファレンス 2002 論文集.
12) 國司光宣, 石山政浩, 植原啓介, 寺岡文男. 移動体通信プロトコル LIN6 の性能評価. 情報処理学会論文誌, Vol. 43, No. 2, pp. 398–407, Feb. 2002. マルチメディアコミュニケーションシステム特集.