# *v6*VIP の設計と実装

舌 間 一 宏[†] 寺 岡 文 男[††]

現在，我々はモーバイルコンピューティング環境に適したネットワークアーキテクチャの構築を目標に研究を行なっている．我々はこれまで移動透過性の実現や移動ノードの認証という観点でアドレス構造の再検討を行ない，アドレス構造の一部にノード自体を識別するためのノード識別子を格納した移動指向アドレスを提案した．さらに，このアプローチの有用性に関する分析を行なうために，移動指向アドレスに基づくネットワークアーキテクチャを，IPv6 をベースに *v6*VIP として現在実装を行なっている．本稿では，*v6*VIP の設計および実装方法について詳しく述べる．

# Design and Implementation of *v6*VIP

Kazuhiro Shitama [†] and Fumio Teraoka[††]

We research to rebuild network architecture for mobile computing environment. In previous works we reconsidered the addressing architecture from a viewpoint of support for host migration transparency and security for mobile node, then proposed *mobility-oriented* address which includes *node identifier* that is unique in internetwork and specifies a node no matter where it is connected. Now, we realize the network architecture based on this addessing approach as *v6*VIP in IPv6 to analyze usefulness and availability of it. In this paper, we describe the design and implementation of *v6*VIP in detail.

## 1. はじめに

近年，コンパクトで高性能な携帯型 PC の普及に伴い，ユーザは PC をいつでもどこへでも持ち運ぶことが可能となった．また単に持ち運ぶだけでなく，その携帯型 PC を移動先のサブネットに接続し，様々なネットワークサービスを受けるという形態も一般化しつつあり，いわゆるモーバイルコンピューティング環境が現実的なものとなってきている．現在，我々はこのようなモーバイルコンピューティング環境に適したネットワークアーキテクチャを構築することを目標に研究を行なっている．

モーバイルコンピューティング環境では，ネットワークに接続しているノードは移動することが前提となるため，ノード間の通信は互いのノードの位置や移動を気にすることなく継続して行なわれるべきであり，このような性質を移動透過性という．また，モーバイルコンピューティング環境でセキュリティを考慮する場合でも，ノードの認証は，ノードの位置や移動に関係なくノード自体に対して行なわれるべきである．

我々はこれまでに，移動透過性の実現や移動ノードの認証という観点でアドレス構造を検討し，ノード自体を識別するためのノード識別子を，アドレス構造の一部に格納した移動指向 (**mobility-oriented**) アドレスを提案した．そして，このアプローチの有用性に関する分析を行なうために，移動指向アドレスに基づくネットワークアーキテクチャを，次世代インターネットプロトコルとして注目されている IPv6(Internet Protocol version 6)[10] をベースに ***v6*VIP** として現在実装している[4]．

本稿では，まず，現在の IPv6 アドレスの仕様に基づいて IPv6 における移動透過性を実現した既存の手法について紹介し，その問題点を示す．その後，移動指向アドレスの考え方を IPv6 アドレスに適用した移動指向 IPv6 アドレスについて説明する．さらに，移動指向 IPv6 アドレスに基づくネットワークアーキテクチャである *v6*VIP の設計および実装方法について詳しく説明する．

## 2. 既存の手法の概要とその問題点

現在の IPv6 アドレスの仕様に基づいて IPv6 におけ

† 慶應義塾大学大学院 理工学研究科 計算機科学専攻
Department of Computer Science, Graduate School of Science and Technology, Keio University
†† (株) ソニーコンピュータサイエンス研究所
Sony Computer Science Laboratory Inc.

る移動透過性を実現した既存の手法として，**Mobile IPv6**[5)6)]と **VIPonV6**[2)3)]の 2 つがある．Mobile IPv6 は，IETF(Internet Engineering Task Force) で開発が進められているプロトコルである．VIPonV6 は，我々が提唱している VIP(Virtual Internet Protocol)[1)]の機構に基づいたプロトコルである．

これら 2 つのプロトコルは，設計思想などは大きく異なるものの，IPv6 アドレス自体は仕様通り，あくまでもインターネット上でノードのインターフェースの位置を示す位置指示子であるとし，ノードの移動に対して不変な識別子を新たに導入するというアプローチをともに採用している．Mobile IPv6 の場合は，ノードのインターフェースを識別するためのインターフェース識別子としてホームアドレスを，VIPonV6 の場合は，ノード自体を識別するためのノード識別子として VIP アドレスをそれぞれ導入し，移動ノードの認証や移動ノードとの通信はこのホームアドレスや VIP アドレスを用いて行なわれる．

確かに Mobile IPv6 や VIPonV6 を用いることで，IPv6 における移動透過性を実現することは可能である．しかし，移動に対して不変な識別子を新たに導入するというアプローチであるため，以下のようなパケットヘッダサイズの増加という問題点が生じてしまう．

通信中の 2 ノードが移動や位置に関係なく互いを認識できるためには，やりとりされるパケットに 2 ノードの識別子を拡張ヘッダ中に格納しておく必要がある．たとえば，VIPonV6 の場合，始点ノード識別子オプションと終点ノード識別子オプションという 2 つの終点オプションヘッダを新たに用意し, 各オプションにそれぞれ始点および終点のノード識別子を格納してパケットを送信している．2 つのオプションのサイズは共に 24 バイトであるため，2 ノードでやりとりされるパケット中のヘッダサイズは通常と比較して 48(24+24) バイト増加してしまう．拡張ヘッダを全く含まない IPv6 ヘッダのサイズは 40 バイトであるため，ヘッダサイズは 2 倍以上となってしまう．この問題はホームアドレスオプションや経路制御ヘッダを用いる Mobile IPv6 でも同様に生じる．帯域が非常に小さいワイヤレス環境で Mobile IPv6 や VIPonv6 を用いる場合，このパケットヘッダの増加は大きな問題になると考えられる．

この問題の本質は，現在の IPv6 アドレス構造自体にあると考えられる．先に述べたように IPv6 アドレスはインターネット上でノードのインターフェースの位置を示すものであり，ノード自体を識別するためのノード識別子を含んでいない．そのため，IPv6 における移動透過性を実現するためには既存のアプローチのように，移動に対して不変なホームアドレスや VIP アドレスを新たに導入せざるを得なかったといえる．そして，この導入が結果としてパケットヘッダサイズの増加という問題を生じさせたと考えられる．

## 3. 移動指向 IPv6 アドレス

2章で示した問題点の本質は，IPv6 アドレスがノードの移動に対して不変なノード識別子を含んでいない点である．しかし，IPv6 アドレス 128 ビット全体をノード識別子としてしまうと，逆にノードの位置に関する情報が完全に欠如してしまうため，サブネット間の経路制御が実質不可能となってしまう．

そこで，我々が提案する移動指向アドレスの一般形として，“ネットワーク指示子 + ノード識別子” のアドレス構造を導入する．このアドレス構造はかつて XNS(Xerox Network Systems)[9)]でも用いられていた構造で，アドレス構造をネットワーク指示子とノード識別子という 2 つの部分に分離して考える方式である．なお，ネットワーク指示子とは，ノードが現在接続しているサブネットの位置を示すもので，そのノードへパケットを送り届けるために用いる．また，ノード識別子とは，ノード自体を識別するもので，ノードの位置や移動に関係なく，ノードを認識したり認証したりするために用いる．

この考え方を IPv6 アドレスに適用すると，まず IPv6 アドレスの下位 64 ビットにはインターネット上でノードを一意に識別するためのノード識別子を格納する．そして，上位 64 ビットにはノードが接続しているサブネットの位置を示すネットワーク指示子 (以後ネットワークプレフィックスと呼ぶ) を格納する．ただし，上位 64 ビットの部分は，IETF で提案されている現仕様[11)] での上位 64 ビットと結果として同じ役割を果たすことになるため，IP 層での経路制御機構をそのまま利用することが可能となる．このアドレス構造を移動指向 **IPv6** アドレスと呼び，図 1に示す．

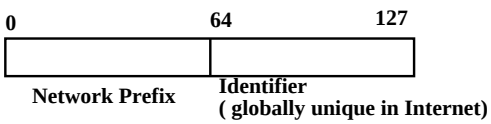



図 1　移動指向 IPv6 アドレス構造

ある移動指向 IPv6 アドレスをもつノード宛てのパケットは，まず上位 64 ビットのネットワークプレフィックスを用いて宛先ノードが接続しているサブネット上

のルータまで転送される．パケットを受信したルータは NDP(Neighbor Discovery Protocol)[14] を用いて，移動指向 IPv6 アドレスに対応する物理アドレスを解決し，最終的に宛先ノードへ転送する．

ここで，移動指向 IPv6 アドレスに格納されているノード識別子とは，ノード自体を識別するものであり，MAC アドレスのような NIC(Network Interface Card) のインターフェース識別子ではない．つまり，ノードに挿している NIC を入れ替えることでインターフェース識別子は変わってもノード識別子は不変である．

移動指向 IPv6 アドレスは，現在提案されている IPv6 アドレスと異なり，アドレス内にノードの移動に対して不変なノード識別子を下位 64 ビットに格納しているため，モーバイルコンピューティングに適している．たとえば，IPv6 アドレス中に格納されたノード識別子を用いて TCP コネクションや IPsec の SA[18] を確立することにより，移動透過性や移動ノードの認証を実現することが可能となる．したがって，既存の手法のように新たな識別子を導入する必要などない．

なお，IPv6 アドレス中にノード識別子を導入した場合の一般的な分析はいくつか行われている[20][21]．移動指向 IPv6 アドレスの有用性についても，こういった分析結果を参考に今後詳しく検討していく必要がある．

## 4. *v6*VIP の設計

3章で述べた移動指向 IPv6 アドレスを基に移動透過性や移動ノードの認証を提供するネットワークアーキテクチャを ***v6*VIP** と呼ぶ．*v6*VIP の基本方針は，(1)IPv6 アドレスとして移動指向 IPv6 アドレスを用い，(2) 移動透過性や移動ノードの認証を実現するための基本概念として VIP の機構を用いる，という 2 点である．

### 4.1 *v6*VIP におけるプロトコル階層

*v6*VIP におけるプロトコル階層を図 2に示す．

TCP/UDP 層より上位層では，移動指向 IPv6 アドレス (以下，特記しない限り IP アドレスとは移動指向 IPv6 アドレスを示す) 下位 64 ビットのノード識別子を用いてノードを識別し，TCP/UDP 層と IP 層の間に挿入した VIP 層でそのノード識別子に対応したネットワークプレフィックス 64 ビットを結合することにより 128 ビットの IP アドレスを生成する．このノード識別子とネットワークプレフィックスの対応付けは，**AMT(Address Mapping Table)** で管理する．こうして生成された IP アドレスを基に IP 層はパケットの送信を行なう．パケット受信時は，VIP 層

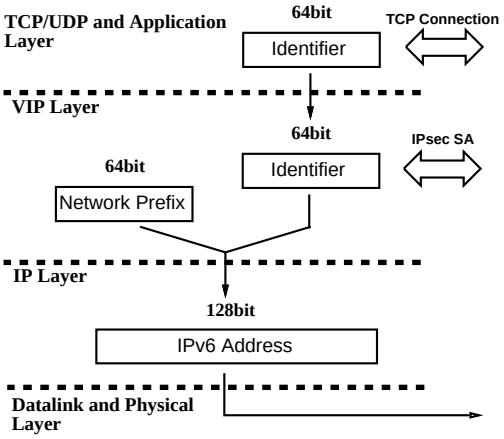



図 2　*v6*VIP におけるプロトコル階層

で IP アドレスからネットワークプレフィックスが単純に除かれ，ノード識別子だけが上位層に渡される．

なお，VIP 層で宛先ノードに関する AMT エントリがない，つまり宛先ノードのノード識別子に対する現在のネットワークプレフィックスが分からない場合は宛先ノードのホームネットワークのネットワークプレフィックスを宛先ノードのノード識別子と結合し IP アドレスを生成する．ノードのホームネットワークとは，そのノードが普段接続されているサブネットを示す．ホームネットワークのネットワークプレフィックス (以後，ホームプレフィックスと呼ぶ) の取得方法は 4.3節で述べる．また，ホームサブネットに接続しているルータをホームルータと呼ぶ．ホームルータは，そのホームルータが管理している移動ノードに関する最新の AMT エントリを保持する．

なお，TCP コネクションを IP アドレス下位 64 ビットのノード識別子を用いて確立することにより，たとえノードが移動した場合でもこれらを継続することが可能となる．また，IPsec SA でも同様に IP アドレス下位 64 ビットを用いることにより，ノードの位置や移動に関係なく，ノードの認証を行なうことが可能となる．

### 4.2 *v6*VIP プロトコル概要

*v6*VIP のプロトコル動作について通信プロトコルとノード移動プロトコルの 2 つの観点から説明する．

#### 4.2.1 通信プロトコル

*v6*VIP において，ノード間の通信は以下のように行なわれる．ここでは，あるノード CN が，HR をホームルータとする移動ノード MN にデータパケットを送信する場合を考える (図 3参照)．

CN が MN に関する AMT エントリを保持してい

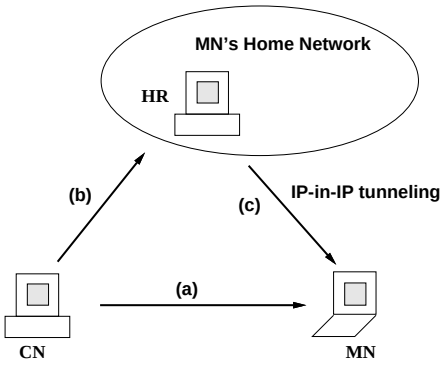



図 3　通信プロトコル

る，すなわち，MN が現在どのサブネットに接続しているかを CN が知っている場合は，MN が現在接続しているサブネットのネットワークプレフィックスと MN のノード識別子を VIP 層で結合して IP アドレスを生成する．IP 層では，その IP アドレスにしたがって経路制御するため，データパケットは，直接 MN に向けて転送される (図 3(a))．

一方，CN が MN に関する AMT エントリを保持していない，すなわち，MN が現在どのサブネットに接続しているか知らない場合は，MN のホームプレフィックスと MN のノード識別子を VIP 層で結合して IP アドレスを生成する．IP 層では，その IP アドレスにしたがって経路制御するため，データパケットは，MN のホームネットワークに向けて転送される (図 3(b))．そのデータパケットを MN のホームルータである HR が受信すると，HR は MN に関する AMT エントリを必ず保持しているため，ホームルータは受信したデータパケットに，MN の移動先の IP アドレスを宛先アドレスとした新しい IPv6 ヘッダを付加し，MN へ転送する (図 3(c)). この転送方法を IP-in-IP トンネリング[16]という．トンネリングを用いる理由は，受信したパケットの終点アドレスフィールドを直接書き換えて移動先のノードへ転送すると，転送したデータパケットが結果としてアドレス偽造パケット[7]だと判断されてしまい，途中のルータで強制的に廃棄される可能性があるためである[8]．

なお，図 3(a), (b), (c) で転送されるパケットの構成をそれぞれ図 4に示す．

VIPonV6 や Mobile IPv6 などの既存の手法の場合は，ノード識別子を新たに導入するという手法であるため，拡張ヘッダを用意する必要があり，データパケット内のヘッダ部分が大きくなるという問題があった．それに対し，*v6*VIP の場合は，図 4からも分かるように，IPv6 アドレス中にノード識別子を格納するという手法であるため，新たに拡張ヘッダを用意する必要はなく，通常の IPv6 パケットと同じサイズとなる．

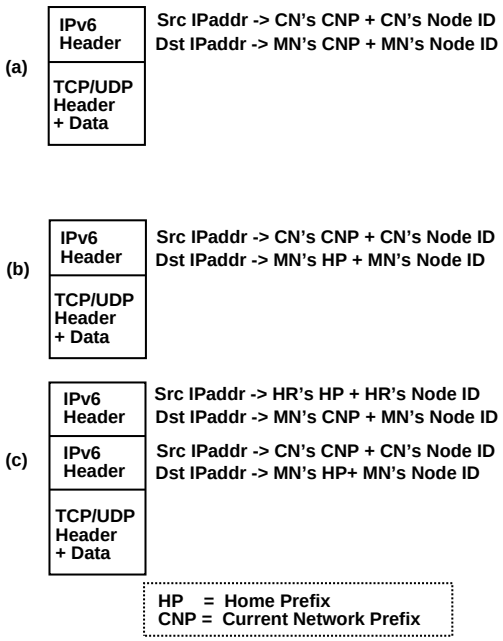



図 4　パケットの構成

### 4.2.2　ノード移動プロトコル

移動ノードはあるサブネットに移動すると，接続先のサブネット上のルータ通知 (router advertisement)[12]を受信し，ルータ通知に含まれるそのサブネットのネットワークプレフィックスと移動ノードのノード識別子を結合し，移動先での IP アドレスを生成する．このように，移動指向 IPv6 アドレスを使用すると，DHCPv6[13]などのステートフル自動設定は特に必要なく，移動先サブネット上で流れているルータ通知を受信するだけで移動先での IP アドレスを獲得可能となる．

移動先のサブネットで新たに IP アドレスを獲得した移動ノードは，その移動ノードのホームルータおよび現在通信中のノードへ AMT アップデートメッセージを送信し，別のサブネットへ移動したことを報告する．この際に用いられるメッセージを **AMT** アップデートメッセージと呼ぶ．

また，移動ノードがホームルータから IP-in-IP トンネリングで転送されたデータパケットを受信した場合，そのデータパケットを送信したノードはこの移動ノードに関する AMT エントリを保持していないと判断して，移動ノードの現在のノード識別子とネットワークプレフィックスの対応付けをそのノードに報告する.

なお，AMT アップデートメッセージには，ホームルータや通信中の相手ノードに不正な AMT エントリが作成されることを回避するために， 拡張ヘッダの 1 つである認証ヘッダ[19]を必ず含まなければならない．

通信中のノードから AMT アップデートメッセージを受信したノードは, メッセージに含まれる認証ヘッダにより送信ノードと通信内容を認証した後，メッセージペイロードに格納されている IP アドレスを取り出し，そのノードが保持する AMT に登録する．ただし，AMT エントリには生存時間が設定されており，タイムアウト後その AMT エントリは削除される．

### 4.3 DNS との連携

*v6*VIP により，VIP 層より上位層のプロトコルやアプリケーションはノード識別子を用いることでノードの位置や移動にかかわらず，TCP コネクションや IPsec SA を継続することが可能となる．しかし，64 ビットで表されるノード識別子は，非常に覚えにくく扱いにくい．さらに，4.1節で述べたように，VIP 層で宛先ノードに対する AMT エントリがない場合のために，通信ノードのホームプレフィックスを獲得する機構が必要となる．*v6*VIP では，この 2 点を解決するために DNS を利用する．つまり，あるノードのホスト名に対して，そのノードのノード識別子 64 ビットとホームプレフィックス 64 ビットを DNS のレコードとして格納しておく．

## 5. *v6*VIP の実装

現在，我々は *v6*VIP を FreeBSD-2.2.5 上で実装している．ベースとなる IPv6 カーネルは，WIDE プロジェクトで開発された hydrangea ☆を用いている．

### 5.1 実 装 方 針

通信時において VIP 層で宛先ノードに関する AMT エントリがない場合，そのノードのホームプレフィックスを知るために，DNS サーバへ要求を出すようにリゾルバを呼び出す必要がある．これにはいくつかの実装方法が考えられる．

まず，宛先ノードに関する AMT エントリがない場合のみ，VIP 層からリゾルバを呼び出し，宛先ノードのホームプレフィックスを獲得するという方法が考えられる．しかしこの方法では，カーネル内からリゾルバを呼び出さねばならないため，実装上非常に面倒である．また，アプリケーションプログラムが宛先ノードのホスト名からノード識別子に変換するためにリゾルバを呼び出すことを考慮すると, もし宛先ノードに関する AMT エントリがない場合，宛先ノードへパケットを送信するためにリゾルバを二度呼び出すこととなってしまい，非常に効率が悪い (図 5参照)．

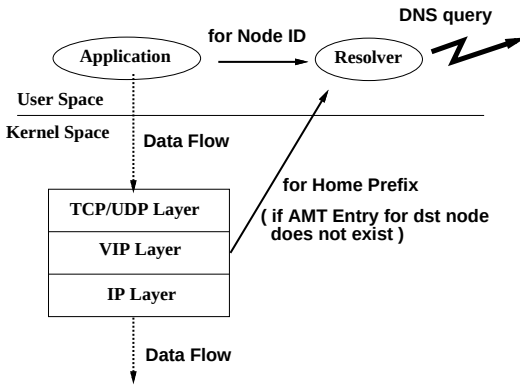



図 5　リゾルバの呼び出し方法 (その 1)

そこで本実装では，アプリケーションが宛先ノードのホスト名からノード識別子に変換するときにリゾルバを呼び出す際に，同時にノードのホームプレフィックスも獲得するようにする (図 6参照)．また現時点では，DNS の AAAA レコードの上位 64 ビットをノードのホームプレフィックス，下位 64 ビットをノード識別子として考え，DNS 自体にノード識別子やホームプレフィックスを格納したレコードの登録や，そのレコードを検索できるようなリゾルバの変更は行なっていない．

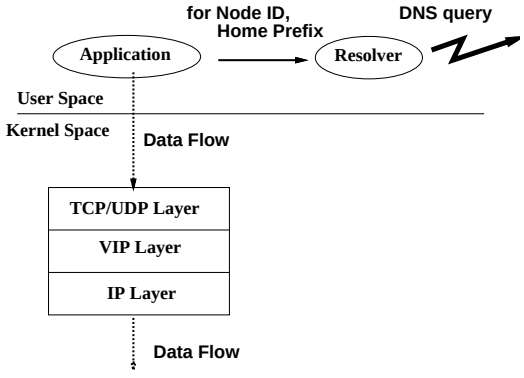



図 6　リゾルバの呼び出し方法 (その 2)

さらに TCP/UDP 層において，IPv6 アドレス下位 64 ビットのノード識別子で TCP コネクションを確立するように，ホームプレフィックスが格納されている IPv6 アドレス上位 64 ビット部分を単純にマスクを取って無視するという手法を現在用いている．

☆ 現在，KAME という名称に変更されている．

そして，VIP 層において，宛先ノードに関する AMT エントリがある場合は，IPv6 アドレスの上位 64 ビットを AMT エントリに格納されているネットワークプレフィックスで置き換える．もし，宛先ノードに関する AMT エントリがない場合は，IPv6 アドレスの上位 64 ビットに格納されているホームプレッフィクスをそのままネットワークプレフィックスとして利用する (図 7参照).

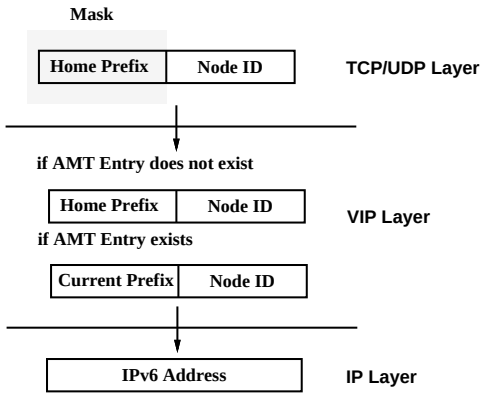



図 7　階層間のアドレスの状態

最後に，モデル的には，IPv6 層と TCP/UDP 層の間に VIP 層を挿入し，VIP 層において *v6*VIP の機能を追加するのが理想的であるが，今回は，実装の容易さを考慮して， IPv6 層に *v6*VIP の機能を直接追加している．

### 5.2　IPv6 層におけるブロックダイアグラム

*v6*VIP の機能を追加した IPv6 層のブロックダイヤグラムを図 8 に示す． IPv6 層に新たに追加した関数について以下それぞれ説明する．

#### 5.2.1　v6vip_output() 関数の動作

まず，送信パケットの宛先アドレス下位 64 ビットで示される宛先ノードのノード識別子に対応する AMT エントリがあるかチェックする．AMT エントリが存在する場合は，送信パケットの宛先アドレス上位 64 ビットを AMT エントリに記載されているネットワークプレフィックスに置き換えて，ip6_output() に処理を戻す．

もし，AMT エントリが存在しない場合は，送信パケットの宛先アドレス上位 64 ビットに格納されている宛先ノードのホームプレフィックスをそのまま利用し，ip6_output() に処理を戻す．

#### 5.2.2　v6vip_input() 関数の動作

ホームルータとして機能しているノードの場合とそうでない移動ノードの場合で動作が異なる．

ホームルータとして機能しているノードの場合，受信したパケットの宛先アドレス下位 64 ビットにそのホームルータが管理している移動ノードのノード識別子が含まれているならば， AMT を参照してそのノードの接続先の IP アドレスを獲得する．そして，その移動ノードへ IP-in-IP トンネリングを用いてパケットを転送する．トンネリングには，hydrangea で採用されている GIF(Generic InterFace)[17]を利用し，GIF の出力系関数である in6_gif_output() に処理を渡す．

一方，移動ノードである場合，受信したパケットがホームルータから IP-in-IP トンネリングで転送されたものならば，その受信パケットを送信したノードへ AMT アップデートメッセージを送信するように，5.4 節で述べる AMT 管理デーモン amtd に要求を出す．

#### 5.2.3　amt_check() 関数の動作

v6vip_input() や v6vip_output() から呼び出され，指定されたノード識別子に関する AMT エントリの有無の参照を行なう．AMT エントリがある場合は，指定されたノード識別子に対応したネットワークプレフィックスを返す．AMT エントリがない場合は–1 を返す．

なお，amt_check() は AMT エントリの検索を行なうだけで，AMT エントリの登録や削除などは，5.4 節で述べる AMT 管理デーモン amtd でのみ行なわれる．

### 5.3　AMT の実装

AMT エントリのフォーマットを図 9 (a) に示す．AMT エントリには，次の 5 つのフィールドから構成されており，1 エントリのサイズは 28 バイトである．

- next pointer フィールド (4 バイト)
  エントリの管理用ポインタである．
- タイムスタンプ (4 バイト)
  そのエントリのノード識別子とネットワークプレフィックスの組のバージョン番号である．
- 生存タイマ (4 バイト)
  このエントリの生存時間である．ある一定時間ごとに減算され，0 になるとこのエントリは削除される．
- ネットワークプレフィックス (8 バイト)
  検索キーによって要求される値である．何も格納されていな場合は，全て 0 である．
- ノード識別子 (8 バイト)
  このエントリの検索キーである．

AMT エントリはハッシュテーブルによって管理される (図 9 (b)). 現在のハッシュ値は単純にノード識別子を 2 進数で表した場合の 1 の個数としている．エ

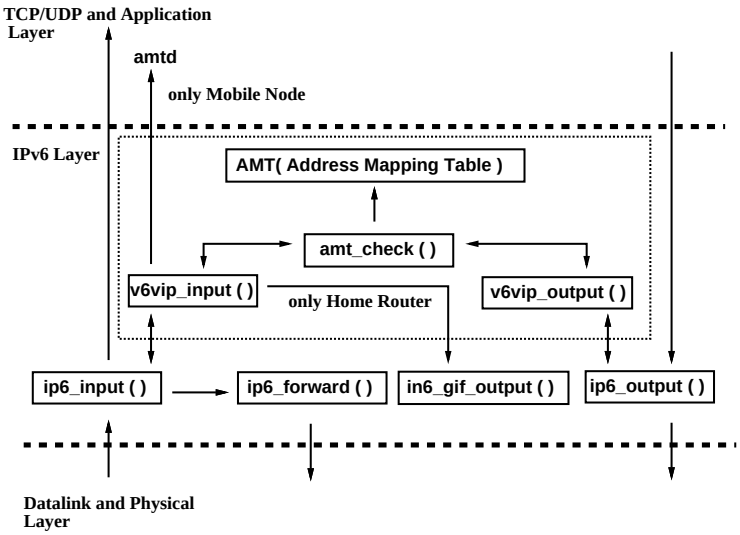
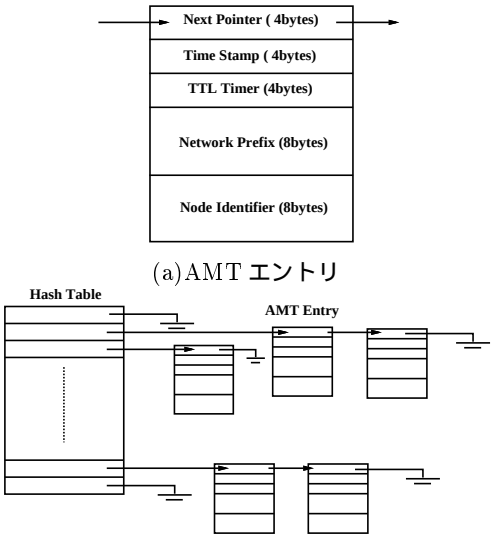



図 8　IPv6 層のブロックダイヤグラム

ントリの追加による衝突が発生するとハッシュスロットにリストとして追加される．

なお，AMT は各ノードの起動時に`ip6_init()` 関数において初期化される．

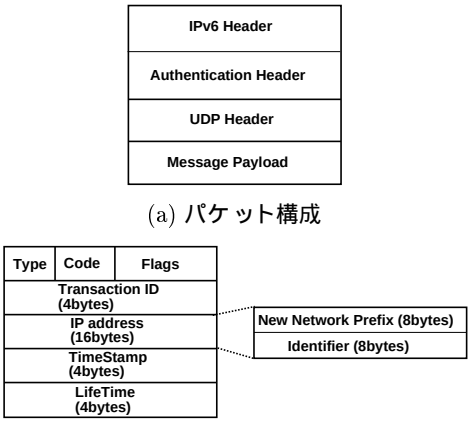
なお，AMT アップデートメッセージは UDP を利用する．AMT アップデートメッセージのパケット構成を図 10 (a) に, メッセージペイロードを図 10 (b) にそれぞれ示す．




(a)AMT エントリ

(b) ハッシュ関数による AMT の管理

図 9　AMT の実装




(a) パケット構成

(b) メッセージペイロード

図 10　AMT アップデートメッセージ

### 5.4　AMT 管理デーモン amtd の実装

AMT アップデートメッセージの送受信およびカーネル内の AMT のエントリの追加と削除を行なう機能は，AMT 管理デーモン amtd としてユーザ空間で実装している．

4.2.2節で述べたように，AMT アップデートメッセージには認証ヘッダが必ず含まれなければならない．また，メッセージペイロードの IP アドレスフィールドに，移動ノードが取得した現在接続しているサブネットのネットワークプレフィックスとノード識別子からなる IP アドレスを格納する．

amtd は，AMT アップデートメッセージを受信すると`ioctl()` システムコールを用いて，カーネル内にあ

るAMTのエントリの更新を行なう．さらに，カーネルからAMTアップデートメッセージの送信要求を受けると即座に，該当ノードへAMTアップデートメッセージを送信する．

## 6. おわりに

我々は，モーバイルコンピューティング環境に適したネットワークアーキテクチャを構築するために，アドレス構造自体を移動透過性の実現や移動ノードの認証という観点から見直し，アドレス構造の一部にノード自体を識別するためのノード識別子を格納した移動指向アドレスを提案している．

本稿では，移動指向アドレスの考え方をIPv6アドレスに適用した移動指向IPv6アドレスを提案し，移動指向IPv6アドレスに基づくネットワークアーキテクチャである*v6*VIPの設計および実装方法について詳しく説明した．

今後は未実装部分の実装を完了させ，移動指向IPv6アドレスおよび*v6*VIPの性能分析を行なうことにより，移動指向アドレスに基づくネットワークアーキテクチャの有用性などについて検討していく予定である．

## 謝　辞

*v6*VIPを実装するにあたり，WIDEプロジェクトおよびKAMEプロジェクトによって開発されているIPv6カーネルを大いに利用させて頂いている．よって，ここに感謝致します．

## 参考文献

1) F. Teraoka, K. Uehara, H. Sunahara, J. Murai, *VIP: A protocol providing host mobility*, CACM, vol.37, no.8, pp.67-75, August 1994
2) F. Teraoka, K. Uehara, *Mobility Support in IPv6 based on the VIP mechanism*, Proc. INET'95, June 1995
3) F. Teraoka, *Mobility Support with Authentic Firewall Traversal in IPv6*, IEICE TRANS. COMMUN., vol. E80-B, no.8, August 1997
4) 舌間一宏，寺岡文男, *IPv6*における移動透過性の実現, 情報処理学会 モーバイルコンピューティング研究会 報告集, 1998年5月
5) J. D. Solomon, *Mobile IP, The Internet Unplugged*, Prentice Hall, 1998
6) C. Perkins, J. David, *Mobility Support in IPv6*, MOBICOM'96, November 1996
7) Computer Emergency Response Team ( CERT), *IP Spoofing Attacks and Hijacked Terminal Connections*, CA-95:01, January 1995
8) 舌間一宏, 寺岡文男, *VIPv4*の設計と実装, 日本ソフトウェア科学会第14回大会 論文集, 1997年9月
9) Xerox Corp, *Internet Transport Protocols*, XSIS028112, December 1981
10) S. Deering, R. Hinden, *Internet Protocol, version 6 (IPv6) specification*, RFC1883, IETF, December 1995
11) R. Hinden, M. O'Dell, S. Deering, *An IPv6 Aggregatable Global Unicast Address Format*, Internet Draft, work in progress, March 1998
12) S. Thomson, T. Narten, *IPv6 Stateless Address Autoconfiguration*, RFC1971, IETF, August 1996
13) J. Bound, C. Perkins, *Dynamic Host Configuration Protocol for IPv6 (DHCPv6)*, Internet Draft, work in progress, March 1998
14) T. Narten, E. Nordmark, W. Simpson, *Neighbor Discovery for IP Version 6 (IPv6)*, RFC1970, IETF, August 1996
15) A. Conta, S. Deering, *Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (IPv6) Specification*, RFC1885, IETF,
16) A. Conta, S. Deering, *Generic Packet Tunneling in IPv6 Specification*, Internet Draft, work in progress, January 1998
17) 山本和彦, *IP*トンネルのモデル化と実装, ソフトウェア科学会 コンピュータソフトウェア, Vol.15, No.2, pp.38-47, 1998年3月
18) R. Atkinson, *Security Architecture for the Internet Protocol*, RFC1825, IETF, August 1995
19) R. Atkinson, *IP Authentication Header*, RFC1826, IETF, August 1995
20) M. O'Dell, *GSE - An Alternate Addressing Architecture for IPv6*, Internet Draft, work in progress, February 1997
21) M. Crawford, A. Mankin, T. Narten, J. W. Stewart III, L. Zhang, *Separating Identifiers and Locators in Addresses: An Analysis of the GSE Proposal for IPv6*, Internet Draft, work in progress, March 1998