第 7 部

移動ノード

第 1 章

はじめに

近年、計算機は急激に小型化され、強力な処理性能を持つ計算機が持ち運べる大きさにまでなった。同時にネットワークも急速に拡大し、計算機を使用する上で必要不可欠なものになりつつある。このような環境下では、当然、各自が手にしている計算機からネットワークにアクセスしたいという要求が出てくる。しかし、現在のネットワークアーキテクチャでは、移動しながらネットワークにアクセスするような利用体系は考慮されていない。具体的に、手持ちの計算機からネットワークにアクセスするためには、アドレスの取得及びその設定、アドレスデータベースの再登録など様々な問題が山積している。しかも、これらの問題を解決出来たとしても、「計算機の移動透過性」を実現することは出来ない。計算機が移動した場合、その上で動作している、もしくはその計算機と通信を行なっているアプリケーションは移動した計算機を認識できなくなる。そのため、計算機環境は移動する前と異なったものになり、ユーザに負担をかけることにる。

このような問題を解決するための研究は、ここ数年、盛んに行なわれるようになってきた。現在、ネットワーク分割方式、パケット転送方式などが存在する。また、これらの方式が Internet Protocol (IP) を基盤としているのに対し、IBM 社では、一般的なネットワークアーキテクチャでのモデルを提案している [41]。

コロンビア大学が提案しているネットワーク分割方式 [42] は、学内における計算機の移動透過性を実現するために提案された方式で、基本的には、複数の物理ネットワークを一つの論理ネットワークとして扱うことにより、計算機の移動透過性を実現している。複数の物理ネットワークを一つの論理ネットワークとして取り扱う時に生じる矛盾を解決するためにこの方式では各物理ネットワークに 一つずつ MSR (Mobile Support Router) を導入している。移動ホストは基本的に一つの論理ネットワークの中だけしか移動できない。あるホスト A が移動ホスト B と通信を行なうためにはホスト A はホスト B のアドレスを設定したパケットを IP のプロトコルに従って送信する。送信されたパケットは IP の経路制御機構によってホスト B が存在する論理ネットワークに経路制御される。その後、このパケットは到着した物理ネットワークの MSR に渡る。MSR は、まず、ホスト B が同じ物理ネットワークに存在しているかを自分が保持しているデータベースに問い合わせる。もし、同じ物理ネットワークに存在していたらホスト B に対してパケットを転送する。同じ物理ネットワークに存在していなかったら、同じ論理ネットワーク上に存在する他の MSR に対して問い合わせをし、ホスト B が存在している物理ネットワーク上にある MSR からの応答を待つ。応答を受けとると、その MSR に対してパケッ

トをカプセル化した形で転送する。それを受けとった MSR はカプセルからもとのパケットを取り出してホスト B へ転送する。ここで、MSR が持っている同じ物理ネットワーク上に存在するホストのデータベースは移動ホストからの通知によって管理されている。この方式の利点は MSR 以外に特殊な機能を必要とせず、MSR を導入するだけで利用できる点である。逆に短所は、同じ論理ネットワーク上の全ての MSR へ問い合わせをするため、ネットワークの規模を大きくすることが難しい点である。

松下電器社ではパケットを順に転送していくパケット転送方式を提案している [43]。この方式では、ホストが移動可能なネットワークに一つずつ用意されている PFS (Packet Forwarding Server) と呼ばれるパケット転送ホストを用いてパケットを転送する。それぞれの移動ホストは 2 つのネットワークアドレスを保持している。一方は移動したネットワークのアドレスである一時アドレス、もう一方はホストに固有なホームアドレスである。移動ホストは移動した時に自分のホームネットワークの PFS に予め一時アドレスを登録しておく。あるホスト A が移動ホスト B と通信する場合、ホスト A が送信したパケットは移動ホスト B のホームネットワーク上の PFS によってホスト B に転送される。その後は、ホスト A が PFS からの通知によって移動ホスト B の一時アドレスを取得できた場合にはホスト A は一時アドレスを使って通信を行なう。このとき、移動ホスト B がさらに他のネットワークに移動したとする。移動ホスト B はホームネットワーク上の PFS に加えて、これまで接続されていたネットワーク上の PFS に新たな一時アドレスを登録する。これによりホスト A が送信したパケットは直前に接続されていたネットワークの PFS によって移動ホスト B の新たな位置に転送される。この方式の利点は、PFS が存在するネットワークなら何処へでも移動できる点である。逆に欠点は、パケットを転送しなければならないためネットワークの分断に弱い点である。

我々は、これまで、仮想ネットワークの概念 [44] を IP に適用した Virtual Internet Protocol (VIP) を提案してきた、これについては 2 章で詳しく述べる。この方式の特徴は IP に依存していない点である。仮想ネットワークという概念を定義しておき、あくまでそれを IP に適用している。このプロトコルには、まだ、幾つかの問題を含んでいる。この問題を解決することが今回の目的である。3 章で VIP の問題点を明らかにする。さらに、この問題を解決するための考察を 4 章で行ない。5 章では Enhanced Virtaul Internet Protocol (EVIP) の仕様をまとめ、6 章でその実装について説明する。7 章では実装した VIP の評価を行なう。最後に 9 章で締めくくる。

第 2 章

Virtual Internet Protocol

この章では我々が提案する仮想ネットワークの概念を IP に適用した Virtual Internet Protocol (VIP) [45] の仕様について説明する。

2.1 仮想ネットワーク

この節では Internet アーキテクチャの問題点をさらに明確にして Virtual Network の概念を導入する。

2.1.1 Internet アーキテクチャの問題点

IP 層は、上位プロトコル (UDP や TCP) から受けとったデータを、正しく通信相手まで配送する手段を規定している層である。IP 層の上位層は、 IP 層にデータを渡す時、同時に、そのデータを届けるべき通信相手を示す Internet アドレスを渡す。ここに問題が一つ存在する。

Internet アドレスは netid と hostid から形成されている。netid はネットワークの識別子であり、ホストの識別とは無関係である。Internet 全体を考えた場合、Internet アドレスはホストの識別子ではなくホストの位置情報として動作する。しかし、IP の上位プロトコルが IP 層に渡す Internet アドレスは、通信相手の位置情報としてではなく、通信相手の識別子として使われている。ホストの存在位置で通信相手を特定する方法は、十分に短い時間内では有効だが、長い時間を想定した場合、この方法で通信相手を特定することは出来ない。実際、現行の IP では相手の Internet アドレスが変わると、異なったホストとして認識され、通信相手の識別さえ出来なくなる。

通信相手を、その存在位置に依存することなく特定するためには、純粋な識別子で通信相手を指定しなければならない。しかし、現行の IP ではこのようなホスト識別子は存在せず、存在位置に依存しない相手の指定方法はない。

2.1.2 仮想ネットワークの概念

上記のような問題を踏まえて、我々はホスト識別子を新たに導入することにした。実際には図 2.1 に示す様に、これまで一つの層であったネットワーク層を、ホスト識別子

でネットワークを構成している仮想ネットワーク層と既存のネットワーク層の二つに分割する。仮想ネットワーク層では、上位にある層から通信相手の識別子を受けとり、さらに位置情報を付加して既存のネットワーク層に渡す。

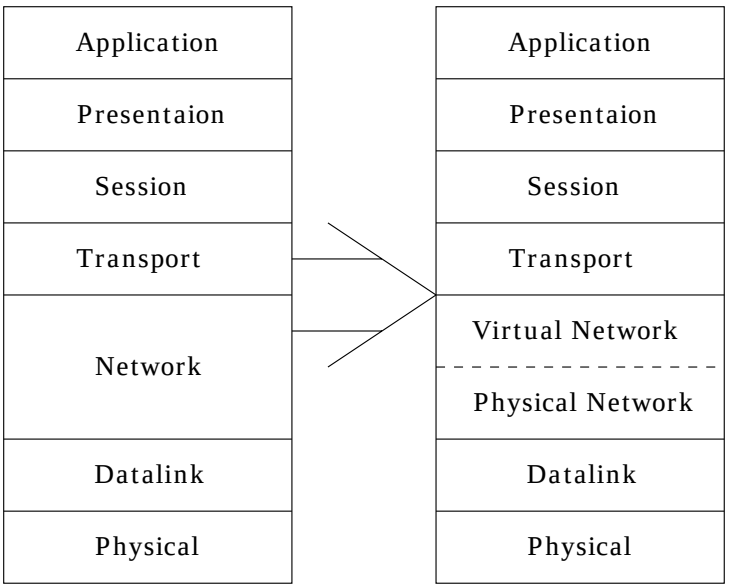

図 2.1: ネットワーク層の分割

ここで、ネットワーク層を既存のネットワーク層と仮想ネットワーク層の二つの層に分けたため、識別子から位置情報への対応付けをする必要が生じる。ネットワーク層より上位に位置する層では、通信相手の位置情報を考慮せず通信を行なうことが本来の目的である。ゆえに、仮想ネットワーク層で位置情報を付加しなければならない。この、位置情報の付加を行なうために、仮想ネットワーク層では、識別子を鍵として位置情報を得るためのデータベースが必要になる。しかも、このデータベースは検索速度が高速で、且つ Internet 上の全てのホストに対する情報が遅れることなく更新されなければならない。

2.1.3 拡散キャッシュ法

上で述べたようなデータベースを管理するために VIP では拡散キャッシュ法 (Propagating Cache Method) を用いている。拡散キャッシュ法とはネットワーク上で通信が行なわれる際に流れるパケットを使ってデータベースエントリを拡散させていく方法で、パケットを転送する際にルータは、そのパケットに含まれている送信ホスト情報を用いて自分のデータベースエントリを更新する。つまり、一つパケットを転送する度にそのパケットを送信したホストのホスト識別子を鍵とするエントリの更新または追加を行うことになる。この方法を用いた場合、データベースは基本的に特別なプロトコルを使って管理する必要がなく、その分のトラヒックを削ることが出来る。また、パケットが流れ

る経路上にないルータにはデータベースエントリが作られることはなく、無関係なホスト上にエントリを作りメモリ資源などを無駄に消費することはない。

2.2 Virtual Internet Protocol の概要

この節では仮想ネットワークの概念を Internet Protocol (IP) に適用した Virtual Internet Protocol (VIP) の概要について説明する。仮想ネットワークの概念を Internet Protocol に適用した場合、その階層化は図 2.2 のようになる。

2.2.1 仮想 Internet アドレス

前章で述べたような仮想ネットワークの概念を IP に適用するためには、ホストの識別子として新たに仮想 Internet アドレス (VIP アドレス) を導入する必要がある。理論上はこの識別子はどのようなフォーマットでも構わないが Internet アーキテクチャに新たに導入する関係上、32 ビット値とする。IP の上位プロトコルである TCP や UDP は下層が IP であることを前提としているため、下層に対して IP アドレスを渡す。そのため、IP の上位プロトコルを変更せずに VIP 層を導入するために、仮想 Internet アドレスは Internet アドレスと同じ 32 ビット値でなければならない。

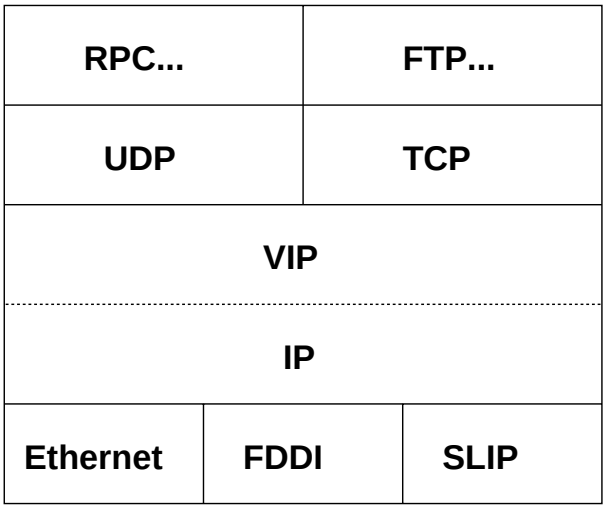

図 2.2: VIP 層と他の層の関係

2.2.2 Address Mapping Table

2.1.3 で述べたように仮想ネットワークの概念を導入するためにはホスト識別子の導入の他に、識別子 (仮想 Internet アドレス) から位置情報 (Internet アドレス) への対応づ

けを行なうためのデータベースの導入が必要になる。この、データベースを VIP では Address Mapping Table (AMT) と呼ぶ。AMT は仮想ネットワーク層に実装された仮想 Internet アドレスと Internet アドレスの組を要素として持つ表で、仮想 Internet アドレスを鍵として検索される。

原理的には AMT は全てのホストに対するエントリを持つ必要がある。しかし、実際には一つのホストが Internet に接続されている全てのホストに対するエントリを持つことはその大きさから不可能である。そこで、各ホストは膨大なデータベースの一部を AMT として持ち、他の部分については暗黙の対応付を用いることで対応している。VIP では AMT を拡散キャッシュ法を用いて管理している。

2.2.3 暗黙の対応付けとホームネットワーク

AMT を管理するために拡散キャッシュ法を使うと前述したが、この方法には一つ問題がある。これは、エントリを持たないノードへのパケットは転送、または送信することが出来ない点である。初期状態においてノードはキャッシュを全然持たないため、どのノードに対してもパケットを送信/転送することが出来ない。

この問題を解決するために VIP ではホームネットワークを導入して暗黙の対応付けを提供している。通信相手の AMT エントリを持たなかった場合、VIP では仮想 Internet アドレスを Internet アドレスとして設定する。こうすることにより、もし、途中のルータがその仮想 Internet アドレスに対するエントリを持っていなかったとしても、そのパケットは必ず自分の仮想 Internet アドレスと同じネットワーク番号を持つネットワークに到達する。自分の仮想 Internet アドレスと同じネットワーク番号を持っているネットワークのことをホームネットワークと呼ぶ。また、ホームネットワークに接続されているルータをホームルータと呼ぶ。ホームルータが移動ホストに対するエントリを持っていることを保証すれば問題は解決する。これを保証するために、移動ホストは実際にホームネットワークの外に移動した時に必ずホームネットワークにそのことを通知することが義務づけられている。また、移動ホストがホームネットワークに接続される時は、そのホストの仮想 Internet アドレスと Internet アドレスは一致し、他のルータ上の AMT エントリが破棄されてもパケットは正しく配送される。

2.3 Virtual Internet Protocol のプロトコル仕様

この節では VIP のプロトコル仕様について詳しく説明する。

2.3.1 VIP 層の実現

現行の階層化に VIP 層を挿入する方法として二つの方法が考えられる。一つは新たに独立した層を設ける方法であり、もう一つは IP のオプションとして挿入する方法である。VIP では後者を選択している。ネットワークアーキテクチャの観点から見れば前者

の方がはるかに整っているが、この方法を用いるためには Internet 上の全てのホストが VIP を実装しなければならず、現実的に不可能であることを踏まえた選択である。

VIP 層を実現するために IP オプションとして実装されている VIP ヘッダは図 2.3に示すようなフォーマットを持つ。ここで、太い線で囲まれた上の部分は IP パケットのヘッダとして定義されている部分で、その下の部分が VIP を実装するために新たに作ったオプション部である。

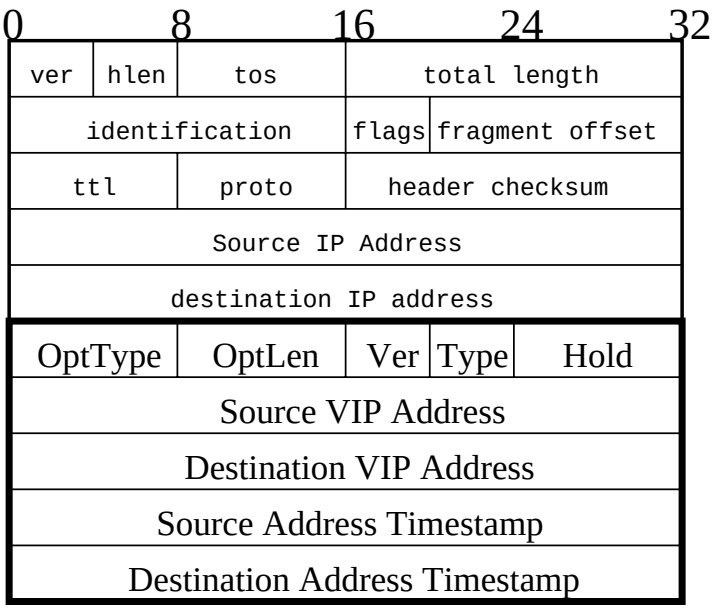

図 2.3: VIP パケットヘッダのフォーマット

それぞれのフィールドは次のような意味を持っている。

- *ver*: VIP のバージョン番号。現在のバージョンは 0。
- *type*: パケットの型を表す。VIP には次のようなパケットの型が定義されている。
  - *VipData* 通常のデータ。
  - *VipConn* 移動ホストが移動した時にホームネットワークに対して送信する通知。
  - *VipConnAck* ホームネットワークからの *VipConn* に対する確認応答。
  - *VipDisc* 移動ホストがネットワークから切断されるときにホームネットワークに対して送信する通知。
  - *VipDelAMT* まわりのノードの AMT エントリを破棄するための通知。
  - *VipErrObs* 古い AMT エントリを使って対応付けが行なわれたパケットを発見した時にルータおよびホストが発行するエラーパケット。

- *hold*: このパケットによって作成・更新された AMT エントリの有効期間 (単位は分)。
- *source VIP address*: 送信ホストの仮想 Internet アドレス。
- *destination VIP address*: 受信ホストの仮想 Internet アドレス。
- *source address timestamp*: 送信ホストの Internet アドレスと仮想 Internet アドレスの組が作成された時のタイムスタンプ。このタイムスタンプは source VIP address フィールドで特定されるホストによってのみ管理されるもので、対応付けを行なった時のタイムスタンプではない。
- *destination address timestamp*: 受信ホストの Internet アドレスと仮想 Internet アドレスの組が作成された時のタイムスタンプ。このタイムスタンプは destination VIP address フィールドで特定されるホストによって管理されるものであり対応付けを行なった時のタイムスタンプではない。途中で対応付けが行なわれる場合も AMT エントリに保存してあるタイムスタンプがこのフィールドに代入される。

このように IP のオプションという形をとる VIP ヘッダは VIP を実装していないホストでは unknown option として無視される。VIP を実装していないホストではこれまでの IP のヘッダとして図 2.3 の上部だけが処理される。

また、仮想 Internet アドレスはインターフェイスの識別子という位置付けで、インターフェイス毎に付ける。

2.3.2 Address Mapping Table

VIP の仕様の中で AMT の取り扱いも同時に定義されている。AMT で保持するべきデータは基本的には仮想 Internet アドレスと Internet アドレスだけだが、それ以外にも表を管理をするために必要な、アドレスの組が作られた時のタイムスタンプやエントリの取り扱いを示すフラグを表の中に持つことをプロトコル仕様として義務づけている。AMT のモデルでは AMT は図 2.4 の様なフォーマットをもつ。

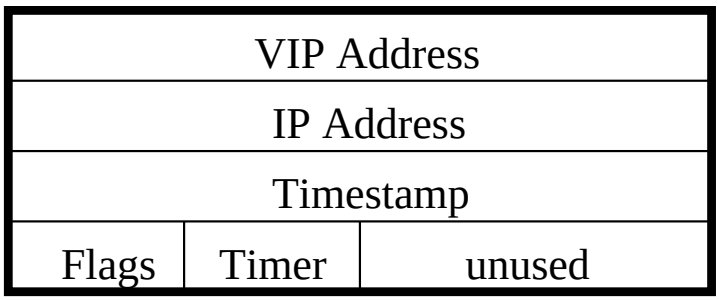

図 2.4: AMT エントリのフォーマット

それぞれのフィールドは次のような意味をもつ。

- *VIP address*: 仮想 Internet アドレス。検索の鍵として使われる。
- *IP address*: Internet アドレス。検索によって引き出す情報。
- *timestamp*: VIP/IP address のペアが作られた時のタイムスタンプ。アドレスの対応付けの時に使われる。
- *flags*: エントリを管理するために使われるフラグ。現在、次のフラグが存在する。
  - *AmtInUse* このエントリが使用中であることを示すフラグ。
  - *AmtPerm* このエントリはタイムアウトしないことを示す非消去フラグ。ホームネットワークで使用される。
- *timer*: エントリのタイムアウトを管理するためのタイマ。このフィールドによりタイムアウトが検出された時、そのエントリは消去される。エントリが作られる時に VIP パケットの hold time フィールドの値がこのフィールドに入れられ、1 分毎に減算される。

2.4　VIP の動作例

ここでは、例を示しながら VIP の全体の動作を説明する。図 2.5 は VIP を実装したホストを含んだネットワークの図である。ここで、四角はルータ、丸はエンドホストを、さらに斜線は VIP を実装しているホストを表している。

まず、この図で Host:A-a が Net:A-b から Net:B-a に移動した時の動作について説明する。Host:A-a は Net:B-a に接続された時、ホームネットワークである Net:A-b に (正確には Net:A-b のルータである GW:A-ab もしくは GW:A-bc に) 接続されたことを図 2.6-(a) のようなパケットを使って通知する。ここで、$var_{currVer}$ と表されているのはそのホストが内部に持つバージョン番号である。この通知は GW:B, GW:BB-ab, GW:A, GW:A-ad に Host:A-a の AMT エントリを作りながら GW:A-ab に到達する。この通知を受信した GW:A-ab は Net:A-b にこの通知を 2.6-(b) のようなパケットでブロードキャストする。このブロードキャストによって GW:A-bc にも Host:A-a の AMT エントリが作成される。このとき、GW:A-ab, GW:A-bc の AMT エントリには非消去フラグ (タイムアウトを無効にするフラグ) が立てられる。

ここで、Host:D-a が Host:A-a と通信することを考える。Host:D-a は Host:A-a を鍵とする AMT エントリを持たないので図 2.7-(a) のようなパケットを送信する。このパケットは IP の経路情報によって GW:BB-ab まで経路制御される。GW:BB-ab は Host:A-a からの接続通知により Host:A-a を鍵とする AMT エントリを持っているので、パケットのヘッダを図 2.7-(b) のように変更して、改めて Net:B-a 上の Host:A-a へ転送する。ここで、AMT_{IP}, AMT_{TS} という表記は AMT エントリの値であることを示している。

次に、Host:A-a が Host:D-a にパケットを送信することを考える。Host:A-a は Host:D-a を鍵とする AMT エントリを持たないので Host:D-a の Internet アドレスとして Host:D-a

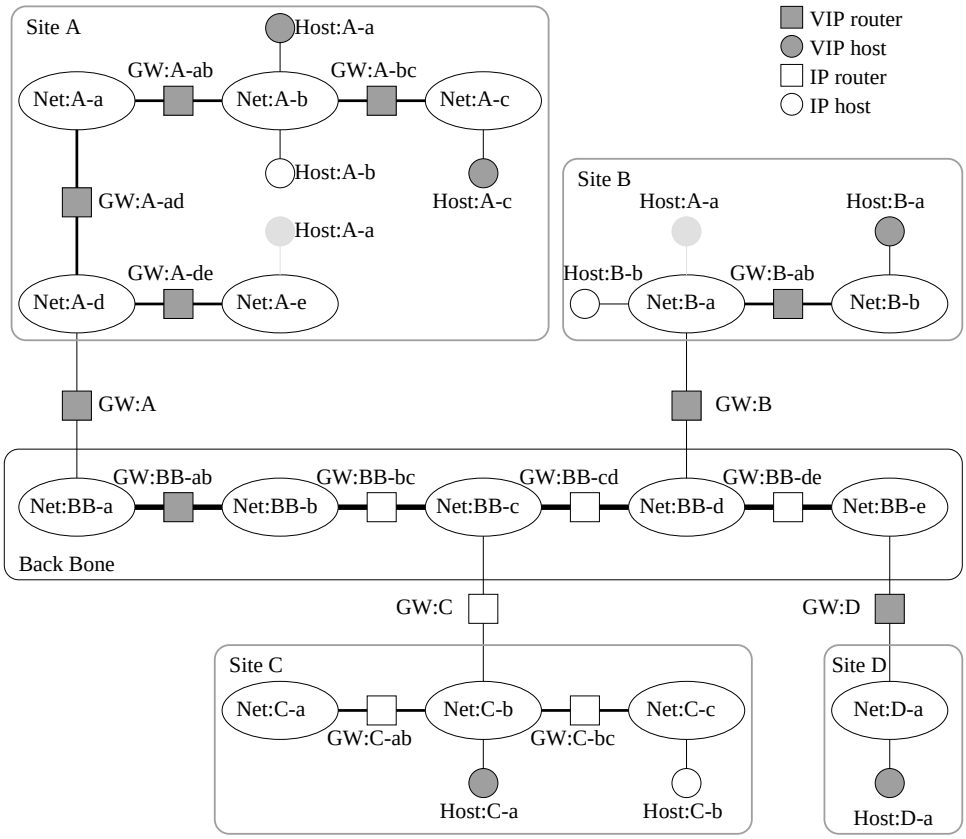

図 2.5: ネットワーク図

の仮想 Internet アドレスを設定したパケットを送信する。どのルータも Host:D-a を鍵とする AMT エントリを持たないので、このパケットは Host:D-a に到達するまで変更が加えられることはない。Host:D-a はこのパケットを受けとると、自分宛のパケットだと判断し、処理するとともに Host:A-a を鍵とする AMT エントリのタイマを初期化する。

この後、さらに Host:D-a が Host:A-a にパケットを送信する場合、Host:D-a は Host:A-a を鍵とする AMT エントリを保持しているので、Host:A-a の Internet アドレスを直接指定して送信することが出来る。

Host:A-a と Host:D-a 以外の通信がこのネットワークで行なわれないとすると、GW:BB-ab, GW:A, GW:A-ad 上の AMT エントリはタイムアウトによって消去される。さらに、Host:A-a, Host:D-a 間の通信が途絶えると、一定時間後に GW:B, GW:D, Host:D-a 上のエントリも消去される。

Net:A-b のルータ以外の全てのホストの AMT エントリが消去された後、Host:D-a が Host:A-a と通信する場合、ホームルータ上の AMT エントリはタイムアウトによって消去されることがないので、前述の GW:BB-ab が行なった変更が GW:A-ab で行なわれるにすぎない。

最後に Host:A-a が Net:B-a から切断される時の動作について説明する。Host:A-a は

field	value
$VIP_{SrcAddr}$	VIP address of *Host:A-a*
$IP_{SrcAddr}$	IP address of *Host:A-a*
VIP_{SrcTS}	$var_{currVer}$ of *Host:A-a*
$VIP_{DstAddr}$	VIP address of *GW:A-ab*
$IP_{DstAddr}$	IP address of *GW:A-ab*
VIP_{DstTS}	0

(a)

field	value
$VIP_{SrcAddr}$	VIP address of *Host:A-a*
$IP_{SrcAddr}$	IP address of *Host:A-a*
VIP_{SrcTS}	$var_{currVer}$ of *Host:A-a*
$VIP_{DstAddr}$	VIP address of *GW:A-ab*
$IP_{DstAddr}$	Bcast address of *Net:A-b*
VIP_{DstTS}	0

(b)

図 2.6: Host:A-a からの通知パケット

field	value
$VIP_{SrcAddr}$	VIP address of *Host:D-a*
$IP_{SrcAddr}$	IP address of *Host:D-a*
VIP_{SrcTS}	$var_{currVer}$ of *Host:D-a*
$VIP_{DstAddr}$	VIP address of *Host:A-a*
$IP_{DstAddr}$	VIP address of *Host:A-a*
VIP_{DstTS}	0

(a)

field	value
$VIP_{SrcAddr}$	VIP address of *Host:D-a*
$IP_{SrcAddr}$	IP address of *Host:D-a*
VIP_{SrcTS}	$var_{currVer}$ of *Host:D-a*
$VIP_{DstAddr}$	VIP address of *Host:A-a*
$IP_{DstAddr}$	value of AMT_{IP} field
VIP_{DstTS}	value of AMT_{TS} field

(b)

図 2.7: Host:A-a からの通知パケット

Net:B-a から切断される前に、切断通知パケット (VipDisc) をホームルータに送信する。GW:B, GW:BB-ab, GW:A, GW:A-ad はこの切断通知パケットを転送するが、その時、Host:A-a を鍵とする AMT エントリを保持していたらそのエントリを破棄して、さらに、AMT エントリ破棄要求 (VipDelAMT) をブロードキャストする。このブロードキャストを受けとった GW:A-bc, GW:A-de なども同じ動作をし、最終的にネットワーク上から Host:A-a を鍵とする AMT エントリは無くなる。

第 3 章

VIP の問題点

前章で説明したプロトコルでも移動するホストを接続した時の移動透過性を保証するという意味では十分である。しかし、実働しているネットワーク上にこのプロトコルを持ち込もうとした時、いくつかの問題が浮き彫りになる。これらの問題のうち今回解決する問題点をここで明確にする。各問題点については、前章の図 2.5 を使って例を挙げる。

3.1 孤立した AMT エントリの残留

ある移動ホストに対する AMT エントリを持つホストが、同じ AMT エントリを持つ他のホストから分断され、1 ホップ以内に同じ AMT エントリを持つ他のホストがなくなると、このエントリは一定の時間が経過するまで消えることがなく、その間、間違った対応づけをする。間違った対応づけは余分なトラヒックを発生するだけではなく、パケットを目的の移動ホストに配送する妨げになる。

Host:A-a が Net:B-a に接続されており、GW:D にその AMT エントリが存在しているとする。ここで、Host:A-a が Net:B-a から切断されて Net:A-e に接続されることを考える。Host:A-a はネットワークから切断される前に、ホームネットワークである Net:A-b に切断通知を行なう。このパケットは GW:B, GW:BB-cd, GW:BB-bc, GW:BB-ab, GW:A, GW:A-ad を経由して GW:A-ab へ到達する。このとき、GW:B, GW:BB-ab, GW:A, GW:A-ad, GW:A-ab は VipDelAMT をブロードキャストする。しかし、これらのパケットは GW:D には到達しないため、GW:D 上の AMT エントリは残留エントリとなる。この状態で Host:D-a が Host:A-a へのパケットを送信した場合、Host:D-a が Host:A-a を鍵とする AMT エントリを持たないならば、パケットは GW:D で Host:A-a の Net:B-a 上での Internet アドレスへ対応付けがなされ、Net:B-a へ経路制御される。

3.2 無期限の余分な転送

移動ホストが VipDisc パケットを送信せずにネットワークを離れると、その移動ホストが次にネットワークに接続されるまでホームネットワークのゲートウェイ上に作られた AMT エントリは消えることはなく、その間、それまで接続されていたネットワークにパケットを転送し続けることになる。この転送は、余分なトラヒックの発生源となる。

前章と同じく Host:A-a が Net:B-a へ接続されていて、GW:B, GW:BB-ab, GW:A, GW:A-ad およびホームネットワークのルータである GW:A-ab, GW:A-bc が Host:A-a を鍵とする AMT エントリを保持しているとする。この状態で、Host:A-a が切断通知を送信せずに、或は何らかの原因で切断通知パケットが途中で紛失されたままネットワークから切断された場合、GW:B, GW:BB-ab, GW:A, GW:A-ad の AMT エントリはいずれ破棄されるが、ホームルータである GW:A-ab, GW:A-bc が保持している AMT エントリは破棄されることはない。この残留 AMT エントリにより、ホームルータは他のホストから Host:A-a へ送信されたパケットを Net:B-a へ転送し続ける。

3.3　IP との互換性

VIP ホストが通常の IP ホストとの間でコネクションを張るとき、 IP ホストは VIP ホストをその物理 Internet アドレスで認識する。このことは VIP ホストが移動した時に、VIP ホストの移動透過性を保証できないことを意味する。

また、通常の運用の場合、移動ホストはネームサーバに名前と仮想 Internet アドレスを登録され、名前でそのアドレスを参照される。IP ホストが移動ホストと通信する場合、ネームサーバから得られたアドレスを使い、そのホームネットワークにパケットを送信するので、通信は不可能である。

Host:A-a が Net:B-a に接続されていて、Host:A-a と Host:C-b の間で通信が行なわれているとする。この時、Host:A-a は VIP ヘッダを含んだパケットを送信するが Host:C-b 側では、それを Unknown Option として処理する。つまり、Host:C-b 上のアプリケーション、および TCP/UDP 層は Host:A-a を Host:A-a の Net:B-a における Internet アドレスで認識している。ここで、Host:A-a が Net:A-b へ移動した場合、Host:C-b はそれを検知することが出来ず Net:B-a にパケットを送り続ける。

3.4　複雑な AMT 管理機構

VIP は多種のコントロールパケットを持っており、その分、ネットワークソフトウェアも複雑なものとなっている。また、3.2 節にも関連しているが、ホームネットワーク上の AMT は特に管理が複雑で、何らかの理由で矛盾が生じた場合、回復が難しい。

例として、何らかの理由でホームネットワーク上にタイムスタンプの値が大きなエントリが作成された時を考える。このエントリは、そのタイムスタンプより新しいタイムスタンプを持つパケットを受信するまで削除されるはことなく、その間、誤ったアドレスの対応づけを行なう。

第 4 章

Enhanced Virtual Internet Protocol への考察

本章では前章で述べた問題点を考慮しながら VIP の改良点について議論する。ここで、新たに改良した VIP をこれまでの VIP と区別するために Enhanced Virtual Internet Protocol (EVIP) と呼ぶ。

4.1 孤立した AMT エントリの残留

AMT エントリは、 VipDelAMT パケットが波の様に広がることにより消去される。しかし、VipDelAMT パケットが拡散することに対し、なんらかの制限を加えなければ、消去すべき AMT エントリに無関係なホストを含む、ネットワーク上の全てのホストに VipDelAMT パケットが配送されることになる。これでは大規模なネットワークでの使用に耐えることはできない。そこで、VIP では消す必要がある AMT エントリを持つホストに限り、VipDelAMT を中継している。この為、同じセグメント上に、消去する必要がある AMT エントリを持った他のホスト (ルータ) がない場合、そのホストに VipDelAMT が届くことはなく、その AMT エントリは更新されるか、一定の時間が経過するまで残留する。

この様な古い孤立した AMT エントリを消去するためには、その様な AMT エントリを発見した他のホストが、そのホストに対して指示をする必要がある。この為には古くなった AMT エントリを発見するための機構が必要となる。古い AMT エントリを持つホストに対する処置としては、これまでにも、VipErrObs コントロールパケットにより古い AMT エントリの消去を行なってきた。しかし、この場合の古い AMT エントリの発見は、発見する側のホストが同じ移動ホストに対する新しい AMT エントリを持っている場合に限られていた。しかし、実際に VIP を運用するにあたっては、古い AMT エントリに代わって新しい AMT エントリが作られる場合より、古い AMT エントリが消去されるだけに留まることが多い。そこで、今回、**AMT エントリの二段階消去**を行なうことにより、古い AMT エントリを発見するための機構を強化した。

実際には、これまで VipDelAMT を受信すると、ホストは直ちにその AMT エントリを消去していたところを、新たな消去機構では、実際に消去するのではなく、一旦、無効フラグを立て、消去タイマを設定するだけにとどめる。以後、無効フラグが立てられ

た AMT エントリは一定時間が経過するまで消去されるべき AMT エントリとして保存され、古い AMT エントリを持つホストを発見するために使われる。古い AMT エントリは実際には消去されず、無効フラグが立てられているだけなので、この AMT エントリを持つホスト (ルータ) を古い対応付けをされたパケットが通過する場合、そのパケットは古い AMT エントリによって対応付けされたものである、ということをホストは知ることが出来る。

しかし、これまでの VIP のパケットヘッダフォーマットではアドレスの対応付けを行なったホストを知ることが出来ない。これまでの VIP にあった VipErrObs コントロールパケットによる AMT エントリの消去では、パケットは同じ経路を通るものと仮定してパケットの発信元に VipErrObs コントロールパケットを送ることにより、途中、もしくは発信元にある古い AMT エントリの消去を行なっていた。しかし、日々複雑化している現在のネットワークでは必ずしもこの仮定は成り立たなくなっている。そこで、今回の改良では、新たにアドレスの対応づけを行なったホストの IP アドレスをヘッダに含めることにした。

さらに、この機構を確実に動作させるために移動ホストが直前に継っていたセグメント上の古い AMT エントリに無効フラグを立てるため、このセグメントにも VipConn を送信するようにする。これにより何処かに古い AMT エントリが残っており、移動ホストが以前に接続されていたセグメントへパケットが経路制御されても、最終的には正しい位置へ経路制御し、古い対応付けをしたホスト上の古い AMT エントリを消去することが出来る。この時、直前に接続されていたネットアークを**一時的代理ネットワーク**、そのネットワークのルータを**一時的代理ルータ**と呼ぶ。

4.2　無期限の余分な転送

VIP ではネットワークから切断される前に、そのことをホームネットワークに通知し、ネットワーク上のホストの AMT エントリを消去するため、VipDisc コントロールパケットを送信しなければならない。

ここで、ホームネットワーク上の AMT エントリは VipDisc/VipDelAmt コントロールパケットによってのみ消去されることに注意しなければならない。もし、移動ホストが VipDisc コントロールパケットを送信せずにネットワークを離れた場合、ホームネットワーク上のホストが次に移動ホストからのパケット (VipConn 等) を受けとるまでホームネットワーク上のホストにある古い AMT エントリは消えることはない。この残留 AMT エントリがあるためにホームネットワーク上のホストは移動ホストに対するパケットを、以前、移動ホストが接続されていたネットワークに転送し続け、トラヒックを増大させる元となる。

しかし、実際に VIP を運用する場合、このような状況は頻繁に発生する。可搬型の計算機の性格上、移動ホストは突然ネットワークから切り離される可能性が高く、それを前もって知ることが出来るのはユーザだけである。これまでの VIP ではユーザが VipDisc コントロールパケットを送信することを意識した上で運用されていたが、この状況は、ホ

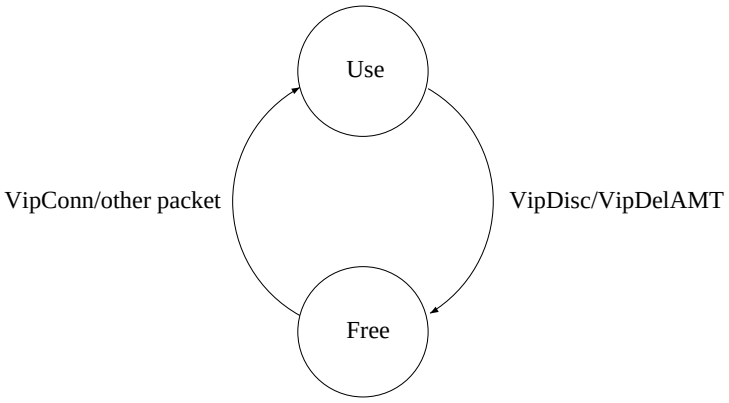

図 4.1: ホームルータの AMT の状態遷移

ストの移動透過性を実現する上で望ましくない。そこで、VipDisc を必要としない新たな AMT 管理機構が必要になる。

この問題は状態遷移図 (図 4.1) をみるとその性質は明らかであり、Use から Free へ移る状態遷移にコントロールパケットに依存しない線を加えることにより解決できる。コントロールパケットに依存しないイベントとしてタイマを選択した。

4.3 IP との互換性

VIP を実装したホストは IP ヘッダの source address, destination address フィールドに Internet アドレスを埋めて送信する。そのため、相手のホストが普通の IP ホストの場合、この Internet アドレスを移動ホストの識別子として使ってしまい、現行の IP と同じ理由により移動透過性を保証することが出来ない。

この問題は IP ヘッダの source address フィールドに Internet アドレスではなく仮想 Internet アドレスを入れることで解決できる。IP では、ホストは IP パケットヘッダの source address フィールドだけで通信相手を識別している。このことを逆に利用したのがこの方法である。移動ホストを仮想 Internet アドレスで認識した IP ホストは常にそのアドレスを使って通信を行なう。これは、VIP の視点から見ると VIP ヘッダはないもののアドレスの対応付けが行なわれていないパケットとしてみることが出来る。VIP ではアドレスの対応づけが行なわれていないパケットは最悪でもホームネットワークでアドレスの対応付けが行なわれることになっている。つまり、このパケットは最終的に移動ホストの仮想 Internet アドレスの位置に到達するまでに少なくとも一度はアドレスの対応づけをされる機会を持っている。この時にアドレスの対応付けだけではなく VIP パケットヘッダの付加を行なう (このことは IP パケットを VIP パケットに変換することを意味する。) ことにより IP ホストと VIP ホストとの間で移動透過性を実現することが出来る。

4.4　複雑な AMT 管理機構

VIP では AMT は基本的に以下の 5 種のコントロールパケットで管理される。

- *VipConn*: 移動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ためのコントロールパケット。
- *VipConnAck*: *VipConn* に対する確認応答パケット。
- *VipDisc*: 移動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 エントリが発見されたことを通知するパケット。

これらのパケットは、全て、AMT を制御するためのコントロールパケットである。しかし、AMT は前節の状態遷移図にあるように、2 つの状態しか持たず、これを制御するためには表 4.1のように 2 つのコントロールパケットで十分である。

表 4.1: AMT の状態遷移

	invalid	valid
Delete	invalid	invalid
Create	valid	valid

そこで、コントロールパケットを 2 種類に削減することで、AMT の制御機構を簡素化しプログラム自身を小さくし、より効率化をはかることが出来る。

- **VipCAmt** ターゲットに関する AMT エントリを作成/更新する。
- **VipDAmt** ターゲットに関する AMT エントリに削除フラグを立てる。

ただし、これまでの VIP では送信側仮想 Internet アドレスフィールドを使って AMT エントリ操作の鍵としていた。今回、明示的に鍵や値を示すように変更したので、鍵や値を特定するためのフィールドが VIP ヘッダに必要になる。

4.5　仮想 Internet アドレスの割当対象

これまでの VIP では、仮想 Internet アドレスはインターフェイス毎に設定していた。しかし、仮想 Internet アドレスは「位置に依存しない通信相手の識別子」という役割上、ホスト毎に設定する方が自然である。そこで、EVIP では仮想 Internet アドレスはホスト

毎に指定するように変更する。変更後は、ルータをしているホストでは複数の Internet アドレスを持つため、一つの仮想 Internet アドレスに対し複数のアドレスの対応を持つことなる。複数のアドレスの対応が存在してもプロトコル上問題は生じないが、固定のルータでは特に仮想 Internet アドレスを設定せず、インターフェイス毎に設定した Internet アドレスを仮想 Internet アドレスとして使用することを推奨する。

4.6 付加機能

これまでの VIP では将来の拡張性が考慮されておらず、機能を追加することが難しかった。そこで、今回の改良では拡張性を考慮して、以下のような付加機能を追加することにした。

- 途中のルータでは AMT エントリの作成・更新を行なわずず、ホームルータでのみ AMT エントリの作成・更新をする。
- 途中のルータではアドレスの対応づけをせず、ホームゲートウェイのみでアドレスの対応づけと行なう。
- AMT エントリを作成・更新をする際に、パケットを送信したホストの認証を行なう。

これらは EVIP に実際に仕様として追加されるが、さらに、機能を拡張する時のことを考慮してフラグフィールドを予約しておく。

4.7 ヘッダフォーマット

今回の改良で必要になったフィールドは次の通りである。

- アドレスの対応付を行なったホストのアドレス *(VIP_{RslvIP})*
- フラグ *(VIP_{Flags})*
- *AMT* エントリの鍵 *($VIP_{TVipAddr}$)* – 仮想 Internet アドレス
- *AMT* エントリの値 *($VIP_{TIpAddr}$)* – Internet アドレス
- *AMT* エントリの鍵/値の対のバージョン番号 *(VIP_{TVer})*
- *AMT* エントリの保存時間 *(VIP_{TVer})*

しかし、これらのフィールドを VIP ヘッダに含めると、ヘッダ長が大きくなり過ぎる。また、対象ホストに関するフィールドはコントロールパケットのみで使われるため、通常の通信の時には不必要なフィールドである。そこで、EVIP では、パケットを 2 種類分類し、データパケットとコントロールパケットで別のヘッダフォーマットを使用する。

4.8　タイプのフラグ化

これまでに述べてきたように、EVIP ではパケットをデータパケットとコントロールパケットに分類した。また、コントロールパケットはさらに CAMT (Create AMT entry) コントロールパケットと DAMT (Delete AMT entry) コントロールパケットに分類されている。これは、これまで同等の扱いをしてきたパケット分類を階層化したことに他ならない。このことを利用して EVIP ではパケットの分類をフラグで表す。

4.9　動作の安定性

EVIP では、ホームネットワーク上のホストの AMT エントリもタイムアウトにより削除されるように変更したため、移動ホストは一定時間 t 毎に VipCAmt コントロールパケットをホームネットワークに対して送信しなければならないが、この時のパケットの中に含まれるエントリ保持時間は t よりも長い必要がある。VIP では、ホームルータが、移動ホストに対する AMT エントリを必ず保持していることを要求している。t がエントリ保持時間より長い時、この条件を満たすことが出来なくり、ホームネットワークに AMT エントリがない時のパケットは紛失され移動ホストに届かなくなるからである。

また、エントリ保持時間と t が同じ値であることも推奨できない。エントリ保持時間と t が等しい場合、パケットを処理する時間が考慮すると僅かながらホームネットワーク上のホストに AMT エントリが存在しない時間が出来てしまい、動作が安定しない。安定した動作を提供するために $t \ll$ hold time とするべきである。

第 5 章

Enhanced Virtual Internet Protocol

この章では Enhanced Virtual Internet Protocol の仕様を記述する。

5.1 EVIP のヘッダフォーマット

EVIP のパケットフォーマットにはデータパケット用のヘッダフォーマットとコントロールパケット用のデータフォーマットが存在する。それぞれのフォーマットを図 5.1に示す。EVIP ヘッダは VIP と同様に IP オプションとして実装され、28 バイトの大きさを持つ。

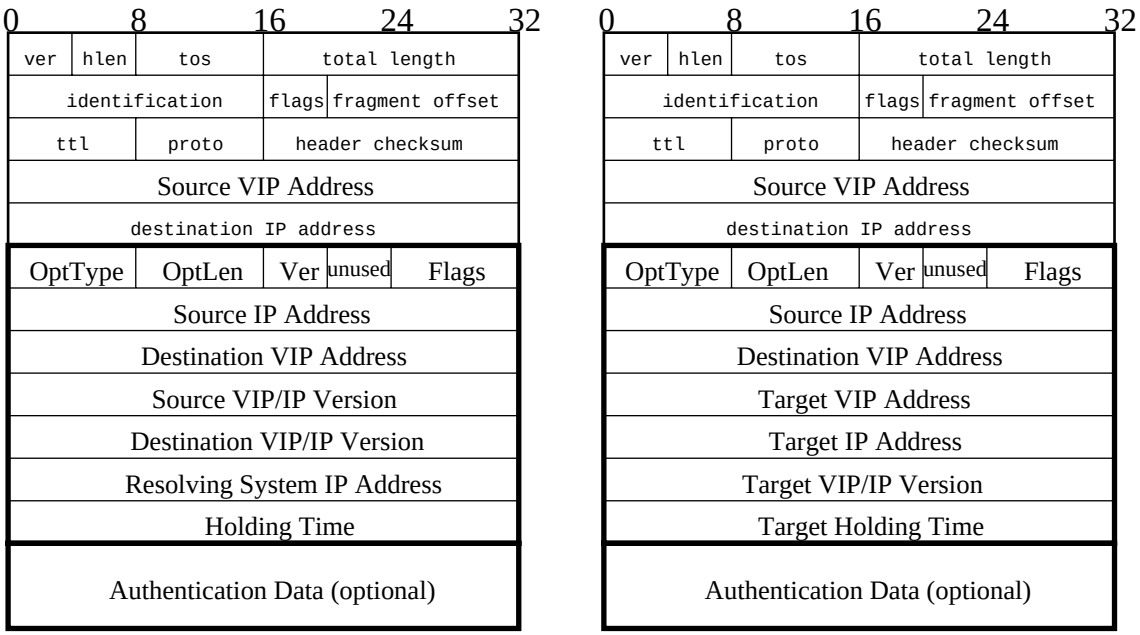

図 5.1: EVIP のヘッダフォーマット

データパケットのヘッダフォーマット

- 送信側仮想 Internet アドレス
  パケットを送信したホストの 仮想 Internet アドレス。

- OptType/OptLen
  IP オプションのタイプ番号およびそのオプションが使うヘッダの長さ。(このフィールドは Internet Protocol によって定義されている。) 現在、VIP ではオプション番号として 140(0x8C) を使用しているが、これは正式に割り当てられた番号ではない。

- VIP バージョン番号
  VIP のバージョン。現在のバージョンは 0 である。

- フラグ
  パケットの取り扱いに対するフラグ。現在、次のようなフラグが用意されている。

  - data/control: データパケットかコントロールパケットかのフラグ。
  - DtCacheIR: 途中のルータでは AMT エントリを作らない。
  - DtRslvIR: 途中のルータではアドレスの対応付けを行なわない。
  - BcastDAamt: ホームネットワークのルータ以外のルータが DAmt コントロールパケットを受けとった時、DAmt パケットをブロードキャストするかどうかを決める。(ホームネットワークでは必ずブロードキャストする。

- 受信側仮想 Internet アドレス
  パケットを受信するホストの仮想 Internet アドレス。

- 送信側アドレス対バージョン番号
  このパケットを送信したホストのアドレス対のバージョン番号。この番号は送信ホストによって生成される単調増加な番号でなければならない。

- 受信側アドレス対バージョン番号
  このパケットを受信するホストのアドレス対のバージョン番号。

- エントリ保持時間
  ルータがこのパケットによって作成・更新した AMT エントリを保持しておくべき時間。単位は秒。

- アドレス解決実行システム Internet アドレス
  このパケットの受信ホストのアドレスの対応付を行なったホストの Internet アドレス。

- 認証用データ
  パケット認証に使用される key などを格納するためのフィールド。(現在は使用されていない。)

コントロールパケットのヘッダフォーマット

- 送信側仮想 Internet アドレス
  データパケットと同じ

- OptType/OptLen
  データパケットと同じ

- VIP バージョン番号
  データパケットと同じ

- フラグ
  データパケットのフラグに次のフラグが追加される。

  - タイプ: 現在次のタイプがある。
    - 0: *CAmt* AMT エントリを作成・更新する。
    - 1: *DAmt* AMT エントリを削除する。

- 送信側アドレス対バージョン番号
  データパケットと同じ

- 対象システム仮想 Internet アドレス
  このコントロールパケットが対象とする AMT エントリの鍵となる仮想 Internet アドレス。

- 対象システム Internet アドレス
  このコントロールパケットが対象とする AMT エントリの値となる Internet アドレス。

- 対象システムアドレス対バージョン番号
  このコントロールパケットが対象とする AMT エントリのアドレス対のバージョン番号

- 対象システムエントリ保持時間
  このコントロールパケットが対象とする AMT エントリを保持しておくべき時間。単位は秒。

- 認証用データ
  コントロールパケットと同じ。

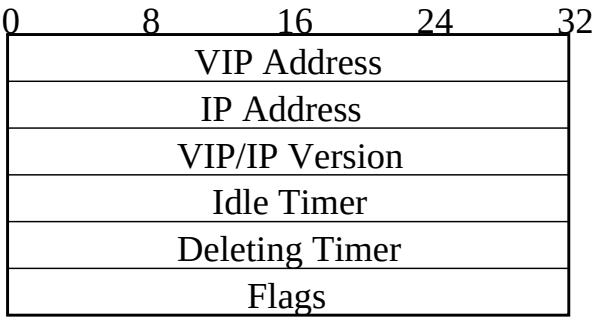

図 5.2: EVIP の AMT エントリのフォーマット

5.2 Address Mapping Table

プロトコルの改良に伴い変更した AMT の構造を図 5.2に示す。AMT の管理機構の改良に伴い、新たに deleting タイマが追加されている。

- *VIP Address*: このエントリの鍵となる仮想 Internet アドレス。
- *IP Address*: このエントリの値となる仮想 Internet アドレス。
- *Address Version*: このエントリのアドレス対のバージョン番号。
- *Holding Timer*: このエントリが無効になるまでの残り時間。このタイマは減算タイマであり、0 になると、エントリは invalid フラグが立てられる。
- *Deleting Timer*: このエントリが消去されるまでの残り時間。このタイマは invalid フラグがオンの間のみ動作する。このタイマは減算タイマであり、0 になると、このエントリは削除される。
- *Flags*: このエントリの制御フラグ。現在、次のようなフラグが存在する。
  - Use/Free: エントリが使用中かどうかを示すフラグ。
  - Valid/Invalid: エントリの有効/無効を示すフラグ。
  - home/not-home: エントリが子ホストのためのものかを示すフラグ。
  - local/not-local 同じセグメントに接続されているホストの為のエントリであることを示すフラグ。

5.3 エントリ保持時間

4.9 節で議論したように VipCAmt の送信間隔 t は t ≪ hold time でなければならない。そこで、EVIP では $3t$ = エントリ保持時間 とする。これにより、VipCAmt コントロールパケットが一回失われることがあっても安定に動作することが可能となる。

5.4 EVIP における接続手順

移動ホストは新たにネットワークに接続された時、VipCAmt コントロールパケットを接続通知としてホームネットワークに送信しなければならない。VipCAmt コントロールパケットを中継するルータは、自分が保持している AMT を更新しつつパケットを転送する。また、この VipCAmt コントロールパケットを受信したホームルータは AMT を更新し、更に VipCAmt コントロールパケットを移動ホストが直前に接続されていたネットワークへ送信する。このパケットにより直前に接続されていたネットワークのルータの AMT エントリは更新され、一時的代理ホストとして動作する。以下に途中ルータ、ホームルータ、一時的代理ホストの動作を示す。

途中ルータ

if (移動ホストに対する AMT エントリを持っている) {
  if (AMT エントリの方が古い) {
    ・VipDAmt をブロードキャストする。
    ・移動ホストに対する AMT エントリを更新する。
    ・パケットを経路情報に従って中継する。
  } else if (AMT エントリとバージョンが等しい) {
    ・移動ホストに対する AMT エントリのタイマをリセットする。
    ・パケットを経路情報に従って中継する。
  } else if (AMT エントリの方が新しい) {
    ・VipDAmt をアドレスの対応づけを行なったホストへ送信する。
    ・パケットを破棄する。
  }
} else {
  ・移動ホストに対する AMT エントリを作成する。
  ・パケットを経路情報に従って中継する。
}

ホームルータ

if (移動ホストに対する AMT エントリを持っている) {
  if (AMT エントリの方が古い) {
    ・VipDAmt をブロードキャストする。

・VipCAmt をホームネットワークにブロードキャストする。
・移動ホストが直前に接続されていたネットワークに VipCAmt を送信する。
・移動ホストに対する AMT エントリを更新する。
} else if (AMT エントリとバージョンが等しい) {
・VipCAmt をホームネットワークにブロードキャストする。
・移動ホストに対する AMT エントリのタイマをリセットする。
} else if (AMT エントリの方が新しい) {
・VipDAmt をアドレスの対応づけを行なったホストへ送信する。
・パケットを破棄する。
}
} else {
・VipCAmt をホームネットワークにブロードキャストする。
・移動ホストに対する AMT エントリを作成する。
}

一時的代理ホスト

if (移動ホストに対する AMT エントリを持っている) {
if (AMT エントリの方が古い) {
・VipDAmt をブロードキャストする。
・VipCAmt を 移動ホストが接続されていたネットワークに
ブロードキャストする。
・移動ホストに対する AMT エントリを更新する。
} else if (AMT エントリとバージョンが等しい) {
・VipCAmt を移動ホストが接続されていたネットワークに
ブロードキャストする。
・移動ホストに対する AMT エントリのタイマをリセットする。
} else if (AMT エントリの方が新しい) {
・VipDAmt をアドレスの対応づけを行なったホストへ送信する。
・パケットを破棄する。
}
} else {
・移動ホストに対する AMT エントリを作成する。
・VipCAmt を移動ホストが接続されていたネットワークに
ブロードキャストする。
}

この様子を図 2.5 を使って説明する。Host:A-a が Net:A-e から Net:B-a へ移動することを考える。現在、GW:A-de, GW:A-ad, GW:A-ab, GW:A-bc に Host:A-a を鍵とする AMT エントリが存在するとする。Net:B-a に移動した後、Host:A-a は図 5.3-(a) の様な VipCAmt コントロールパケットを Net:A-b に対して送信する。これを中継する時、Host:A-a を鍵とする AMT エントリを保持しているルータ GW:A-ad は図 5.3-(b) の様な VipDAmt コントロールパケットをブロードキャストする。また、ホームルータである GW:A-ab は VipDAmt をブロードキャストすると同時に図 5.3-(c) の様な他のホームルータへの通知、図 5.3-(d) の様な一時的代理ホストを作るための VipCAmt コントロールパケットを送信する。

field	value
VIP_{Type}	VipCAmt
$VIP_{SrcAddr}$	VIP address of *Host:A-a*
$IP_{SrcAddr}$	IP address of *Host:A-a*
$VIP_{DstAddr}$	VIP address of *Host:A-a*
$IP_{DstAddr}$	VIP address of *Host:A-a*
$VIP_{TVipAddr}$	VIP address of *Host:A-a*
$VIP_{TIpAddr}$	IP address of *Host:A-a*
$VIP_{TVipVer}$	$var_{currVer}$ of *Host:A-a*

(a)

field	value
VIP_{Type}	VipDAmt
$VIP_{SrcAddr}$	VIP address of *Host:A-ad*
$IP_{SrcAddr}$	IP address of *Host:A-ad*
$VIP_{DstAddr}$	Broadcast address
$IP_{DstAddr}$	Broadcast address
$VIP_{TVipAddr}$	VIP address of *Host:A-a*
$VIP_{TIpAddr}$	IP address of *Host:A-a*
$VIP_{TVipVer}$	$var_{currVer}$ of *Host:A-a*

(b)

field	value
VIP_{Type}	VipCAmt
$VIP_{SrcAddr}$	VIP address of *GW:A-ab*
$IP_{SrcAddr}$	IP address of *GW:A-ab*
$VIP_{DstAddr}$	Broadcast address
$IP_{DstAddr}$	Broadcast address
$VIP_{TVipAddr}$	VIP address of *Host:A-a*
$VIP_{TIpAddr}$	IP address of *Host:A-a*
$VIP_{TVipVer}$	$var_{currVer}$ of *Host:A-a*

(c)

field	value
VIP_{Type}	VipCAmt
$VIP_{SrcAddr}$	VIP address of *GW:A-ab*
$IP_{SrcAddr}$	IP address of *GW:A-ab*
$VIP_{DstAddr}$	value of $AMT_{VipAddr}$ field
$IP_{DstAddr}$	value of AMT_{IpAddr} field
$VIP_{TVipAddr}$	VIP address of *Host:A-a*
$VIP_{TIpAddr}$	IP address of *Host:A-a*
$VIP_{TVipVer}$	$var_{currVer}$ of *Host:A-a*

(d)

図 5.3: 接続処理時のパケットフォーマット

5.5 EVIP における通信手順

この章では EVIP における通常のデータ通信の動作を説明する。

Host:A-a が Net:B-a に接続されているとき Host:D-a との間で通信することを考える。Host:D-a が Host:A-a に最初にパケットを送信する時 Host:D-a は Host:A-a を鍵とする AMT エントリを持っていないので図 5.4-(a) の様なパケットを送信する。このパケットは GW:BB-ab で対応づけが行なわれ 図 5.4-(b) の様なヘッダに変換され Host:A-a に転送される。Host:A-a から Host:D-a へパケットが送られると、そのパケットによって Host:D-a に Host:A-a を鍵とする AMT エントリが作られる。その後 Host:D-a が Host:A-a に対

してパケットを送信する時は、図 5.4-(c) の様なパケットを送信することが出来る。

field	value
VIP_{Type}	VipData
$VIP_{SrcAddr}$	VIP address of *Host:D-a*
$IP_{SrcAddr}$	IP address of *Host:D-a*
VIP_{SrcVer}	$var_{currVer}$ of *Host:D-a*
$VIP_{DstAddr}$	VIP address of *Host:A-a*
$IP_{DstAddr}$	VIP address of *Host:A-a*
VIP_{DstVer}	0
VIP_{RslvIP}	0

(a)

field	value
VIP_{Type}	VipData
$VIP_{SrcAddr}$	VIP address of *Host:D-a*
$IP_{SrcAddr}$	IP address of *Host:D-a*
VIP_{SrcVer}	$var_{currVer}$ of *Host:D-a*
$VIP_{DstAddr}$	VIP address of *Host:A-a*
$IP_{DstAddr}$	value of AMT_{IpAddr}
VIP_{DstVer}	value of AMT_{Ver}
VIP_{RslvIP}	IP address of *GW:BB-ab*

(b)

field	value
VIP_{Type}	VipData
$VIP_{SrcAddr}$	VIP address of *Host:D-a*
$IP_{SrcAddr}$	IP address of *Host:D-a*
VIP_{SrcVer}	$var_{currVer}$ of *Host:D-a*
$VIP_{DstAddr}$	VIP address of *Host:A-a*
$IP_{DstAddr}$	value of AMT_{IpAddr}
VIP_{DstVer}	value of AMT_{Ver}
VIP_{RslvIP}	IP address of *Host:D-a*

(c)

図 5.4: 通信時のパケットフォーマット

5.6　EVIP における切断手順

EVIP では 2 種類の切断方法がある。一つは「通知を行なわない切断」、もう一つは「明示的な切断」である。

通知を行なわない切断:　この方法では、移動ホストは VipDAmt コントロールパケットを送信することなくネットワークから切断される。ネットワーク上に残留している AMT エントリについては、次の VipCAmt パケットによって更新されるか、タイムアウトにより消去されることを期待した方法である。

実際にこの方法で切断される場合は、移動ホストがネットワークから切断されるだけなので特に例を示すことはしない。

明示的な切断:　この方法は、実際にネットワークから切断される際に VipDAmt コントロールパケットを用いてホームネットワークに通知を行なう方式である。この方法を用いた場合、ホームネットワーク上の AMT エントリを積極的に無効に出来るだけでなく、パケットが転送される途中のルータ上の AMT エントリも無効にすることが出来る。さ

らに、コントロールパケットの broadcast DAMT フラグを 1 にしておくことによりこれまでの VIP と同じように VipDAmt コントロールパケットを拡散することも出来る。

前章の続きで Host:A-a が Net:B-a から切断される場合の例を示す。Host:A-a は切断される前に図 5.5-(a) の様なパケットを送信する。このパケットの broadcast DAMT フラグが 1 ならば途中の GW:B, GW:BB-ab, GW:A, GW:A-ad はこのパケットを転送すると同時に図 5.5-(b) のようなパケットをブロードキャストする。

field	value
VIP_{Type}	VipDAmt
$VIP_{SrcAddr}$	VIP address of *Host:A-a*
$IP_{SrcAddr}$	IP address of *Host:A-a*
$VIP_{DstAddr}$	VIP address of *Host:A-a*
$IP_{DstAddr}$	VIP address of *Host:A-a*
$VIP_{TVipAddr}$	VIP address of *Host:A-a*
$VIP_{TIpAddr}$	IP address of *Host:A-a*
$VIP_{TVipVer}$	$var_{currVer}$ of *Host:A-a*

(a)

field	value
VIP_{Type}	VipDAmt
$VIP_{SrcAddr}$	VIP address of source GW
$IP_{SrcAddr}$	IP address of source GW
$VIP_{DstAddr}$	broadcast address
$IP_{DstAddr}$	broadcast address
$VIP_{TVipAddr}$	VIP address of *Host:A-a*
$VIP_{TIpAddr}$	IP address of *Host:A-a*
$VIP_{TVipVer}$	$var_{currVer}$ of *Host:A-a*

(b)

図 5.5: 切断時のパケットフォーマット

第 6 章

EVIP の実装

本研究では 4.3 BSD UNIX オペレーティングシステムを基盤とした二つのオペレーティングシステム (NewsOS 4.1、BSD/386) 上で EVIP を実装した。本章ではこの実装について記述する。

6.1 記憶領域の確保

VIP を実装するためには少なくとも次の 2 つを保存するための領域を確保する必要がある。

- 仮想 Internet アドレス
- Address Mapping Table

これらの領域は kernel 内部に静的変数として確保した。仮想 Internet アドレスは in_addr 構造体で、AMT は付録の amt 構造体を配列で確保した。

6.2 処理系の構成

今回実装した VIP のブロックダイアグラムを図 6.1に示す。ここで細線で表された矢印は IP の処理の流れを表しており、太線が VIP を実装するために付加した部分である。

送信の場合、基本的に VIP オプションを付加するだけである。まず、TCP/UDP 層から ip_output モジュールが呼ばれ、その直後に vip_output モジュールを通る。このモジュールで VIP オプションが作られる。作られた VIP オプションはもう一度 ip_output モジュールに渡され、この中で他のオプションと共に IP ヘッダに付加される。その後、ip_mapdstaddr モジュールが呼ばれ受信ホストのアドレスの対応付けが行なわれる。対応付けが行なわれた後は、通常の IP の送信手順に従い、ルーティングテーブルが検索される。この時点でパケットを送出するインターフェイスが決まり、fill_ip モジュールを呼び出して VIP オプションの送信側 Internet を設定する。その後、if_output モジュールを呼び出してパケットを送出する。

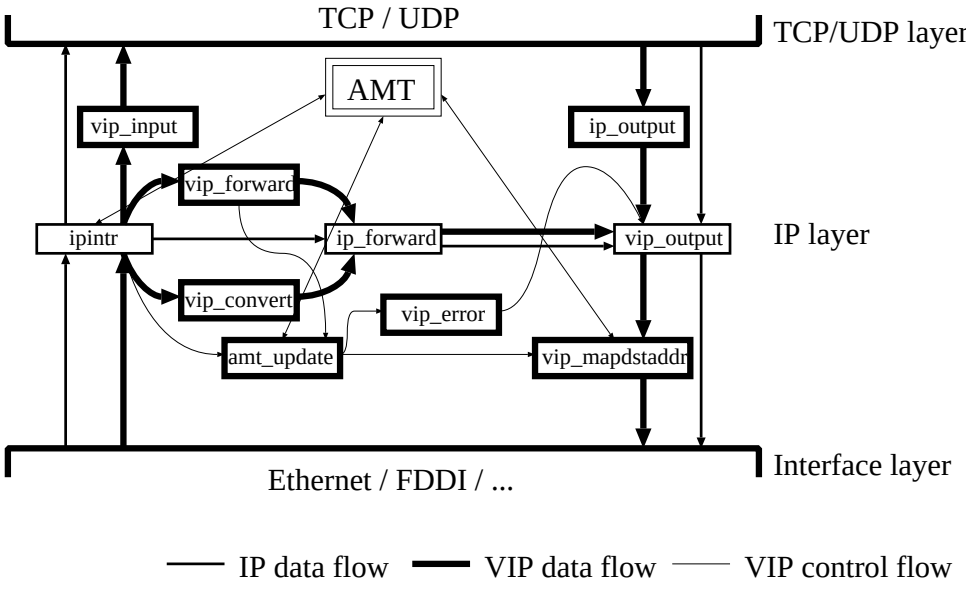

図 6.1: ブロックダイアグラム

受信の場合、Internet アドレス、IP ヘッダのチェックサムをチェックした後、仮想 Internet アドレスを確認しなければならない。インターフェイスがパケットを受信すると割り込みが入り、ipintr モジュールが呼び出される。ipintr モジュールでは送信ホストの仮想 Internet アドレスを鍵とする AMT エントリを更新した後、Internet アドレスの確認を行ない、自分宛のパケットならば vip_input モジュールを呼び出す。vip_input モジュールでは VIP アドレスの確認と、そのパケットがコントロールパケットであるか否かの確認を行ない、自分宛のデータパケットなら受信側 Internet アドレスフィールドに仮想 Internet アドレスを代入し、上位プロトコルに渡す。コントロールパケットは vip_input モジュールで処理を行なった後、破棄し上位プロトコルには渡さない。

中継の場合、受信側のアドレスの対応付けを行なった後に転送すればよい。ipintr モジュールで自分宛のパケットではないと判断された場合、vip_forward モジュールが呼ばれ、もしこのパケットがコントロールパケットならば、ここで一度処理される。その後、ip_forward モジュールに渡されフォワードされる。アドレスの対応付けは、実際にはここでは行なわれず、送信の時と同じように ip_output モジュールから ip_mapdstaddr モジュールが呼ばれる時に行なわれる。

また、中継するパケットが普通の IP パケットだった場合、vip_forward モジュールの代わりに vip_convert モジュールが呼ばれ、VIP パケットに変換されてからフォワードされる。これにより、通常の IP ホストとの通信中も移動ホストの移動透過性を保証することが可能となる。

第 7 章

評価

この章では、実装した EVIP の評価および考察を行なう。

7.1 実験環境

本研究では Sony 社のワークステーションである NEWS を 7 台用いて図 7.1の様なネットワークを構築し、評価を行なった。番号は NEWS の型番である。今回使用した NEWS はいずれも CPU はクロックが 25MHz の MC68030 で、メモリは 4 〜 12 M byte のものである。

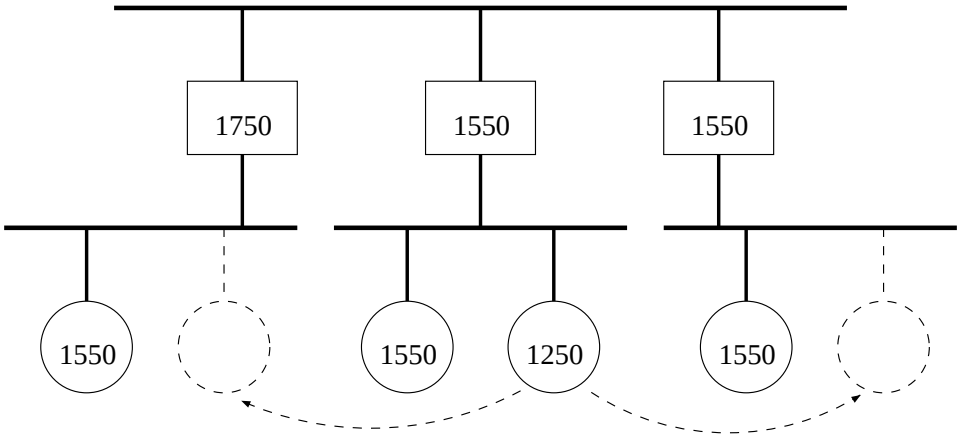

図 7.1: 実験環境

7.2 オーバヘッド

NEWS-1750 上で送信、中継、受信について 1 パケットを処理するのにかかるオーバヘッドを改良前の実装、改良後の実装および既存の IP について計測した結果を図 7.2に示す。図の数値の単位はマイクロ秒であり、括弧の中の数字はオーバヘッドを示してい

る。計測は ICMP ECHO パケットを VIP ホストから送信し、kernel 内にとられた領域に、モジュール毎にタイムスタンプを残し、そのダンプを解析することにより行なった。

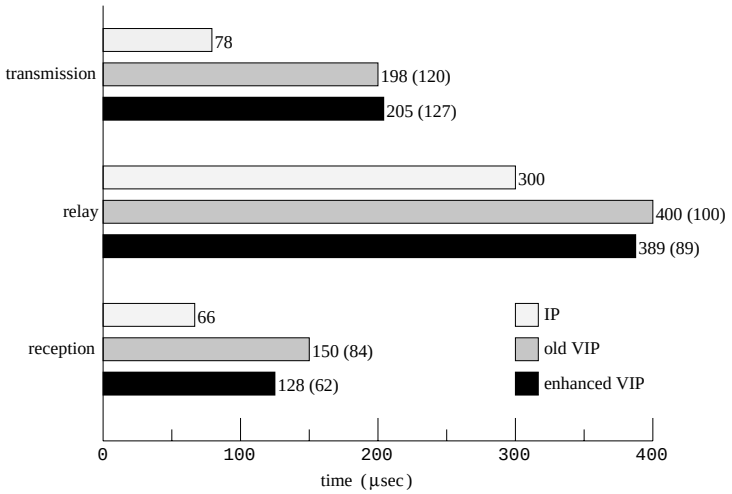

図 7.2: オーバヘッド

また、それぞれのオーバヘッドの詳細を表 7.1から表 7.3に示す。

表 7.1: 送信のオーバヘッドの内訳

transmission overhead: 127				
vip_output()	ip_insertoptions()	vip_mapdstaddr()	fill_ip()	others
43	39	25	17	3

μsec

表 7.2: 転送のオーバヘッドの内訳

relay overhead: 89			
vip_forward()	vip_mapdstaddr()	fill_ip()	others
37	25	17	10

μsec

表 7.3: 受信のオーバヘッドの内訳

reception overhead: 67		
ip_dooptions()	vip_input()	others
23	43	1

μsec

7.3　オーバヘッドの考察

この章では、今回の実装により生じたオーバヘッドについて考察する。

実装は既存のネットワークプログラムに追加する形でなされているため、実装前より速くなることはない。測定結果を見ても、ある程度のオーバヘッドがあることがわかる。この中で、送信時のオーバヘッドは改良前の実装よりさらに処理が遅くなっている。これは、今回の改良により複雑化したヘッダ (アドレス解決実行システム Internet アドレスの挿入等) の処理部に時間がかかるのに対し、他の簡略化された部分 (コントロールパケットのタイプの削減等) が送信時の処理には影響を及ぼさないためである。送信時のオーバヘッドは現行の IP の約 160%もあり、さらに実装の面で改良する必要がある。

送信時のオーバヘッドによる影響は、パケットを送信するホストの CPU の処理時間を多く消費することと、パケットを送り出すまでに要する時間が長くなること以外にはない。これは受信時のオーバヘッドについても同じことが言える。しかし、中継時のオーバヘッドにはホップ数が掛かる。パケットは通常、相手のホストに到達するまでに幾つかのルータを経由する。この経由するルータの数が転送時のオーバヘッドに掛かってくる。この様子を次式に示す。

$$\begin{aligned} \mathit{Host\text{-}to\text{-}Host\ overhead} &= (\mathit{relay\ overhead}) \times N_{hops} \\ &\quad +(\mathit{transmission\ overhead}) + (\mathit{reception\ overhead}) \\ &= 89 \times N_{hops} + 127 + 62 \\ &= 89 \times N_{hops} + 189 \end{aligned} \tag{7.1}$$

現在考えているようなホストが移動した状況ではとくに経由するルータの数が増えることが予想されるので転送時のオーバヘッドを最小限にとどめることは重要である。現在の実装では転送時のオーバヘッドは約 30%である。途中のルータの数に対する VIP のオーバヘッドを表 7.4 に示す。

表 7.4: host-to-host Overhead

	host-to-host overhead	
Hops	old VIP	enhanced VIP
10	1.2	1.1
20	2.2	2.0
30	3.2	2.9

msec

7.4 考察

この章では、3 章で説明した問題についての検証、その他今回の研究により改良された部分について議論する。

7.4.1 孤立した AMT エントリの残留

孤立した AMT が残留する問題について 3.1 節で用いた例と同じ例を用いて検証する。

Host:A-a が Net:B-a に接続されており、GW:D にその AMT エントリが存在している。ここで、Host:A-a が Net:B-a から切断されて Net:A-e に接続されることを考える。Host:A-a はネットワークから切断され Net:A-e に接続された時、ホームネットワークである Net:A-b へ VipCAmt コントロールパケットを送信する。GW:A-ab ではこのパケットを Net:A-b へブロードキャストすると同時に Net:B-a へも VipCAmt パケットを転送する。これにより、Net:B-a 上の残留エントリは更新され GW:B は一時的代理ホストとして動作するようになる。しかし、この状態では、まだ、GW:D 上の AMT エントリは残留エントリとして残っている。この状態で Host:D-a が Host:A-a へパケットを送信することを考える。Host:D-a から送信されたパケットは GW:D-a により古い対応づけをなされて Net:B-a へ転送される。しかし、その入口である GW:B は正しい AMT エントリを保持しているため、パケットは新しい対応づけがなされ Net:A-e へ転送される。GW:B は同時に間違った対応づけを行なった GW:D に対して VipDAmt コントロールパケットを送り、GW:D 上の残留エントリを消去することが出来る。

7.4.2 無期限の余分な転送

この問題を解決するためにホームルータの AMT エントリについてもタイムアウトを設けた。これにより、移動ホストがネットワークから切断され、定期的な VipCAmt が届かなくなると確実にホームルータ上の AMT エントリは消去される。しかし、このことは新たな問題点をつくり出した。定期的な VipCAmt によるトラヒックは 4.9節で述べた t ≪hold time の関係から頻繁に移動するノードでは大きくなってしまう。t = 1.0 (min)

のホストでは約 6bit/sec のトラヒックを発生させることになる。これは移動ノードが増えるに従って大きな問題となる。

7.4.3　IP との互換性

IP との互換性について 3.1 で用いた例と同じ例を用いて検証する。

Host:A-a が Net:B-a に接続されていて、Host:A-a と Host:C-b の間で通信が行なわれているとする。この時、Host:A-a は VIP ヘッダを含んだパケットを送信するが Host:C-b 側では、それを Unknown Option として処理する。しかし、実際には今回の改良により IP ヘッダに含まれている送信ホストアドレスは仮想 Internet アドレスであるため、ホストは Host:A-a をその仮想 Internet アドレスで認識する。ただし、Host:C-b から送信されたパケットは一度 GW:BB-ab に到達し、そこで VIP パケットに変換された後 Net:B-a へ転送されることになる。この場合、Host:A-a が移動した後も他のルータが正しい AMT エントリを保持していれば通信の連続性、つまり、ホストの移動透過性は保証される。

7.4.4　複雑な AMT 管理機構

VIP のコントロールパケットの種類を減らすことにより AMT 管理のメカニズムを簡略化した。これによりソフトウェアも簡略化でき、オーバヘッドの表にも現れているように転送/受信時のコストを削減できた。特に前述した理由により転送時のコストの削減は VIP 全体のパフォーマンスに直接大きな影響を与えている。これは、AMT の二段階消去などの機能を追加したにも関わらずパフォーマンスが落ちていないことを見てもわかる。

第 8 章

今後の課題

これまで述べてきたような問題の他に、まだ、次のような問題点が残っている。

- **IP アドレス取得の問題**
  VIP では移動ホストがネットワークを移動した時に、一時的な Internet アドレスを取得しなければならない。一時的な Internet アドレスの取得過程を自動化するためには問題が二つある。一つはネットワークを移動したことを、ユーザを通さずに如何にして検知するかという問題で、もう一つはどのようなプロトコルで IP アドレスを取得するかである。現在、この問題は解決されつつあり、移動の検知はパケットが届かなくなったことをタイムアウトにより検知し、プロトコルとしては Dynamic Host Configration Protocol (DHCP) を使う方向で検討している。現在、移動したことを検知するためのデーモンプログラムは完成しており、アドレス取得のプロトコルとして Bootstrap Protocol (BOOTP) を使用している。DHCP はまだ正式にリリースされたプロトコルではなく、これがリリースされると同時にこのプロトコルを導入する予定である。また、移動検知の方も kernel レベルで検知できるものについては ioctl または system call を使って CPU 時間の消費を出来るだけ押えた形で実装していく予定である。

- **耐故障性**
  現在の VIP の仕様では Home Network への到達性がないネットワークには接続することが出来ない。当然、何らかのトラブルによってネットワークが切断された場合、通信が不可能になる。これを回避するために、我々は代理ホームネットワークを導入することを検討中である。複数のホームネットワークを導入するためにはホームネットワーク間の AMT エントリの整合性を如何にして保証するかという問題を解決しなければならない。現在、この方法を検討中である。

- **認証**
  移動ノード用のプロトコルは、その名の通り移動できるような小型の計算機を想定したプロトコルであるため、その使用される状況を考えると認証を疎かにすることはできない。小型な計算機は持ち運ぶことが出来るためネットワークのあらゆる場所に接続されることが考えられる。これは頻繁に VipCAmt を送信することによりなりすましが出来ることを意味する。認証が確実に出来てないような状態ではネッ

トワークを信頼することは出来ない。そこで何らかの認証システムが必要になるが IP はもともと認証機構を含んでいないため外側から認証機構を採り入れても結局基本的なところで問題がでてくる。我々は今後 IP 自体に認証機構が採り入れられる様に希望すると同時にその機構の開発に力を注ぐつもりである。

第 9 章

まとめ

本研究では VIP を実用にするため改良を加えた。最大の問題点であった IP との互換性を高めるためヘッダフォーマットの変更、変換ルータを導入を行なった。また、AMT の管理機構をいくつかの面から強化し、古い AMT エントリの残留による無駄なトラヒックを削減した。

また、改良した VIP を実装し、その評価を行なった。改良後の VIP では使用状況によっては定常状態でのトラヒックが増加するという問題点が新たに浮き彫りにされたことを除くと、これまでの VIP より優れた結果が得られた。

しかし、まだオペレーティングシステムレベル、アプリケーションレベルでの移動ノードの支援は十分ではない。携帯できる計算機は比較的小規模であり資源を十分に用意できない。このような場合、当然、ネットワーク上の資源を有効に利用することになる。しかし、ネットワークファイルシステムなどを見るとわかるように、既存のソフトウェアを変更を加えずに利用すると、ネットワークに大きな負荷をかけることは必至である。このような無理な負荷をネットワークにかけないためにも、携帯している計算機で作業をするためにはさらに別の面から研究を進めていく必要がある。

第 10 章

付録

10.1 amt 構造体

以下に amt 構造体を定義しているヘッダファイルを付す。

```
/*
 *    $Header: amt.h,v 0.0 1992/12/20 05:59:15 kei Exp kei $
 */

struct amt {
    struct amt        *next;
    struct amt        *prev;
    struct in_addr    vip;
    struct in_addr    ip;
    u_int         ver;
    u_int         tm_idle;
    u_int         tm_delete;
    u_int         flags;
};

/* amt_flags */
#define    AMT_INUSE    0x01        /* busy flag */
#define    AMT_PERM     0x02        /* permanent */
#define    AMT_DEL       0x04        /* deleting */
```