第 7 部

移動ノード

第 1 章

はじめに

ここ数年、ネットワークの普及にはめざましいものがある。同時にコンピュータも性能が向上し、よりコンパクトになりつつある。このような環境では、ユーザは各自のコンピュータを持ち歩き、地理的な位置に関わらずにネットワークにアクセスしたいと思うであろう。そうなると、広域ネットワークでは「ホスト移動透過性」が重要な性質の一つになると思われる。ホストが移動すると、ホストのアドレスやそのホスト上のアプリケーション・エンティティのアドレスも変化する。既存のネットワーク環境では、ホスト移動はユーザの計算機環境に変化をもたらす。たとえば、ユーザはコンピュータに新しいアドレスを割り当てなければならないし、そのアドレスを関係のあるコンピュータに通知しなければならない。ユーザの観点からは、ホスト移動透過性とはユーザが広域ネットワークのどこにコンピュータを接続しても常に同一の計算機環境を得られることと定義できる。またシステムの観点からは、アプリケーション・エンティティが自分自身や相手エンティティの物理的な移動を認知しないことと定義できる。

OSI [?] や TCP/IP のような既存のプロトコルはホスト移動について考慮していない。また、Mach [?]、V-system [?]、Chorus [?] などの分散システムも開発されており、これらはネットワーク透過性は実現しているが、ホスト移動はやはり考慮していない。例外として、Amoeba は同報通信パケットが届くサブネットワークの範囲でホスト移動をサポートしている [?]。また、XNS [?] はネームサーバ機構を用いて移動するホストの追跡を可能にしている。しかしこれらのシステムで移動ホストと通信するためには、ユーザが相手ホストの位置を考慮しなければならない。

過去においてホストの移動性に関する研究はわずかしかなかったが [?]、ここ数年この分野は次第に注目されはじめている [?] [?]。[?] の方式は、ホストのネットワークアドレスの拡大解釈およびソースルーティングという 2 つの方式に基づいている。この方式では、移動ホストの位置情報を保持する動的な広域データベースを利用する。各移動ホストは移動したときに新しい位置情報をこのデータベースに登録し、移動ホストと通信したいホストはこのデータベースを引いて相手のアドレスを知るのである。しかしこの方式では、データベースのアクセスのためトラフィックが大きく増加すると考えられる。さらにこのデータベースが分散されていると、整合性を保つのが非常に困難になる。[?] では PAS (the Permanent IP-Address Scheme)、TAS (the Temporary IP-address Scheme)、および ENS (the Embedded Network Scheme) という 3 種類の IP アドレスの割り当て法について利点、欠点が論じられている。しかしそこではどのように IP アドレスを割

り当てるか、どのようにルーティングテーブルを操作するか、という議論に限られてしまっている。[?] では IP-within-IP データグラムにより、無線のネットワークインターフェイスを持つ移動ホストとの通信をキャンパス内でサポートしている。各移動ホスト (MH) は移動に依存しない一意な IP アドレスを持つ。各移動ホストサポートステーション (MSS) は無線のサービス領域を持ち、MH を追跡する。しかし通常のホストと MH の通信を可能にするため、すべての MH の IP アドレスは同一の (サブ) ネットワーク番号を持たなければならず、MSS が自分のサービス領域に存在しない MH にパケットを転送するときは、他の MSS に相手 MH の位置情報を聞き回らなければならない。このようにこの方式には拡張性の点で問題があると思われる。

上述した方式はいずれもネットワークアーキテクチャに関して充分な考察をしておらず、単に位置を変えるホストとの通信方法を提案するに留まっている。本研究はホスト移動透過性を実現する新しいネットワークアーキテクチャを提案し、このアーキテクチャに基づいたプロトコル [?]。を開発することを目的としている。本報告書ではこのアーキテクチャを IP (Internet Protocol) に適用して VIP (Virtual Internet Protocol: 仮想インターネットプロトコル) を提案し [?]、その実測したオーバーヘッドも示す [?]。この結果、VIP は無視し得るほどのオーバーヘッドでインターネット上でホスト移動透過性を実現できることがわかる。第 2 章ではホスト移動透過性とは何か、どのレイヤで実現すべきか、について議論する。第 3 章ではホスト移動透過性を実現するために「仮想ネットワーク」という概念を導入し、これを効率良く実現するために従来のネットワークレイヤを 2 つのサブレイヤに分割する。第 4 章では「拡散キャッシュ法」に基づいた仮想ネットワークプロトコルを提案する。第 5 章では VIP の設計の詳細について述べる。第 6 章では SONY NEWS ワークステーション上の VIP の実装について述べる。第 7 章では実測した VIP のオーバーヘッドを示す。第 8 章でまとめを述べる。

第 2 章

ホスト移動透過性

2.1 ホスト移動透過性の定義

まずホスト移動とは何かを定義する。広域ネットワークは多数のサブネットワーク (またはローカルエリアネットワーク (LAN)) の相互接続によって構成され、ホストは一時には一つのサブネットワークに接続されていると考えることができる。このような環境では、ホスト移動とはホストがネットワークに対する接続点をあるサブネットワーク (OSI の IS-IS ルーティングモデルによればエリア) から別のサブネットワークに変更することであると定義できる。従ってホスト移動によりホストのネットワークアドレスは変化する。ホストがあるサブネットワークから切断され、同じサブネットワークの別の地点に接続された場合、そのホストのアドレスは変化しないので、これはホスト移動とは考えない。ホスト移動には 2 つの形態がある。「on-line 移動」と「off-line 移動」である。off-line 移動の場合、移動前にホストはネットワークから切断され、移動後再びネットワークに接続される。移動中はネットワークにアクセスできないが、スタンドアロンマシンとして利用することは可能である。on-line 移動の場合は移動中もホストはネットワークにアクセスでき、通信チャネルも維持されてコネクション・オリエンテッド・モード通信も可能である。現在の有線ネットワークでは、物理的な制約上 off-line 移動しかできないが、ディジタルセルラ電話のような無線ネットワークがデータリンクとして利用可能になれば、on-line 移動も可能になると考えられる。本報告書では、 off-line 移動しかできないホストを「portable host」、on-line 移動もできるホストを「mobile host」と呼ぶことにする。

次にホスト移動透過性の定義を行なう。ユーザがネットワークに接続されているコンピュータを使用する場合、コンピュータ上の作業が他のコンピュータ上の特定のファイルやサーバプロセスに依存することがある。これは計算機環境の一部である。移動により自分のコンピュータが相手から正しく認識されなくなると、他のコンピュータのファイルやサーバプロセスにアクセスできなくなる可能性がある。ユーザにとってホスト移動透過性とは「ユーザが広域ネットワークのどこに自分のコンピュータを接続しても同じ計算機環境を得られること」と定義できる。ユーザに対するサービスは、アプリケーション・エンティティ同士の通信により提供される。コンピュータが移動してそのコンピュータのアドレスが変化すると、他のコンピュータからは移動したコンピュータが識

別できなくなり、そのコンピュータ上のアプリケーション・エンティティも識別できなくなる。ホスト移動に伴うアプリケーション・エンティティの移動を隠蔽することができれば、アプリケーション・エンティティ同士はホスト移動に関わり無く相手を識別して通信を行なうことができる。すなわちシステムにとってホスト移動透過性とは「ホストが移動してもアプリケーション・エンティティはその移動を意識することなく、また何の変更も加えずに相手エンティティを識別して通信できること」と定義できる。

2.2 ホスト移動透過性を提供すべきレイヤ

移動するホストを追跡する方法はいくつか考えられる。たとえば BIND [?] や X.500 [?] のようなネームサーバやネットワークディレクトリはユーザに対して一種の移動透過性を提供することができる。アプリケーション・エンティティはネームサーバに問い合わせることで移動した相手エンティティのアドレスを得るのである。しかしこの方法にはいくつかの問題点がある。まず第一に、アプリケーション・エンティティの修正無しには on-line 移動が実現できない点である。無線データリンクを持ったホストがある無線ネットワークから別の無線ネットワークに移動するとそのホストのアドレスは変化する。アプリケーション・エンティティは相手ホストが移動していることを知る必要があり、また通信チャネルを維持するには移動しているホストの新しいアドレスを得なければならない。第二に、この方式ではネットワーク・トラフィックが増大する。アプリケーション・エンティティとネームサーバ間の通信およびネームサーバがローカル・キャッシュを使っていれば、キャッシュ更新のためのネームサーバ間の通信が必要になるからである。第三に、ネームサーバがローカル・キャッシュを使っている場合、いつキャッシュを更新すべきかが明確でないことである。また同報通信方式もユーザに対して一種の移動透過性を提供することができる。移動ホストにパケットを送信したいホストは問い合わせパケットを同報通信する。移動ホストがこれを受信して現在のアドレスを返すことにより、送信側ホストは移動ホストにパケットを送信できるのである。しかしこの方式の重大な欠点は、広域ネットワークでは使用できないことである。広域ネットワークで同報通信を行なうと、非常に大きなトラフィックの増加を引き起こすからである。

これら 2 つの方式ではアプリケーション・エンティティがホストの位置を認識しなければならない。たとえライブラリ・ルーチンがホストの位置を隠蔽したとしても、移動ホストを追跡するのはアプリケーション・エンティティである。すなわち、アプリケーション・エンティティは送信時に毎回相手の位置の変化を認識し、もし相手が移動していたら新しいアドレスを得なければならないのである。

OSI の 7 層モデルに基づいたネットワークアーキテクチャでは、ホスト間通信はネットワークレイヤが提供するサービスであり、エンド間通信はトランスポートレイヤが提供するサービスである。言い換えれば、トランスポートから上位のレイヤにはホストという概念はなく、相互接続されたネットワークにおいてホストの位置を認識するのはネットワークレイヤである。従って OSI のアーキテクチャによれば、ホストの位置の隠蔽を行なうのはネットワークレイヤが適当であると考えられる。ネットワークレイヤが上位

レイヤにホスト移動透過性を提供すれば、アプリケーション・エンティティには修正を加える必要もなく、また on-line 移動の実現も可能になる。

2.3　基本的サービス

エンド間通信サービスはトランスポートレイヤ以下が提供する。エンド間通信サービスでホスト移動透過性を実現するには、トランスポートレイヤは通常のサービスに加えて以下の 4 つのサービスを提供しなければならない。

- (T-1) 移動透過トランスポート・エンティティ・アドレシング: 上位レイヤに対して、移動に依存しない[1]トランスポート・エンティティの識別子を提供すること。
- (T-2) 移動透過コネクションレス・モード通信: 移動に関わらず、トランスポートレベルのコネクションレス・モード通信を行なうこと。
- (T-3) 移動透過コネクション設定: 移動に関わらずに、トランスポートレベルのコネクションを設定すること。
- (T-4) 移動透過コネクション・オリエンテッド・モード通信: 移動中もトランスポートレベルのコネクションを維持し、コネクション・オリエンテッド・モード通信を行なうこと。

off-line 移動透過性は portable ホストと mobile ホストに提供されるが、これは T-1 から T-3 までのサービスで実現される。on-line 移動透過性は mobile ホストに提供されるが、これは T-1 から T-4 のすべてのサービスで実現される。

ここに挙げたトランスポートレイヤのサービスを実現するには、ネットワークレイヤは通常のサービスに加えて以下の 6 つのサービスを提供しなければならない。なお、ネットワークレイヤにはコネクションレス・ネットワーク・サービスを仮定している。

- (N-1) 移動透過ホスト・アドレシング: 上位レイヤに、移動に依存しないホストの識別子を提供すること。たとえ相手ホストが移動しても、トランスポートレイヤは同一のホスト識別子で相手ホストを指定することが可能になる。
- (N-2) 移動透過コネクションレス・モード通信: 移動に関わらずに、ネットワークレベルのコネクションレス・モード通信を行なうこと。
- (N-3) 切断要求: ネットワークからの切断準備をする機能。
- (N-4) 切断指示: on-line 移動において、現在の無線ネットワークから他の無線ネットワークへの移動準備のため、下位レイヤから呼び出される機能。

[1]「移動に依存しない」とは、on-line 移動に関しては「移動前、移動中および移動後に依存しない」という意味であり、off-line 移動に関しては「移動前と移動後に依存しない」という意味である。

- (N-5) 接続要求: ネットワークへの接続準備をする機能。
- (N-6) 接続指示: on-line 移動において、ある無線ネットワークを離れた後、別の無線ネットワークへ接続するときに下位レイヤから呼び出される機能。

T-1 は N-1 によって実現される。T-2 と T-3 は N-1、N-2、N-3 および N-5 によって実現される。T-4 は N-2、N-4 および N-6 によって実現される。

ネットワークレイヤにおいては、N-3、N-4、N-5 および N-6 は下位レイヤから提供されるサービスで実現される。従って N-1 と N-2、すなわち「移動透過ホスト・アドレシング」と「移動透過コネクションレス・モード通信」がネットワークレイヤ特有のサービスであり、ホスト移動透過性を実現するための基本的なサービスであると考えられる。これら 2 つのサービスを提供するための新しいネットワークアーキテクチャを次章で導入する。

第 3 章

仮想ネットワーク

3.1 概念

既存のネットワークではホストのネットワークレイヤ・アドレスは階層構造を持っている。たとえば以下のようなものである。

(a) area address + ID + SEL
(b) network number + host number

OSI [?] ではネットワークレイヤ・アドレスは (a) のようなフォーマットを持つ。ここで area address はルーティング・ドメイン、ID はそのドメイン内でユニークな番号、SEL はパケットを受け取る上位レイヤのエンティティを表すセレクタである。XNS [?] や DARPA-IP [?] ではホストアドレスは (b) のようなフォーマットを持つ。XNS では host number はシステムを通じて一意であり、network number はルーティングを容易にするために付加されている。DARPA-IP では network number はシステムを通じて一意であり、host number はそのネットワークの中で一意である。このように、一般的にホストアドレスは 2 階層の階層構造を持っていると仮定できる。このようなホストアドレスを規定しているネットワークでは、ホストが他のネットワークに移動すればそのホストのアドレスも変化する。現状ではアプリケーション・エンティティはホストアドレスの変化を認識しなければならないので、ホスト移動透過性は実現されていない。

本研究ではネットワークレイヤでホスト移動透過性を実現するために「仮想ネットワーク」という概念を導入する。仮想ネットワークとは現実のネットワークの上に存在する論理的なネットワークである。現実のネットワークを以後「物理ネットワーク」と呼ぶことにする。仮想ネットワークは相互接続された論理的なネットワークを物理ネットワーク上に形成する。トランスポートレイヤからは仮想ネットワークの世界のみが見えるようになる。各ホストはそれぞれ 1 つの物理ネットワークに接続されると同時に 1 つの仮想ネットワークにも接続される。物理ネットワーク上でホストが移動しても、仮想ネットワークの世界ではホストは移動しない。すなわち、各ホストは常に 1 つの仮想ネットワークに接続されていると考えるのである。このようなネットワークを、そのホストの「ホームネットワーク」と名付ける。各ホストは物理ネットワークにおいてネットワークアドレス (これは「物理ネットワークアドレス」(PN-アドレス) とも呼べる) を持つが、これと同時に仮想ネットワークにおいては「仮想ネットワークアドレス」(VN-アドレス)

をも持つ。ホストは仮想ネットワーク上では移動しないので、たとえホストが物理ネットワーク上で移動してもそのホストの VN-アドレスは変化しない。ホストの PN-アドレスはそのホストの物理ネットワーク上での位置を表し、パケットの経路制御に利用される。トランスポートレイヤからは相手ホストがどこに移動しても VN-アドレスで相手を指定することができるようになる。基本的には PN-アドレスはトランスポートレイヤからは見えない。VN-アドレスと PN-アドレスは同じフォーマットを持つ。

ホストの仮想ネットワークアドレスと物理ネットワークアドレスの関係は、仮想メモリシステムにおける仮想アドレスと物理アドレスの関係と似ている。仮想メモリシステムにおいては、プログラムが物理メモリのどこに配置されようともそのプログラム中で使われている仮想アドレスは変化しない。同様に物理ネットワーク上でホストがどこに移動してもそのホストの仮想ネットワークアドレスは変化しない。

3.2 プロトコル階層

トランスポートレイヤは相手ホストを VN-アドレスで指定するので、パケットを配送するには VN-アドレスを対応する PN-アドレスに変換する必要がある。ネットワークレイヤはホスト間通信サービスを提供しており、これはパケットの中継、経路制御等のファンクションで実現されている。仮想ネットワークのためのアドレス変換はこのようなネットワークレイヤの基本的なファンクションとは異質であるので、本研究ではネットワークレイヤを以下に示す 2 つのサブレイヤに分割する (図 3.1. (a) 参照)。

- 「仮想ネットワーク・サブレイヤ」(VN-サブレイヤ): 物理ネットワーク上でのホストの位置を隠蔽する。ホストには仮想ネットワークアドレス (VN-アドレス) が割り当てられる。VN-アドレスは対応する物理ネットワークアドレス (PN-アドレス) に変換される。

- 「物理ネットワーク・サブレイヤ」(PN-サブレイヤ): 従来のネットワークレイヤである。OSI モデルに従えば、このサブレイヤは subnetwork independent sublayer, subnetwork dependent sublayer, subnetwork access sublayer の 3 つのサブレイヤからなる。このサブレイヤはパケットの中継、経路制御情報の交換等を行なう。ホストには物理ネットワークアドレス (PN-アドレス) が割り当てられる。

この分割により、(物理) ネットワークレイヤ・ヘッダとトランスポートレイヤ・ヘッダの間に VN-サブレイヤ・ヘッダが追加される (図 3.1. (b) 参照)。もし受信したパケットの (物理) ネットワークレイヤ・ヘッダが、上位レイヤとして VN-サブレイヤを指定していれば、このパケットは VN-サブレイヤに渡される。そうでなければ VN-サブレイヤはバイパスされ、そのパケットはトランスポートレイヤに渡される。このように、受信されたパケットが VN-サブレイヤ・ヘッダを持っていなかったり受信したホストが VN-サブレイヤを実装していない場合は、VN-サブレイヤをバイパスすることが可能である。

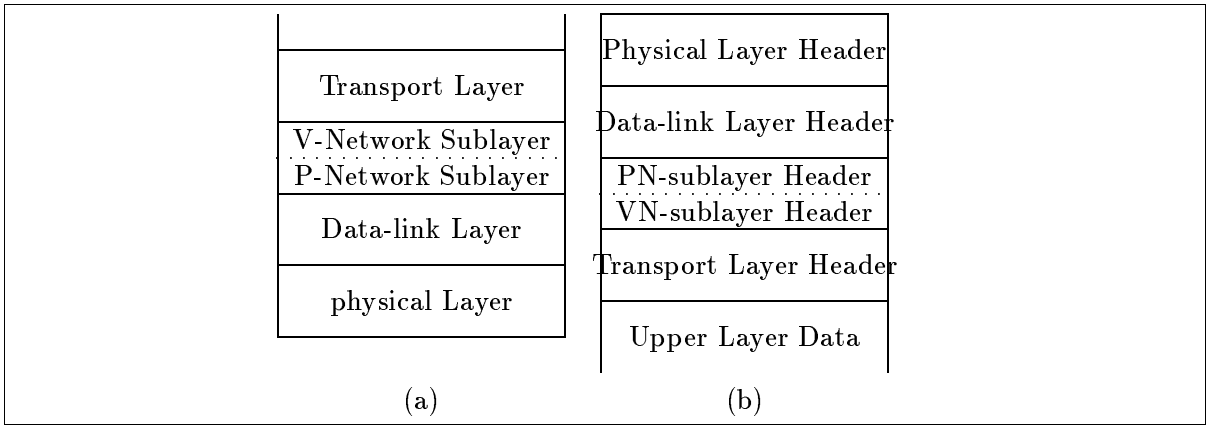

図 3.1: プロトコル階層とパケットフォーマット

第 4 章

仮想ネットワークプロトコル (VNP)

この章では特定のネットワークレイヤ・プロトコルに依存しない仮想ネットワーク・サブレイヤの一般的なプロトコル (VNP) の概略を述べる。VNP はパスカル風な言語と状態遷移図で定義されている [?]。第 3.2 章で述べたように、VN-サブレイヤの主要なサービスはアドレス変換である。以下、PN-サブレイヤについて次のことを仮定する。PN-サブレイヤは従来のコネクションレス・モードのネットワークレイヤであり、経路制御はホストの PN-アドレスを用いて PN-サブレイヤで行なわれている。また従来のエラーコントロールも PN-サブレイヤで行なわれている。

4.1 拡散キャッシュ法

アドレス変換にはいくつかの可能な方法が考えられる。たとえば変換サーバ法、同報通信法である。変換サーバ法の場合、送信側ホストの VN-サブレイヤが送信前に変換サーバに問い合わせを行ない、相手ホストの VN-アドレスを PN-アドレスに変換する。同報通信法の場合、送信側ホストの VN-サブレイヤが相手ホストの VN-アドレスを同報通信し、相手ホストがこのパケットを受信すると自分の PN-アドレスを応答する。しかしこれらの方法は、第 2.2 章で述べたネームサーバ法や同報通信法と同様な欠点がある。

本研究では、アドレス変換のオーバーヘッドをなるべく小さくするために「拡散キャッシュ法」を導入する。この方法では、各ゲートウェイやホスト (OSI の用語では IS/ES) はアドレス変換のためのキャッシュを保持する。送信側ホストが受信側ホストの PN-アドレスを知らない場合、送信側ホストは相手ホストがホームネットワークに接続されていると仮定する。すなわち送信側は受信側ホストの PN-アドレスが VN-アドレスと等しいと仮定するのである。送信されたパケットは受信側ホストのホームネットワークに向かって転送されていく。経路上のゲートウェイが受信側ホストに関するキャッシュを保持していればそのゲートウェイでアドレス変換が行なわれ、パケットは受信側ホストの物理的な位置にフォワードされる。

基本的にホストの VN-アドレスと PN-アドレスの関係は、そのホストのホームネットワーク内のゲートウェイ (「ホームゲートウェイ」と呼ばれる) で管理される。最悪の場合、移動ホストに対して送信されたパケットはそのホストのホームネットワークでフォワードされる。しかし通信が行なわれるにつれてキャッシュの情報が他のゲートウェイ

やホストにも拡散していくため、多くのパケットは送信ホストや経路上のゲートウェイでフォワードされることになる (第 4.4 章参照)。すなわちあるホストが移動ホストからパケットを受信すれば、そのホストは移動ホストの PN-アドレスを知ることになり、それ以降は移動ホストに直接パケットを送信することができるようになるのである。キャッシュ情報の拡散は次の 2 つの事象による。1 つは VN-サブレイヤのコントロールパケット交信、もう 1 つは通常のデータパケット交信である。コントロールパケットはホストがネットワークに接続されたり切断されたりするときに送信される。送信側ホストの仮想および物理ネットワークアドレスはどちらのパケットタイプのヘッダにも含まれているので、ゲートウェイはパケットを中継するときにヘッダの内容を盗み見て自分のローカルキャッシュを更新するのである。

4.2　AMT とヘッダフォーマット

各ゲートウェイやホストで保持されるキャッシュを「アドレス変換テーブル (AMT: Address Mapping Table)」と呼ぶ。ゲートウェイやホストがパケットを受信したとき、送信側ホストの PN-アドレスが VN-アドレスと異なっていると送信側ホストに関する AMT エントリが作成または更新される。AMT エントリは次の 4 つのフィールドからなる。

- 仮想ネットワークアドレス (AMT_{VnAddr}): エントリに対するキーの役割を果たす。
- 物理ネットワークアドレス (AMT_{PnAddr}): 求めたい値。
- アドレス・タイムスタンプ (AMT_{AddrTS}): このエントリが無効かどうかを判断するためのタイムスタンプ。
- アイドル時間 (AMT_{idle}): 使用されていないエントリを消去するためのタイマ。

ネットワークレイヤは 2 つのサブレイヤに分割されているため、ネットワークレイヤ・ヘッダは VN-サブレイヤ・ヘッダと PN-サブレイヤ・ヘッダの両方を含む。VN-サブレイヤ・ヘッダに必須のフィールドを以下に示す。

- 送信側および受信側の仮想ネットワークアドレス ($VN_{SrcAddr}$ および $VN_{DstAddr}$)。
- パケットタイプ (VN_{type})。パケットがデータパケットかコントロールパケットかを示す。
- 送信側アドレス・タイムスタンプ (VN_{SrcTS})。送信側ホストは送信時の時刻をこのフィールドにセットする。このフィールドの値は、送信側ホストに関する AMT エントリの AMT_{AddrTS} フィールドにセットされる。
- 受信側アドレス・タイムスタンプ (VN_{DstTS})。受信側ホストの VN-アドレスが PN-アドレスに変換されたときにこのフィールドの値は設定または更新される。パケット中継時、このフィールドの値が受信側ホストに関する AMT エントリの AMT_{AddrTS}

フィールドの値より古い場合は、PN-サブレイヤ・ヘッダの PN-アドレス・フィールドは AMT エントリに保持されていた値で上書きされる。

受信側ホストに関する AMT エントリが無効になっているかを判断するため、2 つのタイムスタンプ、すなわちパケットの VN_{DstTS} フィールドと AMT エントリの AMT_{AddrTS} フィールド、が比較される。これはシステムに含まれるすべてのホストでクロックの同期が必要であることを求めてはいない。なぜなら、これら 2 つのタイムスタンプは同一のホストのクロックから発生しているからである。システムに含まれる各ホストのクロックが単調に増加しさえすれば、AMT が無効かどうかのチェックは正しく行なわれる。

4.3 通信手続き

送信、中継、受信の手続きは以下の通りである。送信側ホストが受信側ホストに関する AMT エントリを保持していた場合、送信側ホストは AMT エントリに保持されている値をパケットの PN-アドレス・フィールドにセットする。そうでない場合、送信側ホストは受信側ホストの PN-アドレスが VN-アドレスに等しいと仮定する。送信されたパケットはゲートウェイをホップしていく。各ゲートウェイでは、必要であれば送信側ホストに関する AMT エントリが作成または変更され、また必要であれば受信側ホストに関する AMT エントリに従って、パケットヘッダ中の受信側ホストの PN-アドレスが修正され、パケットが中継される。受信側ホストでは、送信側ホストに関する AMT エントリが作成または変更される。

4.4 ホスト移動

図 4.1 にホスト移動時のパケットの流れを示す。図において、楕円はサブネットワーク、小さな円はホスト、正方形はゲートウェイを表す。図では *Host-X* が *Net-A* から *Net-G* へ移動している。ここで *Host-X* のホームネットワークは *Net-A* と仮定する。この移動により *Host-X* の PN-アドレスは変化するが、VN-アドレスは不変である。移動後[1]*Host-X* は *ConnNotify* パケットを自分のホームネットワークである *Net-A* へ送信する。このパケットは VN-アドレスと新しい PN-アドレスを含んでおり、*Net-G*、*Net-C* および *Net-B* を通過するので、*Gw-CG*、*Gw-BC* および *Gw-AB* は *Host-X* の新しい PN-アドレスを知り、AMT エントリを作成または更新する。パケットは最終的には *Net-A* に到着し、*Net-A* 内の各ホストに *Host-X* の PN-アドレスを知らせるため、*Net-A* 内で同報通信される。*Gw-AB* は *Host-X* に確認応答 (ack) パケットを返送する。図で×印のついたゲートウェイは *Host-X* に関する AMT エントリを保持しているものであることを示す。

パケットのフォワード手続きを図 4.2 に示す。図において *Host-Y* は *Host-X* の AMT エントリを保持いていないものとする。従って *Host-Y* はパケットの PN-アドレスフィー

[1]ホストがネットワークから切断されるときの手続きは後述。

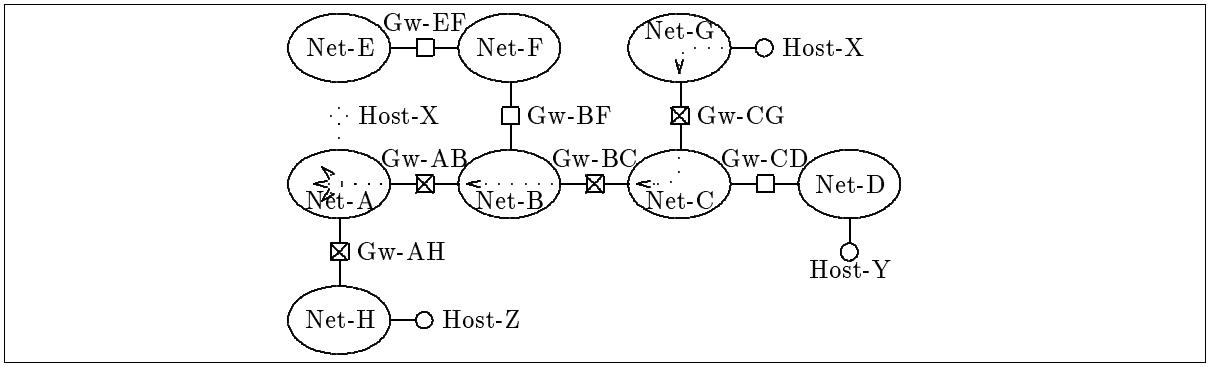

図 4.1: ネットワークへの接続

ルドに *Host-X* の VN-アドレスをセットする。*Host-X* のホームネットワークは *Net-A* であるので、パケットは *Net-A* へ向かう。パケットが *Gw-BC* に達したとき *Gw-BC* は *Host-X* の VN-アドレスを対応する PN-アドレスに変換する。従ってパケットは *Host-X* へフォワードされる。同様に *Host-Z* が *Host-X* の新しい PN-アドレスを知らなければ、パケットは *Net-A* に到達後、*Host-X* にフォワードされる。図において、細い破線は目的ホストの PN-アドレスが正しくセットされていないパケットの軌跡を示し、太い破線は目的ホストの PN-アドレスが正しくセットされているパケットの軌跡を表す。*Host-X* が *Host-Y* にパケットを送れば、*Host-Y* と同様に *Gw-CD* も *Host-X* の PN-アドレスを知ることになる。このように *Host-X* と他のホスト間での通信が行なわれるにつれて、*Host-X* に関するキャッシュ情報がより広く拡散していく。AMT エントリと VN-サブレイヤヘッダはタイムスタンプ・フィールドを持っているので、無効になった AMT エントリが目的ホストの VN-アドレスを無効になった PN-アドレスに変換することは無く、パケットの経路のループは発生しない。

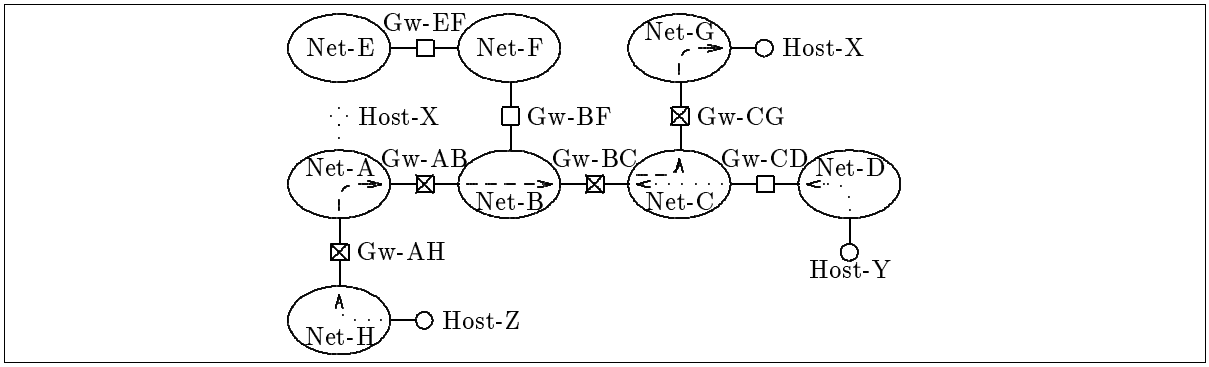

図 4.2: パケットのフォワーディング

図 4.3 にホストがネットワークから切断されるときのパケットの流れを示す。図において *Host-X* は *Net-G* から切断されようとしている。このとき *Host-X* は *DiscNotify* パケットを *Net-G* 内に同報通信する。このパケットは *Host-X* の VN-アドレスを含む。*Gw-CG* がこのパケットを受信すると、*Gw-CG* は *Host-X* に関する AMT エントリを消

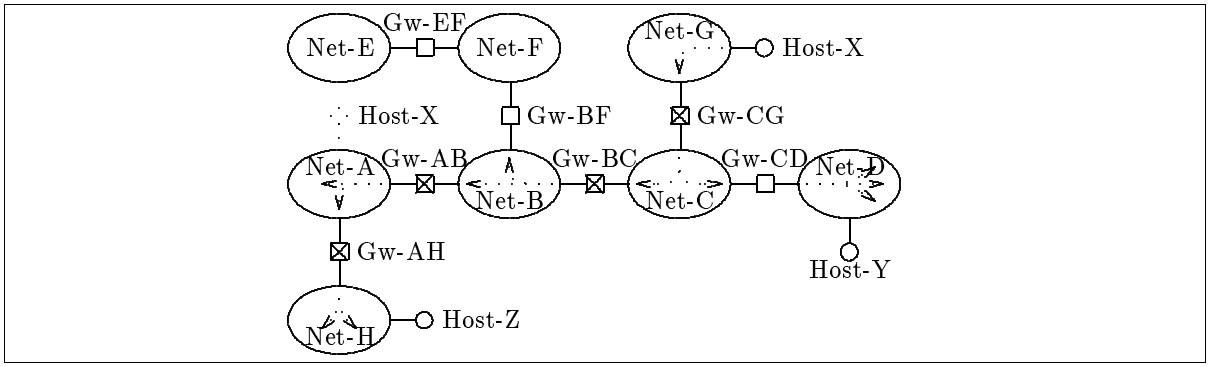

図 4.3: ネットワークからの切断

去し、さらに *Gw-CG* は *Host-X* に関する AMT エントリを持っていたので、*Net-C* 側にこのパケットを同報通信する。同様に *Gw-BC* もこのパケットを *Net-B* 内に同報通信する。しかし、*Gw-BF* は *Host-X* に関する AMT エントリを持っていないので、このパケットを受信しても何もしない。このように大部分の *Host-X* に関する AMT エントリは *Host-X* がネットワークから切断されるときに消去される。*DiscNotify* パケットは関係の無いネットワークには伝わらない。

第 5 章

VIP の設計

仮想ネットワークの概念と拡散キャッシュ法をインターネットプロトコル (IP) に適用して設計されたのが仮想インターネットプロトコル (VIP) である。本章では VIP の詳細について述べる。

5.1 IP アドレスと VIP アドレス

仮想ネットワークの概念に従うと、ネットワークレイヤにおいてホストは VN-addr と PN-addr という 2 つのアドレスを持つことになる。IP アドレスはインターネットでのホストの位置を示しているので、PN-addr と考えることができる。そこで VN-addr として VIP アドレスを導入し、従来の IP レイヤを VIP サブレイヤと IP サブレイヤという 2 つのサブレイヤに分割する。ホストには VIP サブレイヤにおいて VIP アドレスが、IP サブレイヤにおいて IP アドレスが与えられる。たとえホストが物理的に他のサブネットワークへ移動しても VIP アドレスは決して変化しないので、TCP/UDP レイヤやアプリケーションは VIP アドレスで相手ホストを一意に指定することができる。VIP アドレスと IP アドレスは同じフォーマットを持っており、ホストがホームネットワークに存在するときは、これらは同じ値を持つ。基本的に IP アドレスは TCP/UDP レイヤからは見えなくなっている。

IP アドレスは位置に依存するので、ホストが他のサブネットワークに移動すると IP アドレスは変化する。すなわち、ホストがあるサブネットワークに移動するとそこで新しい IP アドレスを割り当ててもらわなくてはならない。VIP 自身にはこのような IP アドレスの割り当て・解放といった機能はない。このような機能を実現するため、本研究では *migd* および *vipd* という 2 つのデーモンプロセスを実装している。紙面の都合上、これらのデーモンプロセスについては述べないこととする。

5.2 パケットおよび AMT のフォーマット

VIP は IP のオプションとして実装することができる。IP と VIP のヘッダフォーマットを図 5.1 に示す。図において上半分は通常の IP ヘッダであり、下半分が IP のオプションとしての VIP ヘッダである。VIP ヘッダは以下に示す 8 つのフィールドから成って

いる。

- オプションタイプ ($VIP_{OptType}$) とオプション長 (VIP_{OptLen}): これらの 2 つのフィールドは IP オプションのため IP で規定されている [?]。VIP_{OptLen} の値、すなわち VIP ヘッダ長は 20 バイトである。

- VIP タイプ (VIP_{Type}): VIP パケットには以下に示す 6 つのタイプがある。

  - *VipData*: 通常のデータパケット。
  - *VipConn*: 送信ホストがサブネットワークに接続されたことを通知するパケット。
  - *VipConnAck*: *VipConn* パケットの確認応答パケット。
  - *VipDisc*: 送信ホストがこれからサブネットワークから切断されることを通知するパケット。
  - *VipDelAmt*: AMT エントリが無効になったことを通知するパケット。
  - *VipErrObs*: 無効な AMT エントリが見つかったことを知らせるエラーパケット。

- 保持時間 ($VIIP_{hold}$): 送信ホストのための AMT エントリを保持しておく時間を指定する。

- 送信側 VIP アドレス ($VIP_{SrcAddr}$) および受信側 VIP アドレス ($VIP_{DstAddr}$)

- 送信側アドレスタイムスタンプ (VIP_{SrcTS}): 送信ホストの VIP/IP アドレスペアのバージョン番号。

- 受信側アドレスタイムスタンプ (VIP_{DstTS}): 受信ホストの VIP/IP アドレスペアのバージョン番号。

VIP のための AMT (Address Mapping Table) エントリは 4 つのデータフィールドとさらに 1 つの管理用のフィールドから成っている。それぞれのフィールドは 4 バイトであり、エントリ全体は 20 バイトである。4 つのデータフィールドを以下に示す。

- VIP アドレス (AMT_{VIP}): このエントリのキーとして働く。

- IP アドレス (AMT_{IP}): 求める値。

- アドレスタイムスタンプ (AMT_{AddrTS}): このエントリの VIP/IP アドレスペアのバージョン番号を示す。

- アイドルタイマー (AMT_{idle}): タイムアウトのためのフィールド。一定時間アクセスされないエントリは消去される。

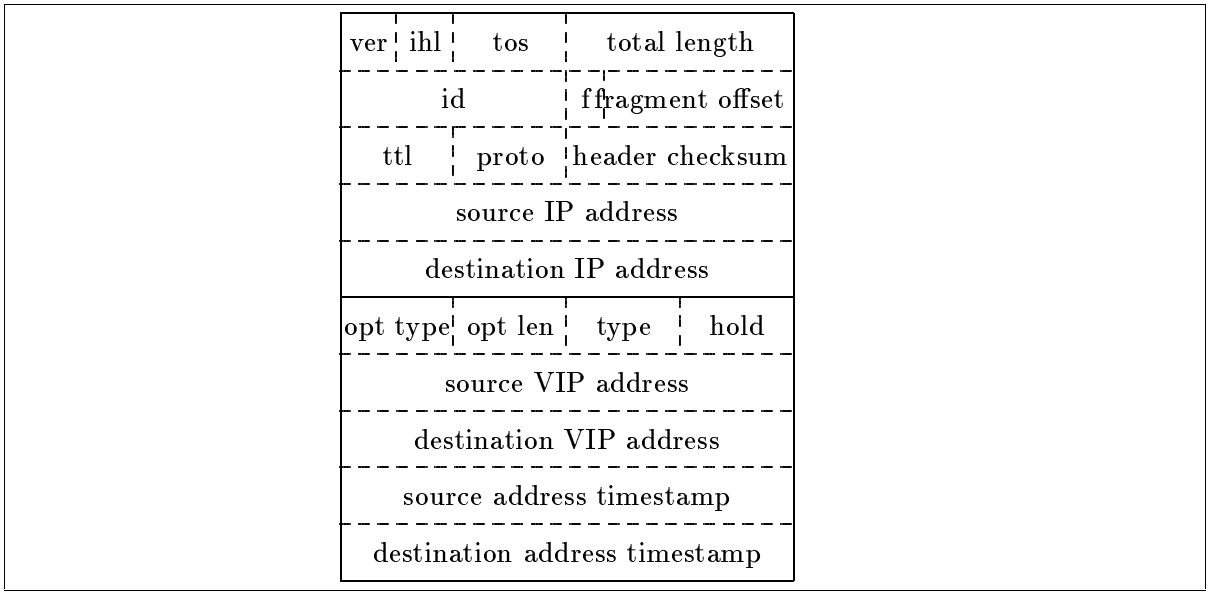

図 5.1: VIP ヘッダフォーマット

5.3　通信手順

パケット送信の際、TCP/UDP レイヤが相手ホストの VIP アドレスを指定して VIP サブレイヤに送信要求を出す。すると VIP は VIP ヘッダを作成し、各フィールドを設定する。次に VIP サブレイヤは相手ホストのための AMT エントリを検索する。エントリが見つかると、VIP サブレイヤは相手ホストの VIP アドレスを対応する IP アドレスに変換する。見つからない場合は、VIP サブレイヤは相手ホストがホームネットワークに存在すると仮定し、相手の IP アドレスが VIP アドレスと等しいとみなす。IP サブレイヤの処理は従来の IP レイヤの処理と同じである。最終的に IP サブレイヤはネットワーク・インターフェイス・レイヤに送信要求を出す。

パケット受信の場合には、IP の処理にいくつかの変更が必要になる。IP サブレイヤは VIP サブレイヤに存在する以下の 2 つのファンクションを呼び出さなくてはならない。

- *(f-1)*: 受信パケットの送信ホストのための AMT エントリを作成または更新するファンクション。送信ホストのための AMT エントリが存在せず、さらに送信ホストの IP アドレスと VIP アドレスが異なる場合はエントリが作成される。エントリが既に存在する場合、以下の 2 つの条件のうち 1 つが満たされるとエントリの更新が行なわれる。1) 送信ホストの IP アドレスが変化した場合、2) 受信パケットの VIP_{SrcTS} フィールドの値が AMT エントリの AMT_{AddrTS} よりも古い場合。

- *(f-2)*: 必要な場合、中継するパケットの受信側VIP アドレスを対応する IP アドレスに変換するファンクション。受信ホストのための AMT エントリが存在し、パケットの VIP_{DstTS} フィールドの値がエントリの AMT_{AddrTS} の値よりも古い場合、このファンクションは AMT_{IP} および AMT_{AddrTS} フィールドの値を返す。

ネットワーク・インターフェイス・レイヤが IP サブレイヤに対してパケット受信の信号を送ると、IP サブレイヤは *(f-1)* ファンクションを呼び出し、その後受信パケットの受信側 IP アドレスフィールドの値の自分の IP アドレスの比較を行なう。これら 2 つの IP アドレスが一致すると、IP サブレイヤは VIP サブレイヤにパケットを渡す。そうでない場合 IP サブレイヤは *(f-2)* ファンクションを呼び出す。アドレス変換が行なわれると IP サブレイヤは IP ヘッダの相手側 IP アドレスフィールドを更新し、VIP サブレイヤは VIP ヘッダの VIP_{DstTS} フィールドを更新する。そしてパケットは次のゲートウェイまたは受信ホストへ中継される。

5.4 ホスト移動手順

移動ホストがサブネットワークに接続されると、このホストはまず *VipConn* パケットをホームネットワークへ送信する。この *VipConn* パケットはゲートウェイで次々に中継されていく。*VipConn* パケットは送信ホストの VIP および IP アドレスを保持しているため、途中のゲートウェイは送信ホストのための AMT エントリを作成することができる。移動ホストのホームネットワークに存在するゲートウェイをホームゲートウェイと呼ぶが、ホームゲートウェイが *VipConn* パケットを受信すると移動ホストのための AMT エントリを作成し、*VipConnAck* パケットを返送する。ホームゲートウェイはこのエントリをタイムアウトでは決して消去しないことに注意されたい。またホームネットワークがイーサーネットのような同報通信型ネットワークの場合、ホームゲートウェイはホームネットワークに *VipConn* パケットを同報通信する。加えて移動ホストに対する ARP (Address Resolution Protocol) リクエストに応答するため、ホームゲートウェイは ARP テーブルに移動ホストのためのエントリを作成する。

移動ホストへのパケット送信の場合、送信ホストが受信ホストのための AMT エントリを保持していないと送信ホストは受信ホストの IP アドレスと VIP アドレスは等しいとみなしてパケットを送信する。パケットは受信ホストのホームネットワークに向かって転送されていくが、途中のゲートウェイが受信ホストのための AMT エントリを保持しているとそのゲートウェイでアドレス変換が行なわれ、受信ホストの実際の位置へパケットを転送する。受信ホストが送信ホストへパケットを返送すると、送信ホストも受信ホストのための AMT エントリを作成するので、次の送信からは送信時にアドレス変換を行なうことができる。

移動ホストはサブネットワークから切断されるときには *VipDisc* パケットをホームゲートウェイに送信する。ホームゲートウェイが *VipDisc* パケットを受信すると自分が接続されているすべてのサブネットワークに対して *VipDelAmt* パケットを同報通信する。*VipDelAmt* パケットを受信したゲートウェイは、その移動ホストのための AMT エントリを保持していればそれを消去し、他のすべてのサブネットワークに *VipDelAmt* を同報通信する。その移動ホストのための AMT エントリを保持していない場合は何も行なわない。このように移動ホストがサブネットワークから切断される際、ほとんどの AMT エントリは消去される。

5.5　議論

[**フラグメンテーションとリアセンブリ**]　IP データグラムがデータリンクの最大送信長よりも大きいとフラグメンテーションが行なわれる場合がある。このとき、IP オプションとしての VIP ヘッダは各フラグメントにコピーされなければならない。また従来の IP では、各フラグメントは送信側および受信側の IP アドレスと ID フィールドという三つ組で識別されるが、VIP の場合は IP アドレスの代わりに VIP アドレスが用いられる。

[**IP オプションとの整合性**]　IP オプションには 8 つの種類があるが、Loose/Strict Source Routing を除いて VIP とは整合性がある。もともとソースルーティングという考え方は送信ホストが相手ホストの物理的な位置のみならず、相互に接続されたネットワークの構成も知っていることを前提とする。従って移動透過性という考え方とは両立しないものである。このように VIP の環境では Loose/Strict Source Routing は適用できない。

[**ICMP Redirect**]　ICMP [?] には IP データグラムの送信ホストにコントロールメッセージを知らせるため、11 のメッセージが定義されている。ゲートウェイがパケットを中継する際、送信するネットワーク・インターフェイスと受信したネットワーク・インターフェイスが同一の場合、送信ホストに ICMP Redirect メッセージが返される。しかし VIP の場合は、アドレス変換を行なったゲートウェイは受信したネットワーク・インターフェイスに再びパケットを送出することもあり得る。これは正しい処理であり、ICMP redirect を発生させてはならない。

[**拡張性**]　IP ネットワークはさらに規模が大きくなり、移動ホストの数も増加し続けると考えられる。従って移動ホストをサポートするプロトコルはこれらの要素に対して拡張性がなければならない。VIP はアドレス変換のために AMT と呼ばれるキャッシュを利用しているが、ホストやゲートウェイが持たなければならない AMT エントリの数はネットワークの規模や移動ホストの総数には依存しない。ホストは同時に通信するホストの数だけの AMT エントリを保持すれば十分である。ホストは通信しないホストのための AMT エントリは作成しない。またゲートウェイは自分自身をホームゲートウェイとしているホストの数だけの AMT エントリを保持すれば良い。ゲートウェイは中継するパケットに対するアドレス変換のための AMT エントリを保持することもあるが、このようなエントリは失われても良い。なぜなら、移動ホストのための AMT エントリはその移動ホストのホームゲートウェイが必ず管理しているからである。現在の実装では、ホストは 100 個の AMT エントリを持ち、ゲートウェイは 2000 個の AMT エントリを持つ。これらが占めるメモリ領域はホストでは 2 キロバイト、ゲートウェイでは 40 キロバイトと大変少ない。

[**AMT エントリの正当性**]　AMT エントリの正当性はタイムスタンプ機構で保たれている。すなわち受信ホストのアドレス変換においては AMT_{AddrTS} フィールドと VIP ヘッ

ダの VIP_{DstTS} フィールドの値が比較され、AMT エントリの更新の際は AMT_{AddrTS} フィールドと VIP_{SrcTS} フィールドが比較される。しかし VIP はグローバルクロックを必要とはしない。なぜならば、比較されるタイムスタンプは常に同一のホストが発生したものだからである。従って各ホストのクロックが単調に増加すれば、タイムスタンプの比較は正しく行なわれる。ホストやゲートウェイはあるホスト (たとえば *Host-A*) からパケットを受信したとき、そのホストのための AMT エントリを作成する。従って *Host-A* のための AMT エントリの AMT_{AddrTS} フィールドは *Host-A* が発生したものである。*Host-A* に向かうパケットに対するアドレス変換において、パケットの VIP_{DstTS} フィールドには *Host-A* のための AMT エントリの AMT_{AddrTS} フィールドの値が代入される。このように比較される 2 つのアドレススタンプは常に同じホスト (この例では *Host-A*) が発生したものである。AMT エントリの更新の場合、AMT エントリの AMT_{AddrTS} と VIP パケットの VIP_{SrcTS} の値が同じホストが発生したものであることは明白である。

[**無効 AMT エントリの影響**] ある移動ホストのための *VipDelAmt* パケットを受信するとホストやゲートウェイはその移動ホストのための AMT エントリを消去する。さらにある一定の時間、移動ホストからパケットを受信しないとその移動ホストのための AMT エントリは消去される。しかし無効な AMT エントリがホストやゲートウェイに残される可能性もある。このようなエントリはパケットを誤ったホストに配送するかもしれない。この場合、受信したホストの VIP アドレスと配送されたパケットの受信側 VIP アドレスの値が一致しないので、パケットは捨てられて *VipErrObs* パケットが送信ホストに返送される。*VipErrObs* パケットは転送路に存在する無効な AMT エントリを消去していく。このように無効な AMT エントリは次のメソッドにより消去あるいは更新される。1) タイムアウト、2) データパケット、3) *VipConn* パケット、4) *VipDelAmt* パケット、5) *VipErrObs* パケットである。

[**ネットワーク切断に対する耐故障性**] 最悪の場合、移動ホストに対するパケットはそのホームゲートウェイが転送することになる。従って移動ホストの現在の位置とそのホームネットワークを分離するようなネットワークの切断が発生すると問題が生じる可能性がある。2 つの場合が考えられる。移動ホストがネットワークに接続される前にネットワークが切断されるか、後に切断されるかである。後者の場合移動ホストは既に接続手順を実行しているので、この移動ホストのための AMT エントリを保持するゲートウェイがネットワーク上に存在することになる。前者の場合、移動ホストが送信した *VipConn* パケットはホームゲートウェイに届かない。このような場合、移動ホストは *VipConnAck* パケットを受信するまで *VipConn* パケットを再送信するが、このパケットはホームゲートウェイに向かうパス上のいくつかのゲートウェイまでは届くことも考えられる。またどちらの場合も移動ホストがあるホストに対して送信すれば、その移動ホストのための AMT エントリも広まることになる。このように VIP は、ネットワークの切断に対してある程度の耐故障性を持つことがわかる。

第 6 章

VIP の実装

6.1 内部構造

現在 VIP は SONY NEWS ワークステーションで稼働している。VIP は 4.3BSD UNIX に基づく NEWS-OS4.1 のカーネルを改造して実装されている。変更したステップ数は約 800 である。図 6.1 に IP レイヤのブロックダイアグラムを示す。図において実線で示された矩形は元々 IP レイヤに実装されていたファンクションを、破線で示された矩形は VIP を実装するために加えられたファンクションを示す。加えられたファンクションの機能を以下に示す。

- *vip_output*: TCP/UDP レイヤが送信要求を出したときに VIP ヘッダを生成する。同時に受信ホストのための AMT エントリの検索も行なう。もしエントリが見つかれば受信ホストの VIP アドレスは IP アドレスに変換され、IP ヘッダの受信側 IP アドレスフィールドに設定される。

- *vip_input*: 受信した VIP パケットのヘッダのチェックを行なう。

- *vip_modamt*: 受信した VIP パケットの送信ホストのための AMT エントリの作成・更新を行なう。

- *vip_mapdst*: 中継するパケットの受信ホストのための AMT エントリを検索する。エントリが見つかり、パケットの VIP_{DstTS} フィールドの値がエントリの AMT_{AddrTS} フィールドの値より古い場合、IP ヘッダの受信側 IP アドレスフィールドを AMT エントリのに保持されている値で置き換える。さらに VIP ヘッダの VIP_{DstTS} フィールドを AMT エントリの AMT_{AddrTS} フィールドの値で置き換える。

- *vip_cntl*: 接続・切断手順を実行する。たとえば *VipConn* パケットや *VipDisc* パケットの送信である。

- AMT ファンクション群: AMT を操作するファンクション群。たとえば AMT エントリの割り当て・解放、検索などである。

AMT エントリの効率的な検索のためハッシュ関数を用いているが、現在のハッシュ関数は大変単純であり、VIP アドレスのドット表記における 4 つの整数の和 (ハッシュテー

ブルサイズのモジュロ) となっている。新しい AMT エントリを登録する時は *AMT entry pool* からフリーなエントリを獲得し、ハッシュテーブルにつなげる。登録の際に衝突が発生したらそのエントリは既に存在しているエントリの後につなげられる。消去の際は AMT エントリはハッシュテーブルから除かれて *AMT entry pool* に戻される。

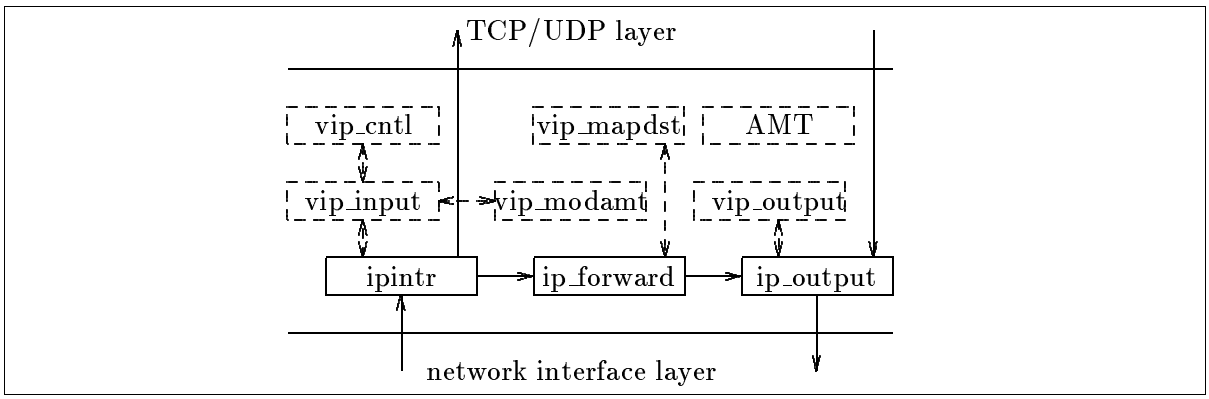

図 6.1: IP レイヤのブロックダイアグラム

6.2 VIP の処理手順

従来の IP では送信、中継、受信の処理は以下のように行なわれる。送信の際、TCP/UDP レイヤの送信要求は *ip_output* が受けとる。*ip_output* は IP ヘッダを生成し、ネットワーク・インターフェイス・レイヤに送信要求を出す。パケットが受信されるとネットワーク・インターフェイス・レイヤは *ipintr* に受信シグナルを送る。*ipintr* は受信したパケットの受信側 IP アドレスをチェックし、パケットが自局宛なら TCP/UDP レイヤに渡す。そうでない場合は *ip_forward* および *ip_output* を介してパケットを中継する。

VIP の場合、第 5.3、5.4 章で示した処理は以下のように行なわれる。

- データパケットの送信の場合、*ip_output* は TCP/UDP レイヤから送信要求を受けとり、*vip_output* を呼び出す。*vip_output* は VIP ヘッダを生成し、可能なら受信ホストの VIP アドレスを IP アドレスに変換する。最後に *ip_output* はネットワーク・インターフェイス・レイヤに送信要求を出す。

- データパケットの受信の場合、*ipintr* はネットワーク・インターフェイス・レイヤから受信シグナルを受けとり、*vip_input* を呼び出す。*vip_input* は *vip_modamt* を呼び出して送信ホストの AMT エントリの作成・更新を行なう。もしパケットが自局宛なら *ipintr* は TCP/UDP レイヤにパケットを渡す。そうでない場合は *ipintr* は *ip_forward* を呼び出す。*ip_forward* は *vip_mapdst* を呼び出して、必要なら受信ホストの VIP アドレスを IP アドレスに変換する。そしてパケットは *ip_output* を介して中継される。中継の際には *vip_output* は呼び出されないことに注意されたい。

- ホストがサブネットワークに接続されると、*vip_cntl* は *VipConn* パケットを生成して *ip_output* を呼び出す。そしてタイムアウトのためのタイマーをスタートさせ、*VipConnAck* パケットを待つ。*VipConn* パケットはホームゲートウェイに向かって転送されていくが、途中のゲートウェイでは通常のデータパケットと同様に扱われる。ホームゲートウェイが *VipConn* パケットを受信すると、*vip_cntl* はこのパケットをホームネットワークに同報通信し、*VipConnAck* パケットを送信ホストに返送する。

- ホストがサブネットワークから切断されるとき、*vip_cntl* は *VipDisc* パケットを生成し、*ip_output* を呼び出す。このパケットは移動ホストのホームゲートウェイに転送される。ホームゲートウェイがこのパケットを受信すると、移動ホストのための AMT エントリを消去し、*VipDelAmt* パケットをすべての接続しているサブネットワークに同報通信する。この移動ホストのための AMT エントリを保持しているゲートウェイがこのパケットを受信すると、*vip_cntl* はこのエントリを消去し、他のすべてのサブネットワークにこのパケットを同報通信する。この移動ホストのための AMT エントリを保持していないゲートウェイやホストはこのパケットを受信しても何も行なわない。

第 7 章

VIP の評価

7.1 実測した VIP のオーバーヘッド

VIP の性能評価は、6 台の SONY NEWS ワークステーション (移動ホスト 1 台、固定ホスト 2 台、ゲートウェイ 3 台)、3 本のイーサーネットおよび 1 本のシリアルラインからなるテスト環境で行なった。各 SONY NEWS ワークステーションは MC68030 1 つを CPU として持ち、4-8 メガバイトの主記憶容量を持つ。1 マイクロ秒まで測定できるタイマーボードを用いて IP および VIP+IP について送信、中継、受信に要求される時間を測定した。送信時間とは、*ip_output* が TCP/UDP レイヤから送信要求を受けとった瞬間から、*ip_output* がネットワーク・インターフェイス・レイヤに送信要求を出す瞬間までである。中継時間とは、*ipintr* がネットワーク・インターフェイス・レイヤから受信シグナルを受けとった瞬間から、*ip_output* がネットワーク・インターフェイス・レイヤに送信要求を出す瞬間までである。受信時間とは、*ipintr* がネットワーク・インターフェイス・レイヤから受信シグナルを受けとった瞬間から、TCP/UDP レイヤにパケットを渡す瞬間までである。表 7.1 に測定結果を示す。もしインターネット内の各ホストやゲートウェイの処理能力が SONY NEWS と同程度ならホスト間のオーバーヘッドはホップ数の関数として以下の式で表される。

$$\text{host-to-host overhead} = 86 \times N_{hop} + 170 \ (\mu sec)$$

IP に対する VIP のオーバーヘッドは、ホップ数が 5 のときは約 0.6 ミリ秒であり、ホップ数が 10 のときは約 1.0 ミリ秒となる。インターネットにおける IP の送信時間は 100 ミリ秒の単位であるから、VIP のオーバーヘッドは無視し得る。

7.2 オーバーヘッドの詳細

表 7.1 に示したように、VIP のオーバーヘッドは送信では 104 マイクロ秒 (130 %)、中継では 86 マイクロ秒 (29 %)、受信では 66 マイクロ秒 (99 %) である。中継のオーバーヘッドに比較して送信と受信のオーバーヘッドが大きくなっていることがわかる。この章ではこのアンバランスについて述べる。

表 7.2 から 7.4 に実測したオーバーヘッドの詳細を示す。送信においては *vip_output* ファンクションが 55 マイクロ秒を消費し、*ip_insertoptions* ファンクションが 42 マイクロ秒を

表 7.1: 処理時間の比較

	transmission	relay	reception
IP	80	296	67
VIP+IP	184	382	133
overhead	104	86	66

(μsec)

消費している。*ip_insertoptions* は IP レイヤに元々組み込まれているファンクションであり、IP オプションが指定されたとき、IP ヘッダに IP オプションを組み込む役割を果たす。通常の IP パケット送信では IP オプションは指定されないので、*ip_insertoptions* は呼び出されない。しかし VIP は IP オプションフィールドを利用しているので *ip_insertoptions* が呼び出され、送信オーバーヘッドの 40 % を消費してしまっている。送信オーバーヘッドの 53 % は *vip_output* で発生しているが、これは VIP ヘッダの生成、VIP ヘッダのためのバッファ領域の確保 (*MGET*) およびアドレス変換 (*amt_search*) という 3 つの主な機能によって占められている。

受信においては *ip_dooptions* ファンクションが 23 マイクロ秒を消費しており、*vip_modamt* ファンクションが 31 マイクロ秒を消費している。*ip_dooptionos* も元々 IP レイヤに組み込まれているファンクションであり、受信したパケットのオプションの処理を行なう。やはり通常の IP パケットはオプションを持たないので、このファンクションは呼び出されない。実際、*ip_dooptions* は VIP に対しては何の役割も果たさない。しかし VIP は IP オプションとして実装されているので、*ip_dooptions* は呼び出されてしまい、受信オーバーヘッドの 35 % を占めている。このように、ホスト間の VIP のオーバーヘッドは無視し得るほどであるが、効率の面からは現在の 4.3BSD の実装は VIP に対しては適切でないことがわかる。

表 7.2: パケット送信のオーバーヘッド

vip_output()	ip_insert options()	others	total
55	42	7	104

(μsec)

vip_output()				
header	MGET()	amt_search()	others	total
13	12	22	8	55

(μsec)

表 7.3: パケット中継のオーバーヘッド

ip_ dooptions()	vip_ modamt()	vip_ mapdst()	others	total
23	31	27	5	86

(μsec)

表 7.4: パケット受信のオーバーヘッド

ip_ dooptions()	vip_ modamt()	vip_ input()	others	total
23	31	10	2	66

(μsec)

第 8 章

まとめ

自動車のような移動体のための電話システムは既に利用可能になっている。また無線によるローカルエリアネットワークの構築も進みつつある。しかしこのようなシステムにおいては、無線の利用はサブネットワーク内に限られている。すなわち対応する技術はデータリンクレイヤのものである。近い将来大規模な相互接続された無線ネットワークが構築されることが予想されるが、このような環境ではホスト移動透過性が非常に重要になる。従って無線ネットワーク間を移動する mobile/portable ホストのための新しいネットワークアーキテクチャが必要とされる。

本研究では、OSI の 7 レイヤモデルに従えばホスト移動透過性はネットワークレイヤで提供すべきであると主張し、仮想ネットワークという概念を導入した。この概念により従来のネットワークレイヤは仮想ネットワークサブレイヤと物理ネットワークサブレイヤの 2 つに分割される。各ホストは位置や移動に依存しない識別子 (仮想ネットワークアドレス) と、位置や移動に依存する識別子 (物理ネットワークアドレス) という 2 つの識別子を持つ。パケットの転送を行なうには仮想ネットワークアドレスを物理ネットワークアドレスに変換しなければならないが、アドレス変換のオーバーヘッドを最小限に抑えるため本論文では拡散キャッシュ法を導入した。データリンクとして無線ネットワークが広範囲で利用可能になれば、他のネットワークに移動しているホストは移動中もコネクション・オリエンテッド・モードの通信を続けることができるようになる。仮想ネットワークの概念は DARPA-IP にも適応可能であり、Virtual Internet Protocol (VIP) の提案も行なった。VIP は現在 7 台のワークステーションと 4 本のイーサネットから成るテスト環境で動作している。実測の結果、VIP のオーバーヘッドは無視できるものであることがわかった。VIP は現在 (株) ソニーコンピュータサイエンス研究所で開発されているオブジェクト指向分散オペレーティングシステムである MuseOS [?] にも実装中である。