マルチメディア通信と分散処理 78－14
（1996. 9. 27）

シームレスなIPv4/IPv6の相互通信方式

高村 真俊† 河部 展† 森島 直人† 門林 雄基‡ 山口 英†

†奈良先端科学技術大学院大学情報科学研究科
‡大阪大学大型計算機センター

現在、IPv4からIPv6へのスムーズな移行を行なうためにさまざまな考察が行なわれている。それらの考察の一つとしてIPv4/IPv6混在環境におけるネットワークの運用の技術があげられるが、IPv4ノードとIPv6ノード間の通信に関する考察はあまり行なわれていないのが現状である。

本稿では、現在提案されているIPv4/IPv6混在環境下での技術について考察し、それらの技術では触れられていないIPv4/IPv6の相互接続の具体的な方法を提案する。最終的にはIPv4/IPv6ノード間で制約なく、大規模なネットワークに耐え得る相互接続を目指している。

A seamless interconnection technique for IPv4/IPv6 internets

Masatoshi Takamura† Hiraku Kawabe† Naoto Morishima†
Youki Kadobayashi‡ Suguru Yamaguchi†

†Graduate School of Information Science, Nara Institute of Science and Technology
‡Computation Center, Osaka University

Despite many ongoing efforts on IPv4/IPv6 transition, few efforts are being made on communication between IPv4 and IPv6 nodes. In this paper, we identify limitations of current IPv4/IPv6 interconnection techniques. Then we describe our system which supports seamless interconnection of IPv4 and IPv6 networks. Our goal is to provide scalable communication between these two internet protocols without both any restrictions and sacrificing IPv4 address space for IPv4/IPv6 connectivity.

1 はじめに

今後のインターネットにおけるIPアドレスの新規割り当ては、大部分がIPv6のものになるとみられる。しかしながら、既存のIPv4環境からIPv6環境への移行には長期間を要すると考えられており、この間はIPv4とIPv6の混在環境が生まれる。これらの相互の資源の利用のために、両環境の相互接続をするしくみが求めらている。

RFC1933 [1]で提案されているトンネリングの技術は、IPv4ネットワークを中間に介して接続されているIPv6ノード間の通信を可能にしている。一方IPv4ノードとIPv6ノードの通信に関しては、IPv6ノード側が、IPv6のアドレス空間の一部に割り当てられた、IPv4-compatible IPv6アドレス [2]と呼ばれる空間を使用してのみ、通信可能である

IPv4-compatible IPv6アドレスとは、128bitのIPv6アドレスの下位32bitを利用して、従来のIPv4アドレスを表現できるようにしたものある。つまりIPv4-compatible IPv6アドレス空間はIPv4アドレス空間に等しい。IPv4ノードと通信

可能なIPv6ノードの数は、IPv4アドレス空間の制約をうけ、現実的な方法ではないと思われる。

本提案では、IPv6アドレスを通信時にIPv4アドレス空間に動的に写像することによって、制約なくIPv4ノードとIPv6ノードの通信を可能にする方法について述べる。

2 IP version6

IP(Internet Protocol)はインターネットを支える基盤プロトコルである。しかし現在使用されているIPversion4(以後IPv4)は、開発から既に20年をへて、IPアドレス数の不足、経路制御情報の肥大化などの問題を抱えるようになり、現状にそぐわないものとなってきた。IETF(Internet Engineering Task Force)では、このような問題を解決し、また将来需要が大きくなるであろう音声や動画のような大容量のデータ通信に対する要請に答える新プロトコルに関する議論がなされてきたが、最終的に1994年、IP version6(以後IPv6)として標準化された。

IPv6では、大きな変更点としてIPv4では32bitであったIPアドレスが、IPv6では128bitに拡張されたことがあげられる。またIPv4では積極的に使用されることの無かったフィールドが整理されるなどして、パケットヘッダの構造が大きく変更された[3]。

3 RFC1933で述べられている移行方式とその問題点

RFC1933で述べられている移行方式は，(図1)のようにIPv4ネットワークで接続されているIPv6ノード間の通信方法の提案である。

この提案を実現する方法として，"configured tunneling" と "automatic tunneling" という方法が提示されている。

configured tunneling とは、IPv4-compatible IPv6 アドレスを持たない IPv6 ノード間での通信を実現する方法である。トンネルの入口でカプセル化する際に，IPv4パケットのソースアドレスをトンネルの入口のノードのIPv4アドレスにし，トンネルの終端のノードのIPv4アドレスを終点

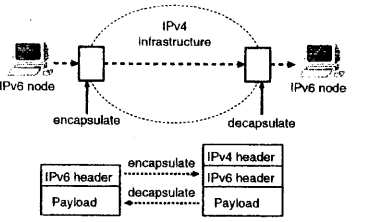

図 1: RFC1933のトンネリング

アドレスにする。この方法をもちいた場合，トンネルを構成する両端のノードの情報を相互に保持していなければならない。この情報の設定方法に関する具体的な方法は述べられていない。

automatic tunneling とは、IPv4-compatible IPv6アドレスを持つIPv6ノード間での通信を実現する方法である。IPv6パケットをIPv4パケットでencapsulate/decapsulateする際，IPv4パケットの始点および終点アドレスに，IPv6パケットの始点および終点アドレスの下位32ビットを使用する。この方法をもちいた場合はトンネルを構成する両端のノードの情報を保持する必要はない。しかしIPv4-compatible IPv6アドレスを使用するという方法では，IPv4のアドレス空間以上のノードは制御できない。

ここまで[1]で提案している移行方式について述べてきたが、この移行方式はあくまでも「IPv6ノード間の通信方式」であった。

しかし、世界中のIPv4ノードがあるとき一斉にIPv6ノードになるということは非現実的であり、すべての計算機がIPv6へ移行するまでにある程度の時間を要することを考慮すると、この移行方式だけでは不十分である。

なぜならその移行期間は、IPv4ノードとIPv6ノードが同様のサービスを受けることはできないし、互いに直接通信することもできないからである。

よってIPv4環境とIPv6環境が混在している環境下で、どのノードも制約を受けずに運用するためにはIPv4パケットとIPv6パケットの相互変換が必要であると考える。

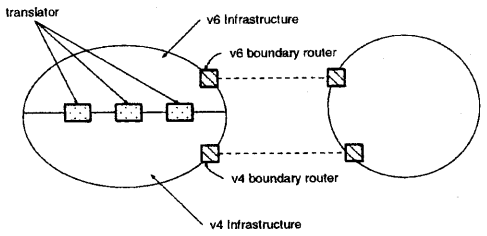

図 2: AS 内のトランスレータの位置づけ

4 シームレスな相互通信方式

本節では IPv4、IPv6 相互のノード間での制約のない通信を可能とする、パケット形式の変換をおこなうパケットトランスレータ (以下トランスレータ) の概要を述べる。

4.1 トランスレータ

ここで提案するトランスレータは AS (Autonomous System) を一つの単位として考える。(図 2) のように AS 内に IPv4 環境と IPv6 環境が存在することを想定し、これらの環境をこのトランスレータを介して相互接続することで、AS 内の IPv4-IPv6 間のパケット交換の要求を集中処理する。

AS 内の IPv4 ノードから As 外部の IPv6 ノード、あるいは AS 内の IPv6 ノードから AS 外部の IPv4 ノードへのパケットの送出は、AS 内でのトランスレートを経て行われる。

トランスレータではパケットを受け取ると、そのヘッダ内の、IPv4、IPv6 の両ヘッダ間で対応するフィールドをコピーし、新しいヘッダを作成して、古いパケットと置き換えて送出する。

ここでアドレスフィールドに関しては前述のように IPv4、IPv6 双方のアドレス空間が 1 対 1 に対応しないため単純にコピーすることはできない。そこでアドレスについては IPv4 → IPv6 のパケット転送と IPv6 → IPv4 のパケット転送、それぞれの場合に関して次のような動作をする。

(a) IPv6 ノードから IPv4 ノードへパケットを転送する場合

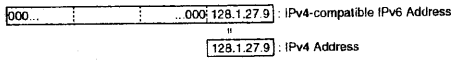

図 3: IPv4-compatible IPv6 アドレス

従来の IPv4 アドレス空間は、IPv6 のアドレス空間内では「IPv4-compatible IPv6 アドレス」として表現される。この IPv4-compatible IPv6 アドレスは、128bit の IPv6 アドレスの下位 32bit に IPv4 アドレスを挿入して、残りの bit を 0 で埋めたものである。したがってトランスレータは、受け取った IPv6 のパケットヘッダの終点アドレスのフィールドの下位 32bit を、そのまま送出する IPv4 パケットのヘッダ内の終点アドレスとして使用することができる (図 3)。

一方始点アドレスについては、通常使用されていない IPv4 空間のアドレスをこの IPv6 ノードに対して、IPv4 ノードから見たみかけのアドレスとして動的に割り当てることで解決する。

この割り当てをされた IPv4 アドレス (以降仮想アドレスと称する) を始点アドレスとしてトランスレータは IPv4 パケットを作成し送出する。

仮想アドレスとして使用する IPv4 アドレスは RFC1918 [4] で規定されている Private Address を使用する。

(b) IPv4 ノードから IPv6 ノードへパケットを転送する場合。

トランスレータは通信をしようとしている IPv4 ノードに対して、送信先となる IPv6 アドレスに対応する仮想アドレスを割り当てる。

IPv4 ノードでは、割り当てられた仮想アドレスを終点アドレスとして、IPv6 ノードに向けてパケットを送出する。このパケットを受け取った途中経路のトランスレータは、使用されている仮想アドレスに対応する IPv6 アドレスを使用して IPv6 パケットを作成し送出する。

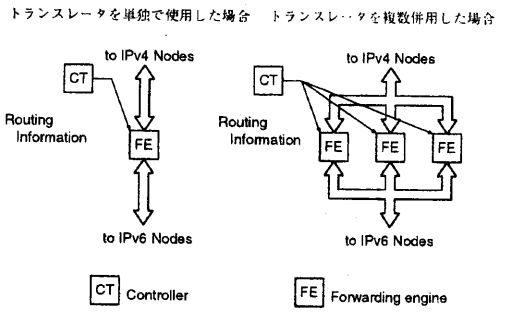

図 4: システム構成

トランスレータは、管理している仮想アドレスのプールから、接続要求ごとに仮想アドレスをひとつの接続に対して動的に割り当てる。トランスレータはそれぞれの接続についてパケットの流れを監視し、一定時間パケットが流れなければ通信が終了したとみなして、割り当てた仮想アドレスを回収しプールに戻す。戻された仮想アドレスは再利用される。

4.2 システム構成

トランスレータは、実際のパケットの変換をおこなう「パケット変換部 (forwarding engine)」と、仮想アドレスの動的割り当てをする「制御部」の、二つの論理的なブロックで構成される。制御部は同時にルータの機能も併せて持ち、IPv4、IPv6の両環境に対して、トランスレータへパケットを導くための経路情報を提供する (図 4)。

4.2.1 スケーラビリティとルーティング

トランスレータを利用する IPv4-IPv6 環境間の接続ノードの数が多くなり、負荷が大きくなる場合には、パケット変換部を並列に複数台配置することで負荷分散をすることができる。この場合はトランスレータはパケットを各パケット変換部に分散するような経路情報を生成する。

4.2.2 アドレスの相互変換

アドレスの動的割り当てに関しては、トランスレータに対応した DDNS(Dynamic DNS) との連係が必要となる。DDNS 自体には、アドレスを自動的に割り当てる機能はないが、リソース・レコードを更新する機能をもっているので、それにアドレスの自動割り当て機能を追加し、トランスレータに対応した DNS サーバを実現する。具体的な方法については 5章で詳しく述べている。

4.2.3 Path MTU について

IP ではパケット長を途中経路の MTU (Maximum Transmission Unit) に適合させるために、フラグメンテーションの機能が規定されている。IPv6 では途中経路でのフラグメント化が禁止されるなど、フラグメント化の機構が変更がなされた [3]。この点が、トランスレータの動作にどのように影響するかを考察する。

1. IPv6 環境→ IPv4 環境の通信

   IPv4 環境下では経路上の各ノードは、必要に応じてフラグメント化を行なうことができるので、IPv4 経路上の MTU についてはトランスレータは関知する必要はない。

2. IPv4 環境→ IPv6 環境の通信

   IPv6 においては経路途中でのフラグメント化は許されていない。そのため IPv4 から IPv6 への配送では、トランスレータが Path MTU Discovery プロトコルを用いて、トランスレータ以降の経路についての MTU を調べ、必要に応じてフラグメント化を行なう必要がある。

4.3 動作例

以下ではトランスレータが実際にどのような手順で動作するのかを、DDNS との連係を示しながら説明する。

- IPv6 ノードから IPv4 ノードへの通信

  1. IPv6 ノードは DDNS に対して、通信先となるホスト名に対応した IPv4 アドレスを要求する。
  2. DDNS は、IPv4-compatible IPv6 アドレスを IPv6 に返す。
  3. IPv6 ノードは IPv4-compatible IPv6 アドレスを終点アドレスとしてパケットを送信する。

4. パケット変換部は、受け取ったIPv6パケットのヘッダを取り外して、新しくIPv4パケットのヘッダを付加してIPv4ノードに向け送信する。このとき終点アドレスのIPv4-compatible IPv6アドレスは、上位96bitを外してIPv4アドレスに変えられる。また始点アドレスのIPv6アドレスは、仮想アドレスに置き換えられる。

- IPv4ノードからIPv6ノードへの通信

1. IPv4ノードはDDNSに対して、通信先となるホスト名に対応したIPv6アドレスを要求する。
2. DDNSは、得られたIPv6アドレスに対応した仮想アドレスを生成して、この仮想アドレスをIPv4ノードに対して返す。
3. (2)と同時に、DDNSはパケット変換部に対して対応表をわたす。
4. IPv4ノードは仮想アドレスを終点アドレスとしてパケットを送信する。
5. パケット変換部は、受け取ったIPv4パケットのヘッダを取り外し、新しくIPv6パケットのヘッダを付加して、パケットをIPv6ノードに向け送信する。このとき終点アドレスの仮想アドレスは、本来のIPv6アドレスに置き換えられ、始点アドレスは、対応したIPv4-compatible IPv6アドレスに変えられる。

5 Dynamic DNSとアドレスの動的割り当てについて

Dynamic DNSは、RFC1035で定めているDNSの機能に、「クライアントが追加/削除したいリソース・レコードをサーバに送り、サーバがそのリソース・レコードを追加/削除する機能」を追加したものである。[5]、[6]などでこれを実現する方法が述べられている。

Dynamic DNSでは、サーバが自動的にリソース・レコードを生成し、割り当てるといった機能はない。しかしDHCPとDNSとの連係で自動割り当てを実現する方法が[7]で述べられていてお

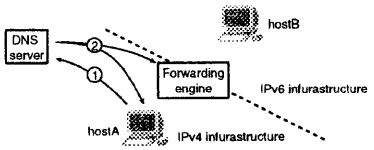

図5: 第一案でのDNSの構成

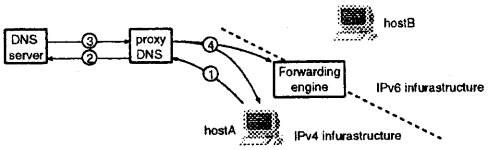

図6: 第二案でのDNSの構成

り、トランスレータの構成に必要なアドレスマッピングのしくみを考える上で非常に参考になった。

現在、トランスレータ用の仮想アドレスを割り当るしくみについて次の2つの案を考えている(図1)。

1. 第一案

   hostAがhostBについてDDNSサーバに問い合わせがあったとする。DDNSサーバは問い合わせを受けとった段階で、

   - hostBの仮想アドレスの自動割り当てを行なう。
   - リソース・レコードの更新を行なう。

   という2つ動作をする。よってこの案ではDDNSサーバがアドレスの自動割り当ての機能とDNSのリソース・レコードを更新する機能の2つを持たなくてはならない。実装を考えるとサーバのコードが複雑になる可能性がある。

   割り当てられた仮想アドレスとhostBのアドレスの組はパケット変換部に通知され、hostAには仮想アドレスが通知される(図1)。

2. 第二案

   第一案と同様の問い合わせがあったとする。

hostA と DDNS サーバの間を仲介する proxy DNS がアドレスの自動割り当てを行なう。

proxy DNS は、割り当てた仮想アドレスを追加するリソース・レコードとして DDNS サーバに送る。proxy DNS と DDNS サーバの間の送受信は [5] で提案されている方法を使用する。その後の通知に関する動作は proxy DNS が行なう。通知先と通知内容は第一案と同様である。

この案を使用すれば、DNS のリソース・レコードの追加/削除の機能は Dynamic DNS をそのまま使用でき、実装の面では第一案より優れているが、proxy DNS で仲介することによって、問い合わせに対する応答の遅延問題が発生する可能性がある。

6 予想される問題点

まず、IPv6 アドレスを仮想アドレスに写像するために Private Address を使用して良いかという問題がある。Private Address に対するルーティング情報は外部に流してはならないので AS boundary ルータで、ルーティング情報を隠蔽するなどの考慮が必要になる。

また、IPv6 アドレスと仮想アドレスの組の膨大なデータベースの制御方法については今後の大きな課題である。

7 まとめ

IPv6 の標準化後、既にいくつかのサンプル実装が発表されており、それらの接続実験が開始されるなど、現在は IPv6 への移行の準備期に入っている。

しかしながら具体的な移行方法が定まっておらず、長期間にわたって既存の IPv4 環境が存続するものと考えられることから、その移行期間中の IPv4-IPv6 間の相互接続性を確保することが重要である。

本稿では、現在提案されている方法での IPv4 から IPv6 への移行期における相互接続性の問題を指摘し、その解決方法としてのパケットトランスレータを提案した。またこのトランスレータと協調する DDNS(Dynamic Domain Name Service) の拡張に関して考察した。

今後、NetBSD を基盤 OS として実装を行い、IPv4-IPv6 環境間での相互接続性をテストする。

参考文献

[1] R. Gilligan, et al, "Transition Mechanisms for IPv6 Hosts and Routers", RFC1933, April 1996.

[2] R.Hinden, et al, "IP Version 6 Addressing Architecture", RFC1884, December 1995.

[3] S. Deering and Hinden. "Internet Protocol, Version 6 (IPv6) Specification." RFC1883, December 1995.

[4] Y.Rekhter, et al, "Address Allocation for Private Internets.", RFC1918,February 1996.

[5] P. Vixie (Ed), et al, "Dynamic Updates in the Domain Name System." Internet Draft(draft-ietf-dnsind-dynDNS-09.txt), March 1996.

[6] P.Vixie (Ed), "Deferred Dynamic Updates in the Domain Name System." Internet Draft (draft-ietf-dnsind-defupd-00.txt), May 1996.

[7] Yakov Rekhter, "Interaction between DHCP and DNS", Internet Draft(draft-ietf-dhc-dhcp-dns-01.txt), February 1996.