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

石山政浩[†1] 國司光宣[†2]
河野通宗[†3] 寺岡文男[†4,†3]

本稿では，多様な名前空間上の名前をノード識別子として利用可能な IPv6 上での移動透過性保証プロトコルを提案する．提案方式では，任意の名前空間上の名前をノード識別子とすることにより，ノードの現在位置や識別子の空間の制限を受けることなく，ネットワーク上の位置に依存しない通信が可能となる．同時に，名前を元に識別された相手と動的にノード間で一意に定まる仮想的な識別子の合意をとり，これを IPv6 アドレスと互換性のある形式にすることによって，既存の IPv6 アプリケーションが利用できる．本提案は LINA の上に実現されるため，簡潔でネットワーク負荷の少ない移動透過性を実現する．また，動的な仮想識別子の利用によって，匿名性を保った通信を可能とする．

A Mobility Protocol Framework to Support Multiple Namespaces

Masahiro Ishiyama,[†1] Mitsunobu Kunishi,[†2] Michimune Kohno[†3]
and Fumio Teraoka[†4,†3]

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．

同時に，現在幾つかの移動体通信プロトコルが提案されているが，ノード識別子の自由性についての問題がある．たとえば Mobile IPv6[6] では Home Address がノード識別子となるが，この識別子によって Home Agent の位置が決定するため，Home Agent の位置が変化してしまう場合には，Home Address 自体も変えなければならないといった問題がある．

また，ノードの識別子の多様性の問題もある．現在のインターネットは 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が，DNS を利用した名前は基本的にインターネット上のすべてのノードに公開されるため，公開したくない名前に対して利用することは難しい．このため，グローバルなネットワークで解決可能なプライバシーを守った名前空間が求められる．一方，zeroconf[13] などで解決可能なローカルな名前の利用も考えられる．また，Auto-ID[1] に代表されるような新しい識別子空間の提案もある．このように，インターネットがより普及していくにつれて，さまざまなノード識別子，すなわち名前の利用が考えられる．ネットワークに接続するノードの数が増加する中，移動するノードも増加し，移動透過性保証プロトコルにはこのようなさまざまな名前空間にも対応できる性質が求められると考える．

加えて，プライバシーの問題もある．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可能なため，匿名性を持った通信ができなかった．しかし現在の電話のように，発呼した相手に対して自分の電話番号，すなわち識別子を通知したくない

†1 (株) 東芝 研究開発センター 通信プラットホームラボラトリー
Communication Platform Laboratory, R&D Center, Toshiba Corporation.
†2 慶應義塾大学 大学院 理工学研究科
Graduate School of Science and Technology, Keio University.
†3 (株) ソニーコンピュータサイエンス研究所
Sony Computer Science Laboratories, Inc.
†4 慶応義塾大学理工学部
Faculty of Science and Technology, Keio University.

というニーズ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，自分の通信相手を特定されたくないという要求もある．

以後，本稿では，グローバルに識別可能なノード識別子の集合を扱う空間を「名前空間」と呼ぶ．また，そのインスタンス，すなわちノードを識別する値を単に「名前」と呼ぶ．

本稿では，以下の 4 点を特徴とする新しい移動透過性保証プロトコルを提案する．

任意の名前空間を用いた通信
: 各ネットワークエンティティは任意の名前空間内で定義される名前によって識別される．このため，名前空間による制限を受けず，またより柔軟なノードの識別が可能とする．また，通信相手と同じ名前空間上の名前を必要としない非対象な通信も可能とする．

既存の IPv6 との互換性を保持
: 現在の IPv6 インフラストラクチャとアプリケーションをそのまま利用して，緩やかな移行を可能とする．

単一障害点を持たない効率的通信
: LINA[5] ネットワークアーキテクチャを利用することにより，end-to-end の通信による，単一障害点 を持たない通信メカニズムを提案する．同時に，簡潔でネットワーク負荷の少ない移動透過性を実現する．

匿名通信
: 通信パケットの盗聴では通信している 2 者を名前空間上で特定することを困難とする．また，発呼する側は自分の名前を通信相手に公開せずに通信可能とする．

2. 関 連 研 究

Mobile IPv6 は現在 IETF で標準化が行なわれている，IP 層で移動透過性を保証するプロトコルである．Mobile IPv6 では Home Agent が単一障害点となるが，HA の位置はホームアドレス依存となるため分散化が困難という問題を持っている．同時に，Home Agent の位置が変化してしまう場合には，ノード識別子である Home Address 自身も変更しなければならないといった制約がある．また，通信時に多くの IPv6 拡張ヘッダを利用するため，通信時のパケット長オーバヘッドがある．

Location Independent Networking for IPv6 (LIN6)[5] 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 Mobile IPv6 の問題点を解決しているが，LIN6ID はグローバルに一意性を保証しなければならなたいめ，LIN6ID の配付方式や一意性保証といった管理方法について課題が残っている．

End-to-End Mobility[8] は，ノードが移動しても TCP のセッションを保つ手法を提供している．しかし connection oriented な通信にしか利用できないため，現在注目を集めているストリーミングサービス等への応用には若干の課題がある．また，現在位置の管理のために Dynamic DNS Update[11),12)] を必要とするが，DNS への負荷の集中という懸念もある．

Internet Indirection Infrastracture (i3)[9] は，trigger と呼ばれるランデブーポイントをハッシュ値で決めることが出来るため，比較的名前空間を自由に利用できる．しかし既存のアプリケーションとの互換性がなく，すでにある資産を利用しにくいという問題がある．

3. LINA の概要

本章では LINA の概要について述べる．詳細は[5] を参照されたい．

3.1 基 本 概 念

従来のネットワークアーキテクチャはアドレスに対して位置指示子とノード識別子という二つの意味を持たせていたため，移動透過性保証が困難になっていた．LINA はこの**ネットワークアドレスの二重性**という問題を解決するため，ノード識別子と位置指示子を分離することを前提とした新しいネットワークアーキテクチャである．**ノード識別子**はネットワーク上のノードを位置に依存することなく一意に定めるものであり，ノードが複数のサブネットに接続している場合でも 1 つでよい．また，**位置指示子**はネットワーク上でのノードのネットワークインタフェースの位置を一意に定めるものであり，経路制御のための情報として利用される．位置指示子は 1 つのノードに対して複数保持できる．これら 2 つの概念の分離により，ネットワーク層より上位層では位置に依存しないノード識別子を用いて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．これにより移動透過性が保証できる．

LINA のようなノード識別子と位置指示子の概念の分離に基づく提案はいくつか行われている[2),7),14)] がそれらは主にネットワーク層における改良にとどまっている．しかし，実際の通信においてはアプリケーションからの要求も考慮にいれなければならない．そこで LINA では，ネットワーク層からアプリケーション層までの統合的なアーキテクチャを構築する．

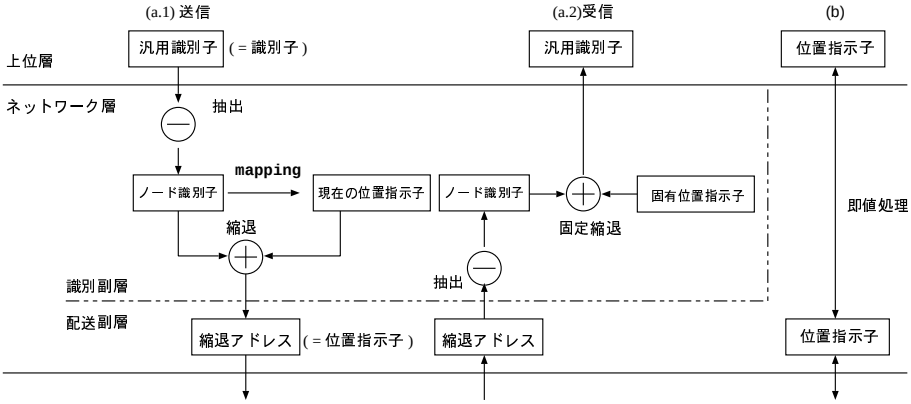

図 1 LINA における通信モデル: 送信時には汎用識別子からノード識別子が抽出され，mapping から得られる現在の位置指示子との縮退操作によって縮退アドレスに変換される．受信時には，縮退アドレスは固定縮退によって汎用識別子に変換される．汎用識別子は上位層における識別子として利用され，縮退アドレスはパケットの経路制御のために利用される．直接位置指示子が指定された通信では，変換操作が行なわれない即値処理が適用される．

3.2 縮退アドレスモデル

従来のネットワークプロトコルでは，アドレスが 2 つのノード識別子と位置指示子の 2 つの意味をもっていたため，これらの処理がネットワーク層という一つの層で処理されていた．しかし，LINA ではこの 2 つの概念は分離されているため，概念上ネットワーク層は 2 つの副層に分離され，それぞれの機能に写像される．位置指示子を扱う層を**配送副層**, ノード識別子を扱う層を**識別副層**と呼ぶ．しかしながら，層の単純な分割はヘッダ長増大の問題や，後方互換性の問題が生じるため現実的ではないと考える．

そこで，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する．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埋め込むという構造を取る．このノード識別子が埋め込まれた位置指示子を**縮退アドレス**と呼ぶ．すなわち縮退アドレスは形式としては位置指示子に従うが，縮退アドレスから位置指示子を得ることが可能である．ノード識別子を位置指示子に埋め込む処理を**縮退**と呼び，逆に縮退アドレスからノード識別子を得る処理を**抽出**と呼ぶ．この縮退アドレスモデルにより，概念的に 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．

次に，ネットワーク層より上位層で用いられる識別子について再考する．識別子の構造は，工学的な見地から位置指示子と同様の構造をしていることが望ましい．しかし，LINA のノード識別子はこの要求を満していない．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**という概念を用いる．固有位置指示子はあらかじめ定められた固定値である位置指示子である．この**固有位置指示子**にノード識別子を縮退させて得られる値を**汎用識別子**と呼ぶ．汎用識別子は位置に依存しない位置指示子に縮退されているため，ノード識別子として利用できる．また，汎用識別子は縮退アドレスでもあるため，位置指示子と同一の形式となる．上位層でこの汎用識別子を用いることにより，アドレスの処理を平易にすることができる．

3.3 ノード識別子と位置指示子の対応づけ

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関係を得る必要がある． この対応関係を **mapping** と呼ぶ．LINA では，mapping を管理する Mapping Agent (MA) と呼ばれる機能を導入する．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 はノード識別子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，指定されたノード識別子に対する位置指示子を通知する役割を担う．

3.4 LINA の動作

LINA の送受信の動作をまとめると次のようになる．送信時 (図 1 (a.1)) に上位層から汎用識別子が指定された場合，まず識別副層で汎用識別子からノード識別子を抽出する．次に MA から mapping を取得し，抽出されたノード識別子に対する現在の位置指示子を導く．得られた位置指示子にノード識別子を縮退させ，縮退アドレスに変換する．縮退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 1 (a.2)) で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．到達パケットから得られた縮退アドレスは識別副層に渡されノード識別子が抽出される．そして抽出

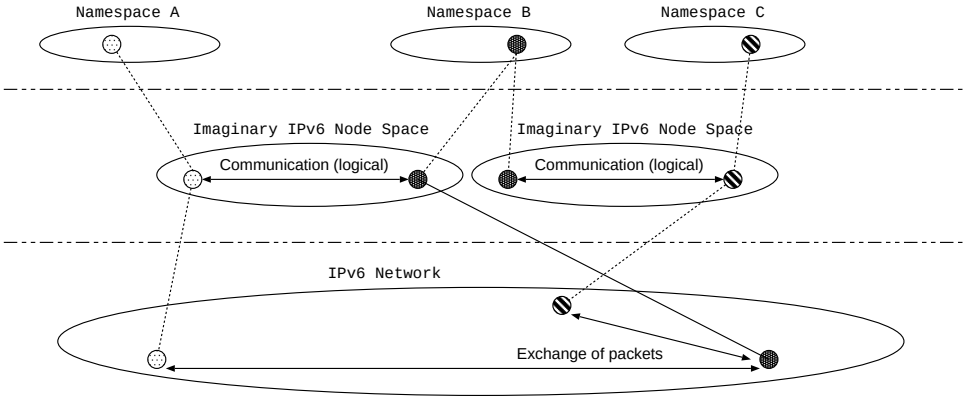

図 2 提案方式の通信モデル: ノードは任意の名前空間上の名前で識別され，仮想的な IPv6 ネットワーク空間へと写像される．仮想的な IPv6 アドレスから，実際のインターフェイスアドレスへの関係をノードが知ることにより，パケットが交換可能になる．

されたノード識別子を固定位置指示子に縮退させ，汎用識別子に変換する．変換された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 1 (b)) では LINA 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．この処理を**即値処理**と呼ぶ．

4. 提 案 方 式

提案方式は，LINA のアーキテクチャを適用することによって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によって IPv6 での通信を実現する．パケットの配送機構は現在の IPv6 インラストラクチャをそのまま利用する．ユーザあるいはアプリケーションはノードを任意の名前空間上の名前で指定する．指定されたノードは動的に決定された仮想的な IPv6 アドレスによって IPv6 ネットワーク上のノードへと写像される．この仮想的な IPv6 アドレスと，IPv6 ネットワーク上の位置を示す IPv6 インターフェイス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 (図 2).

ただし，それぞれの識別子空間にはノード識別子とそのノードの現在の位置情報の関係を管理する Mapping Agent (MA) を発見する解決するための仕組みが必要となるが，これは個々の識別子空間管理方式に依存することになる．

4.1 提案方式における IPv6 上での汎用識別子の実現

現在，IPv6 の通信で主に使用されている Aggregatable Global Unicast Address (AGUA)[3] 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となっている (図 3 (a)).

本提案方式では，この構造を利用して LINA の通信モデルを IPv6 上で実現する．

提案方式においては，ノードの識別は任意の名前空間上の名前で行なわれるため，この識別子を IPv6 上で取り扱えるような中間表現形式を導入する．すなわち，提案方式では任意の名前空間上の名前に対し，仮想的な IPv6 アドレスを動的に生成し，この関係を記憶することによって任意の名前空間上の名前を IPv6 上でも識別できるようにする．この仮想的なアドレスを **仮想汎用識別子 (Imaginary Generalized ID, IGID)** と呼ぶ．これは LINA における汎用識別子に相当する．

IGID は，図 3 (b) に示すような形式となる．上位 64bit は，仮想的な prefix であり，提案方式を実装するノードすべてが知っている既定な固定値である．以後この値を **Dedicated Prefix** と呼ぶ．続く 64bit は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．この値を **Negotiated Endpoint Number (NEN)** と呼ぶ．NEN の最後の 1 bit は **initiator bit** と呼ばれ，ネゴシエーションを始めた側がこの bit を立てた値を自分の IGID に使用し，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．以後，ネゴシエーションを開始したノードを **initiator**，これを受けるノードを **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

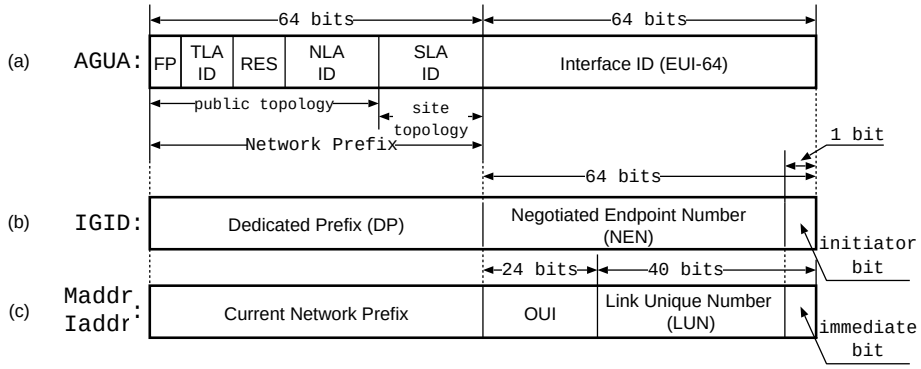

図 3 アドレス形式: IGID, Maddr および Iaddr は IPv6 アドレスと互換性を持つ.

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．

4.2 提案方式における IPv6 上での縮退アドレスの実現

IPv6 ネットワークにおけるパケット配送は，その時点においてノードのインターフェイ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なわれる．ゆえに，送信時に上位層によって IGID によって指定された通信相手は，このインターフェイスアドレスに変換する必要があり，また受信時にはパケットはインターフェイスアドレスによって経路制御されて配送されるため，これを IGID に変換する必要がある．すなわち，ノード識別子–位置指示子の相互変換が可能でなければならない．提案方式ではこれを LINA の縮退アドレスを適用して実現する．

各ノードは，インターフェイスにアドレスを割り当てる際に自分の仮インターフェイス識別子を決定し，AGUA における Interface ID に相当する値としてこの値を利用する．この仮インターフェイス識別子 は，図 3 (c) に示すような構造を持つ．まず最初の 24bit は，提案方式に割り当てられた特定の Organizationally Unique Identifier (OUI)[4] である．これに続く 40bit は，**Link Unique Number (LUN)** と呼ばれるノードが任意に定める値である．LUN のうち最後の 1bit は，**immediate bit** と呼ばれ，LINA における即値処理を実現するための bit である．この bit が立っているアドレスによってパケットが配送された場合は，縮退等の操作を行なわず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，インターフェイスには immediate bit を立てた値と立てない値 2 つのアドレスを割り付ける．以後，immediate bit が立っている アドレスを **Immediate Address (Iaddr)**，immediate bit が立っていないアドレスを **Mobility Address (Maddr)** と呼ぶ．以後，自分に割り当てた LUN は **local LUN (lLUN)** と呼び，相手側の LUN を **target LUN (tLUN)** と呼ぶ．

NEN ネゴシエーションの際に，ノードはお互いの名前空間とその名前，そして LUN を通知しあう．また，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既に分かっている．この結果，ノードは以下のようなタプルを管理する．

(名前空間の識別子, 相手の名前, NEN, lLUN, tLUN, target Maddr)

このタプルを **Mapping Entry** と呼び，この集合を **Mapping Table** と呼ぶ．

Mapping Table を利用することにより，抽出及び縮退は以下のように実現される．抽出は，対象となる IGID または Maddr から NEN を導出する処理で実現される．パケット受信時の抽出は，パケットに使われている送信元アドレス及び送信先アドレスからそれぞれ LUN が求められ，(lLUN, tLUN) を鍵として Mapping Entry を検索することで相手の NEN が求まる．送信時には，IGID の下位 64bit から直接 NEN が求まる．縮退は，対象となる NEN から Maddr を導出する処理で実現される．Maddr は NEN を鍵として Mapping Entry を検索することで求まる．固定縮退は，IGID を導出する処理となり，Dedicated Prefix を対象となる NEN に連結することによって実現される．

4.3 従来の IPv6 ノードとの通信

縮退アドレスが用いられているかどうか OUI の領域を調べれば分かるため，既存の IPv6 ノードから送られてきたパケットはについては即値処理が適用される．これにより，移動透過性保証は提供できないが，既存の IPv6 ノードとの通信も可能となる．

4.4 移 動 処 理

ノードが移動した場合について考える．まず，移動したノードは現在使用しているそれぞれの lLUN に

ついて，Maddr 及び Iaddr をインターフェイスに割り当てる．その後，自分の MA に対して Maddr の更新処理を行なう．現在繁雑に通信していると思われる相手に対しては，新しい Iaddr を使用することにより Maddr の更新通知 (Mapping Update) を直接通信相手に行なってもよい．

一方，各ノードは自分の管理している Mapping Table の各 Mapping Entry について最終更新時刻を記録しておく．パケットを送信する際に使用した Mapping Entry の最終更新時刻が十分に古い場合には，その対象ノードの MA に対して Maddr を問い合わせ，Mapping Entry の内容を更新する．

5. 提案方式の通信例

具体例を利用して提案方式の通信モデルを説明する．Node A と Node B それぞにおいて，所属する名前空間の識別子を NSa, NSb, それぞれの名前空間での名前を Na, Nb, 現在接続しているネットワークプレフィクスを Pca, Pcb とする．また，Dedicated Prefix の値を DP と表記する．

5.1 LUN の決定

各ノードは，自分の LUN を決定する．決定方法は，例えば自分の名前をハッシュにかけたものなど，ランダムになるものが望ましい．決定後， Maddr 及び Iaddr を自分のインターフェイスに割り当てる．Na の LUN を ia とし，immediate bit が 0, 1 であるものをそれぞれ ia(0), ia(1) とすると，Maddr は Pca::ia(0), Iaddr は Pca::ia(1) となる． Nb も同様に Pcb::ib(0), Pcb::ib(1) とする☆.

なお LUN は一つに限らず，複数個持ってもよい．その場合それぞれの LUN の値について Iaddr, Maddr をアドレスとしてインターフェイスに割り当てる．

Maddr が決定したら，自分の名前から自分の MA を求め，その MA に対して Maddr を登録する．この通信のアドレスとしては Iaddr，すなわち Pcb::ib(1) を使用する．

5.2 Maddr の解決

Nb が Na 対して通信を開始するとする．この場合，通信を開始するユーザやアプリケーションは，Na を名前で指定する．この名前から，Na の MA の位置が導出される．MA の導出方式は各名前空間依存であるのでここでは触れない．なお，一つの名前に対して MA は複数個存在してよい．Nb は得られた Na の MA の中から一つ選択し，Na の Maddr を，Na の名前を鍵としてその MA に問い合わせる．なお，この問い合わせの通信にも Iaddr，すなわち Pcb::ib(1) を使用する．

5.3 NEN ネゴシエーション

Nb は，得られた Na の Maddr より Na の Iaddr である Pca::ia(1) を導き，互いの Iaddr を利用して NEN ネゴシエーションを行なう．この結果得られた NEN を nab とし，initiator bit が 0, 1 のものをそれぞれ nab(0), nab(1) とすると，NEN ネゴシエーションが終了した時点において，Nb の Mapping Table の Mapping Entry として以下のタプルが追加されることになる．

(NSa，Na, nab(0), ib(0), ia(0), Pca::ia(0))

これを受けて，アプリケーションには Na の IPv6 アドレスとして DP::nab(0) が伝わる．また，Nb 側のアドレスとしては，initiator bit を立てた値である DP::nab(1) が使用される．

5.4 パケットの送信処理

上位層では従来の IPv6 と全く同じ処理が行なわれる．パケットがインターフェイスから出る直前において，送信アドレスおよび宛先アドレスが IGID かどうか，すなわち，上位 64bit が DP であるかどうかを確認する．Dedicated Prefix が使用されてなければ，即値処理が適用される．Dedicated Prefix が使用されている場合は，Na へ送信するパケットを例にすると，宛先アドレスとして DP::nab(0) が使われているので，この値を鍵に Mapping Table を検索する．すると Maddr として Pca::ia(0) が得られるので，パケットの宛先アドレスをこのアドレスに入れ換える．送信元アドレスとしては，lLUN として ib(0) が分かるので，現在の Prefix である Pcb::を結合して Pcb::ib(0) に入れ換えて送信する (図 4).

5.5 パケットの受信処理

パケットを受信した際，まず送信元アドレスおよび宛先アドレスの形式を確認する．もしアドレスが Iaddr の形式であったり，従来の IPv6 アドレスであったりした場合は即値処理が行なわれる．そうでなければ抽出処理が行なわれる．Nb が Na からパケットを受信した場合，パケットの宛先アドレスと送信元アドレスからそれぞれの LUN の値として (ib(0), ia(0)) の組が得られる．これを鍵として Mapping Table を検索すると nab(0) が得られる．これを固定縮退することによって得られる DP::nab(0) にパケットの送信元アドレスを入れ換え， 宛先アドレスを initiator bit を反転させた値である DP::nab(1) に入れ換える．この結果を得られたパケットを上位層に送る．

☆ 正確には Pcb::提案方式固有の OUI:ib(0) と表記すべきだが，簡潔にするため OUI は省略して表記する．

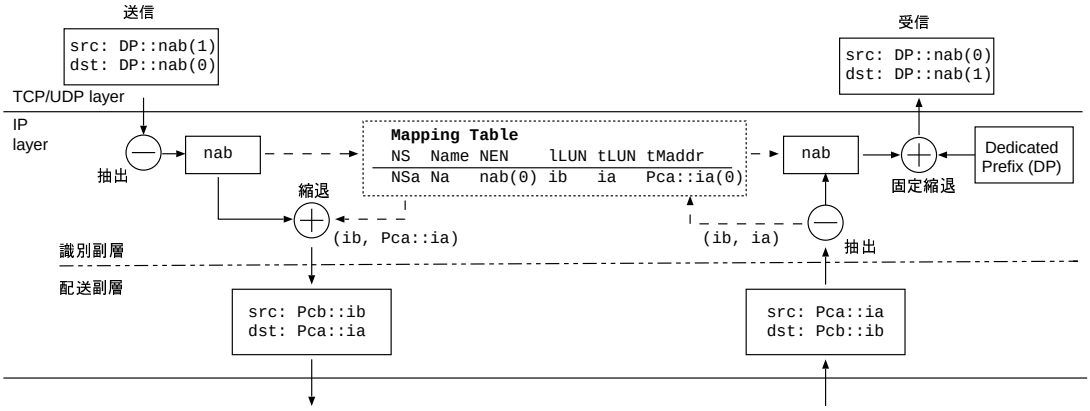

図 4 提案方式による送受信処理: 送信時には IGID から NEN が抽出され，Mapping Table の情報を使用して縮退アドレスである Maddr へと変換される．受信時には Maddr から (lLUN, tLUN) が抽出され，Mapping Table の情報を使用して IGID へと変換される．

6. 衝 突 処 理

提案方式では，幾つかのノードが任意に定める値を利用するため，それらがすでに他のノードで使用されていないかという衝突検知とその処理を行なわなければならない場合がある．以下に，衝突可能性のある値と，その処理方法について述べる．

6.1 NEN が衝突する場合

NEN ネゴシエーションを行なう際に，NEN の衝突が起きる可能性がある．NEN は，通信する 2 者間でのみ一意に定まればよいため，有限の NEN 空間が制限するのは 2 者間のある有限時間内の同時通信者数のみであり，名前空間の数及び名前空間に所属するノードの数ではない．一方，通信者間においては，NEN が衝突しないようにネゴシエーションを行なって決定する必要がある．すなわち，ある任意のノード Na と Nb が通信を開始する場合，Na と Nb がすでに利用している NEN の集合をそれぞれ nen_a, nen_b とした場合，この 2 者間が合意する NEN を n とすると，$n \notin (nen_a \cup nen_b)$ を満たす必要がある．

NEN の衝突を完全になくすことは出来ないが，その確率を下げることは可能である．まず，NEN に図 5 に示すような構造を持たせる．

initiator は，NEN を選ぶ際に，現在使用している NEN の中で initiator preferable value が一意に定められる場合には，structured bit を立てた上で， responder preferable value には乱数を入れて通知する．responder 内で仮にこの値が衝突していた場合でも，responder preferable value を変化させる範囲内で衝突を回避できれば，その値は NEN の条件 を満たすこ

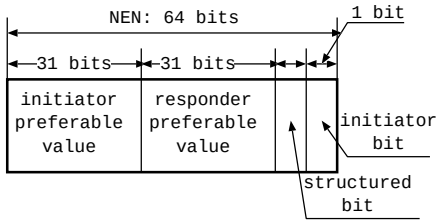

図 5 NEN の構造: 衝突を避けるため，領域を分割し，initiator はノード内で一意となる値を initiator preferable value として選び structured bit を立てる．

とになる．もし initiator が initiator preferable value を一意にできなかったり， responder が，responder preferable value を変化させるだけで衝突回避出来ない場合には structured bit を 0 にして，衝突しない値が見つかるまで乱数を使用する．しかし， initiator preferable value 及び responder preferable value の領域は十分に広いので， structured bit が 0 になるケースはほとんど無視できると考えられる．

6.2 (lLUN, tLUN) の組が衝突する場合

NEN のネゴシエーションの際には，ノード間において，NEN の一意性だけでなく，(lLUN, tLUN) も同様に一意性を保たなくてはならない．これは受信処理の際に (lLUN, tLUN) を鍵にして NEN を求めるためである．すなわち，NEN と同じ条件が (lLUN, tLUN) にも適用される． LUN で自由に定められる空間は 39bit であり，この組の空間は NEN よりも広いので，さらに衝突しにくいと考えられるが，一方で NEN のような手法をとるには問題がある．それは LUN がアドレスに使用されるため，通信相手毎に LUN を選ぶにはオーバヘッドが大きすぎるためであ

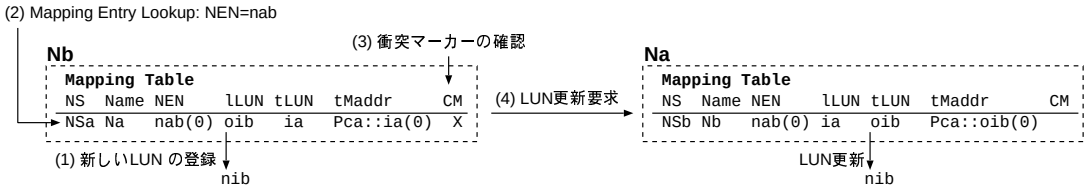

図 6 LUN の衝突処理手順: 新しい LUN を生成し，自己の Mapping Entry を更新する．更新したエントリには衝突マーカー (CM) を付ける．パケットの送信時に対象エントリに衝突マーカー (CM) が付いている場合には，そのエントリの対象となるノードに LUN の更新要求を送る．

る．このため，この組については衝突を検知してからその新しい LUN を決定し，再ネゴシエーションする形となる．最も悪いケースであっても，両方のノードが新しい LUN を生成すれば衝突は回避できる．

6.3 LUN が link 上で衝突する場合

各ノードは，ネットワークに接続した際に LUN を元にインターフェイスにアドレスを割り当てることになるが，その link 上に存在するノードとの間で LUN が衝突している可能性がある．この衝突はアドレスを割り当てた際に行なわれる Duplicate Address Detection (DAD)[10] で検知される．この場合，ノードは衝突した LUN を破棄しなければならない．そして，新しい LUN を生成し，これによって新しく割り当てた Maddr を MA に登録する．しかし，衝突した LUN はすでに通信相手に通知している場合があるため，この更新処理が必要となる．

以下，5 章の例を元に説明する．ここではノード Nb の lLUN が link 上衝突した場合を考える．まず link 上で衝突しない新しい lLUN を定める．衝突した lLUN を `oib`，新しく定めた lLUN を `nib` とする．Nb は nib から Maddr を導出し，MA に登録する (4.4 章参照)．

次に，自分の Mapping Table 上において，衝突した oib を使用している Mapping Entry に対して，衝突マーカー (CM) を付け，oib を nib に更新する (図 6 (1))．

ノードがパケットを送信する際には Mapping Entry を参照することになるが (図 6 (2))，この時にこの衝突マーカーを確認する (図 6 (3))．もし使用する Mapping Entry に衝突マーカーが付いていた場合には，ノードは通信相手に自分の lLUN が nib へと変化したことを通知して衝突マーカーを消す (図 6 (4))．

また，逆に通信相手の LUN が変化した場合について考える．パケットを送信するときには mapping の定期的な更新が行なわれるため，相手からの直接の通知がなくとも Maddr が変化したことを知ることができる．ゆえに通信相手からの LUN の更新通知を待たなくても，パケットの送信は可能である．そして受信側においては，衝突マーカーが付いている Mapping Entry は，すでに新しい nib が含まれているために，このパケットを正しく固定縮退することができる．もしノードがパケットを受信した際に，使用した Mapping Entry に衝突マーカーが付いた場合には，この時点でノードは通信相手に LUN の変化を通知して，衝突マーカーを消すことができる．あるいは，受信時にはこの処理を行なわず，対象ノードにパケットを送信する時点まで LUN の更新を遅延させることも可能である．

この手法により，同時に多くの通信相手を持つノードでも，通信相手にパケットを送信するまでは LUN の更新パケットを送らなくてもよい．ゆえに LUN の衝突によってバースト的な通信が発生するということはない．

7. 考　　察

7.1 匿名性をもった通信

本提案手法では，移動透過性を保証しながらも，以下の 2 種類の匿名性を保った通信が可能となる．

7.1.1 通信パケットの盗聴では通信している 2 者を特定することは困難

通信パケットの送信元アドレス及び宛先アドレスは，自由に定められる LUN を元に行なわれる．LUN からノードを識別する名前への関係は通信を行なっているノードにしかわからない．また，パケットのアドレスとして使用される，ノードの現在位置である Maddr あるいは Iaddr からノードを特定することは難しい．よって，通信パケットのみ盗聴を行なっても通信している 2 者の名前を求めるのは困難である．

7.1.2 発呼側の名前を通信相手に公開せずに通信可能

発呼側から見た場合，相手に名前が伝わるのは NEN ネゴシエーションの際だけである．また，発呼者の名前が必要となるのは以下の 2 つの処理の際だけである．

- 発呼者の MA の発見
- MA に対して，発呼者の Maddr の問い合わせ

よって，以下の手順を導入することにより発呼側は匿名性を保つことが可能となると考えられる．

- MA に対して仮の名前を登録する．これは例えば適当な長さの hash 値である．
- NEN ネゴシエーションの際には，この仮名と，MA のアドレスリストを渡す．

MA のアドレスリストからノードを特定することは困難であるため，発呼側は匿名性を保って通信を行なうことができる．

7.2 名前と Mapping Agent の関係

名前と Mapping Agent の関係は，基本的に名前空間の実装依存となる．すなわち，ある名前空間において，ある名前に対する Mapping Agent の位置の関係を知る方式を定義することにより，提案方式 の移動透過性保証プロトコルが利用可能になる．

本章では，例として名前空間として Fully Qualified Domain Name (FQDN) を使用 し，既存の DNS を利用して MA を解決する方式について述べる．

まず，現在の DNS に対して MA を識別するレコードを新しく定義する．ノードは自分の FQDN に対してこの MA レコードの値を設定する．MA レコードには， 自分の MA のアドレスが入る．

ノードが通信を開始する際には，FQDN で通信相手を指定する．アプリケーションは FQDN から IPv6 アドレスを解決しようとする．この時に，FQDN からまず MA レコードを解決する．もし解決出来た場合，得られた MA のアドレスに対して，対象ノードの Maddr を問い合わせる．この結果 Maddr が得られ，IGID が定まるので，IGID を対象の IPv6 アドレスとしてアプリケーションに通知する．一方，MA レコードがなかった場合は AAAA レコードを解決する．この方式により，既存の IPv6 ノードと通信可能としながら，提案方式も利用可能となる．

8. おわりに

本稿では，多様な名前空間上で識別される名前をノード識別子として利用した IPv6 上での移動透過性保証プロトコルを提案した．提案方式は任意の名前空間で相手を識別しながら現在の IPv6 インフラストラクチャとアプリケーションをそのまま利用でき，また新しい名前空間や，名前解決システムにも容易に適用可能である．また，縮退アドレスを利用したパケットヘッダオーバヘッドのない通信が可能である．加えて，発呼する側が匿名性を保った通信も可能であることを示し，名前空間として FQDN を利用する例を挙げ，緩かな移行が可能であることを示した．

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．

参 考 文 献

1) Auto-ID Center. http://www.autoidcenter.org/.
2) Fumio Teraoka and Keisuke Uehara and Hideki Sunahara and Jun Murai. VIP: A Protocol Providing Host Mobility. *Communications of the ACM*, Vol. 37, No. 8, pp. 67–75, August 1994.
3) R. Hinden, M. O'Dell, and S. Deering. *An IPv6 Aggregatable Global Unicast Address Format*, July 1998. RFC 2374.
4) IEEE. Guidelines for 64-bit Global Identifier (EUI-64) Registration Authority, 1997.
5) 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.
6) David B. Johnson, Charles Perkins, and Jari Arkko. *Mobility Support in IPv6*, June 2002. Internet-draft.
7) Mike O'Dell. *GSE - An Alternate Addressing Architecture for IPv6*, November 1997. Internet-draft.
8) A. Snoeren and H. Balakrishnan. An End-to-End Approach to Host Mobility. In *Proceedings of MOBICOM 2000*. ACM, 2000.
9) Ion Stoica, et al. Internet Indirection Infrastructure. In *Proceedings of the First International Workshop on Peer-to-Peer Systems*. ACM, March 2002.
10) S. Thomson and T. Narten. *IPv6 Stateless Address Autoconfiguration*, December 1998. RFC 2462.
11) P.Vixie, S.Thomson, Y.Rekhter, and J.Bound. *Dynamic Updates in the Domain Name System (DNS UPDATE)*, April 1997. RFC 2136.
12) B. Wellington. *Secure Domain Name System (DNS) Dynamic Update*, November 2000. RFC 3007.
13) A. Williams. *Zeroconf IP Host Requirements*, February 2002. Internet-draft.
14) Xerox. *Internet Transport Protocols. XSIS 028112*, 1981.