

IPv4 と IPv6 のトランスレータに関する考察

山本和彦
奈良先端科学技術大学院大学
kazu@mew.org

角川宗近
(株)日立製作所
sumikawa@ebina.hitachi.co.jp

島慶一
シャープ(株)
shima@slab.tnr.sharp.co.jp

要約

6bone の運用実験に象徴されるように、IPv4 から IPv6 への移行が始まっている。移行の後期では IPv4 ホストと IPv6 ホストが混在するため、現在の技術では両者間の通信を実現できないと考えられる。そこで本稿では、IPv4 ホストと IPv6 ホストの通信を実現するためのトランスレータについて考察する。まず、トランスレータの本質的な構成要素を明らかにした後、それぞれの構成要素に利用できる技術について検討する。この検討を踏まえて、導入のコストが低く汎用的なトランスレータを提案する。

Study on Translator for IPv4 and IPv6

Kazuhiko Yamamoto
Nara Inst. of Sci. and Tech.
kazu@mew.org

Munechika Sumikawa
Hitachi, Ltd.
sumikawa@ebina.hitachi.co.jp

Keiichi Shima
SHARP Corp.
shima@slab.tnr.sharp.co.jp

Asbtract

As symbolized by the deployment of 6bone, transition from IPv4 to IPv6 has started. Since IPv4 hosts and IPv6 hosts will co-exist in the late stage of the transition, it will be impossible for them to communicate each other. This paper thus focuses translator technology which enables such communication. We first clarify essential components of the translator then discuss available technologies for each component. With this study we propose a cost-effective and generic translator.

1 はじめに

IPv4[1] アドレスの枯渇問題を解決するために、今後インターネットはIPv4からIPv6[2]へ円滑に移行していく必要がある。移行初期である現在は、IPv6の実験バックボーンとして6boneが実験運用されている。ここでは、IPv4インターネットという大海において孤立したIPv6の島をIPv6 in IPv4トンネル[3]を使って接続することで、IPv6ホスト同士の通信を可能にしている。また、初期のIPv6ホストはIPv4の実装も義務づけられているため、IPv4を利用すればIPv4ホストとも通信できる。

しかし、移行の後期では、生産コストや保守のコストを下げるためにIPv6のみのホストが登場する。同様に、バックボーンにおいてもIPv4アドレスを持たないルータが現れる。この時点でIPv4とIPv6の立場は逆転し、IPv6インターネットという大海にIPv4の島が点在するようになる。IPv4ホスト同士の通信を実現するためには、IPv4の島をIPv4 in IPv6トンネルを使って接続すればよい。しかしながら、IPv4ホストとIPv6ホスト間の通信は、現在提案されている技術だけでは不可能である。

IPv4ホストは長期に渡って残り続けるであろう。IPv4インターネットで築かれた豊富な資源をIPv6ホストから利用可能し、逆にIPv6インターネットで提供されるサービスをIPv4からアクセスできるようにすることは重要な課題である。そこで、本研究では「IPv4ホストとIPv6ホストの相互接続の実現」のために利用できる技術について考察し、それらを組み合わせる方法を提案する。

2 トランスレータの構成要素

トランスレータは、導入コストが低く、高速で、かつさまざまな組織に適応できる汎用性を持つ必要がある。この節ではこの条件を踏まえてトランスレータの本質的な構成要素について考察する。

導入コストを低減するには、IPv4とIPv6の変換をIPv4の島とIPv6インターネットを結ぶゲートウェイで一括して行うのが最もよい。

これを「変換ゲートウェイ」と呼ぶ。変換ゲートウェイは十分に高速でなければならない。また、複数の変換ゲートウェイが必要な組織もあるだろう。

IPv4ホストとIPv6ホストが通信を開始する際には、多くの場合終点の識別子としてホスト名が与えられる。ホスト名からはIPv4ホストとIPv6ホストの区別はつかない。ここで仮に、IPv4ホストがあるホスト名からアドレスを求めたとする。IPv6アドレスが返された場合は、IPv4ホストはそのホスト名で識別されるホストとは通信できない。そこで、ホスト名からアドレスへの変換機構は、問い合わせ元のホストが扱えるアドレスをなんらかの方法で入手し返答する必要がある。この機能を実現するシステムを「ホスト名サーバ」と呼ぶ。

変換ゲートウェイやホスト名サーバは、IPv4とIPv6の対応表を共有すべきである。また、IPv4からIPv6への対応、あるいはその逆は、他の通信に害を与えないように管理される必要がある。アドレスを対応付け、そして、変換ゲートウェイやホスト名サーバに対応表を提供するシステムを「アドレス・マッパー」と呼ぶ。

以上から、トランスレータの本質的な構成要素は、変換ゲートウェイ、ホスト名サーバ、アドレス・マッパーであると分かった。これらの関係を図1に示す。

3 利用できる技術

この章では、変換ゲートウェイ、アドレス・マッパー、および、ホスト名サーバに対し利用できる技術について説明する。

3.1 変換ゲートウェイ

変換ゲートウェイに利用できる技術としては、ヘッダ変換、トランスポート・リレー、および、代理サーバがある。それぞれ、ネットワーク層、トランスポート層、および、アプリケーション層のゲートウェイに相当する。以下それぞれについて解説する。

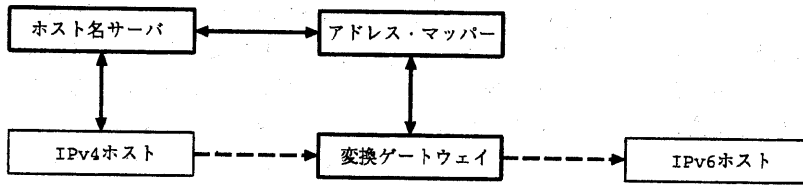


図 1: トランスレータの構成要素

3.1.1 ヘッダ変換

ヘッダ変換とは、その名前の通りネットワーク層レベルの通信単位であるデータグラムに対して、ペイロード部分をそのまま残し、ヘッダ部分を変換することである。同種のヘッダへの変換と異種のヘッダへの変換とに大別される。同種のヘッダへの変換の例としては、IPv4 ヘッダを IPv4 ヘッダに変換する NAT[4] が挙げられる。異種のヘッダへの変換にはさまざまな組合せがあるが、身近なものとしては IPv4 と AppleTalk とを変換する FastPath や GatorBox がある。

ヘッダ変換の長所としては、通常のルータにおけるパケット転送速度にヘッダ変換のコストが付加された程度の速度が出せることである。欠点としては、ネットワーク・プロトコルで規定されている制御やエラーのセマンティクスを完全には変換できないことである。たとえば、IPv4 の制御プロトコル ICMP[5] には始点のデータ転送速度を制御する source quench が定義されているが、IPv6 用の ICMPv6[6] には対応する機能はない。さらに、ネットワーク・プロトコルに依存するデータがペイロード中に埋め込まれている場合には対応できない。たとえば、IPv4 と IPv6 用の FTP[7] は、それぞれ IPv4 と IPv6 アドレスを交換する。

3.1.2 トランスポート・リレー

トランスポート・リレーとは、TCP[8] などのコネクション・オリエンテッドな通信において、途中のゲートウェイが始点からのコネクションを一旦終了し、終点へ別のコネクションを開いてデータ・ストリームをリレーする方式である。トランスポート・リレーでは、ヘッダ変換の場

合に問題となったセマンティクスは完全に保存される。ただし、単純なデータ・ストリームのリレーでは、ネットワーク・プロトコルに依存するデータを変換できない。また、変換ゲートウェイ上でトランスポート機能を実現するためのコストが付加されるために、ヘッダ変換に比べると速度が落ちる。

3.1.3 代理サーバ

代理サーバとは、あるサービスにおいて始点のクライアントが終点のサーバに直接アクセスするのではなく、途中のサーバにサービスを代行してもらう方法である。たとえば、最近では HTTP[9] 用の代理サーバが広く用いられている。

代理サーバが複数のネットワーク・プロトコルに対応すれば、ヘッダ変換やトランスポート・リレーの際に問題になったプロトコル依存のデータの問題は解決される。しかしながら、この方式にはトランスポート層とアプリケーション層の両方のコストが加わるため、トランスポート・リレーよりも速度は遅い。また、サービス毎にサーバを改造しなければならないという問題がある。

3.2 アドレス・マッパー

IPv4 から IPv6 へのアドレス変換は、組織に割り当てられた IPv6 アドレスの一部を利用すればよい。また、IPv6 から IPv4 へのアドレス変換は、IPv4 のプライベート・アドレス[10] を利用する方法が考えられる。前者の変換は静的であり、後者は動的である。よって、アドレス・マッパーは動的な後者の変換機能のみ提供

すればよい。つまり、一貫性を保ちながら IPv6 アドレスに対し IPv4 のプライベート・アドレスを割り当てる役割を負う。前者の変換は、ホスト名サーバや変換ゲートウェイにあらかじめ組み込める。

3.3 ホスト名サーバ

この小節では、ホスト名からアドレスを求めサービスにおいて、利用可能なシステムであるネームサーバと SOCKS 5 について述べる。

3.3.1 ネームサーバ

DNS[11] は、ホスト名とアドレスの対応を管理する広域分散データベースである。インターネットのほとんどのホストが、ネームサーバに対応している。ホスト名から IPv4 アドレスを検索するには、検索項目として A レコードを指定する。IPv6 アドレスの検索項目には AAAA レコード [12] が定められた。

拡張ネームサーバは、IPv4 ホストのホスト名に対する A レコードの検索要求と IPv6 ホストのホスト名に対する AAAA レコードの検索要求に対しては通常通りに振舞う。これによって従来のネームサーバと互換性を保つ。

IPv6 ホストのホスト名に対し A レコードの検索が要求された場合には、まず AAAA レコードを検索して IPv6 アドレスを求める。そして、アドレス・マッパーにこの IPv6 アドレスに対応する IPv4 アドレスを割り当ててもらいこれを返答する。IPv4 ホストのホスト名に対し AAAA レコードの検索が要求された場合には、その組織の IPv6 アドレスの一部に変換して返す。

3.3.2 SOCKS 5

SOCKS は、トランスポート・リレーを利用した防火壁技術である。SOCKS ホストは、まず SOCKS サーバに終点の識別子としてアドレスとポートを指定し、SOCKS サーバが扱える中継用のアドレスとポートを割り当ててもらふ。次に、SOCKS ホストが中継用のアドレスとポートに対してコネクションを張れば、SOCKS

サーバがこのコネクションを受け取る。SOCKS サーバは、最初に与えられた終点のアドレスとポートに対しコネクションを確立し、2 つのコネクションをリレーする。

SOCKS 5[13] では、終点の識別子としてホスト名を使えるようになった。よって、SOCKS 5 サーバは、IPv4 でホスト名を指定された場合、ホスト名が識別するホストが IPv6 ホストであっても、中継用に IPv4 アドレスを割り当てて返答するよう改造可能である。しかしながら、ホストが SOCKS 5 に対応するにはコマンドやライブラリなどの入れ換えが必要である。

4 トランスレータの提案

アドレス・マッパーの実装では、ホスト名サーバ、あるいは、変換ゲートウェイにその機能を持たせる方法が考えられる。しかし、複数の変換ゲートウェイが存在する環境に適応できるよう独立したサーバとすることを提案する。

ホスト名サーバとしては、現在の IPv4 ホストのほとんどが対応しているネームサーバを拡張して使用することを提案する。

代理サーバ方式はサービスごとにサーバを拡張する必要があるため、実装のコストが高い。そこで、変換ゲートウェイとして仮にヘッダ変換方式を採用した。

ここで、我々が提案したトランスレータにおけるホスト名サーバ、アドレス・マッパー、変換ゲートウェイが協調動作について、IPv4 から IPv6 への通信を例にとって説明する。

IPv4 から IPv6 への通信は図 2 に示す手順に従う。まず、IPv4 ホスト $H4$ は、拡張ネームサーバに $H6$ に対する A レコードの検索を要求する (1)。拡張ネームサーバは、A レコードの検索に失敗するので、次に AAAA レコードを調べ (2)、 $H6$ の IPv6 アドレス $AddrH6$ を得る (3)。さらに、 $AddrH6$ をアドレス・マッパーに送る (4)。

アドレス・マッパーは $AddrH6$ に対して、IPv4 のプライベート・アドレス $Paddr41$ が割り当てられていればそれを、さもなければ新たに $Paddr41$ を割り当てて拡張ネームサーバへ返

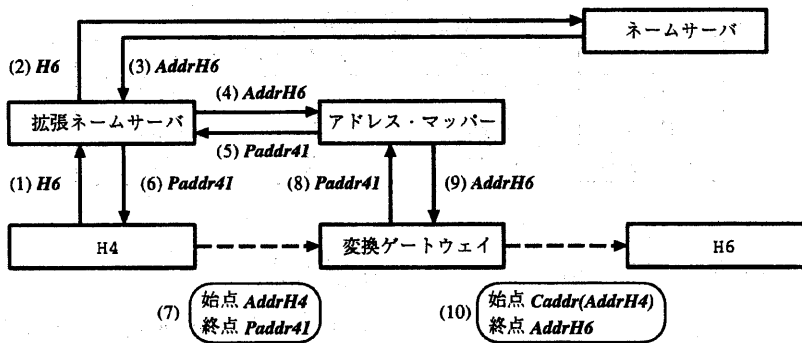


図 2: IPv4 から IPv6 への通信

す (5)。そして、拡張ネームサーバは受け取った *Paddr41* を *H4* へ回答する (6)。

H4 は、始点が *H4* のアドレス *AddrH4*、終点が *Paddr41* である IPv4 パケットを作成し送信する (7)。この IPv4 パケットは、組織とインターネットの接続点である単数あるいは複数の変換ゲートウェイの内 1 つに届く。IPv6 パケットを受け取った変換ゲートウェイは、終点がプライベート・アドレスであるためヘッダ変換が必要であると判断する。

そこで、変換ゲートウェイは *Paddr41* に対応する実際の IPv6 アドレスをアドレス・マップパーに問い合わせる (8)。アドレス・マップパーは、*AddrH6* を返す (9)。始点の *AddrH4* に関しては、予約していた IPv6 アドレス *Caddr6(AddrH4)* に変換する。他のフィールドにも適切な変換を施された後、変換ゲートウェイからは始点が *Caddr6(AddrH4)*、終点が *AddrH6* である IPv6 パケットが出力される (10)。

この IPv6 パケット受け取った *H6* が *H4* へ IPv6 パケットを送る場合は、始点に *AddrH6*、終点に *Caddr6(AddrH4)* を指定すればよい。*H6* からの IPv6 パケットを受け取った変換ゲートウェイは終点が予約していた IPv6 アドレスであるために変換が必要であると分かる。この変換ゲートウェイが上記の変換ゲートウェイであれば、それまでの情報を元に変換が可能である。他の変換ゲートウェイであれば、まず終点の *Caddr6(AddrH4)* を単純に *AddrH4* へ変換

する。また、始点の *AddrH6* をアドレス・マップパーに問い合わせれば *Paddr41* を得られるため、ヘッダの変換が可能である。

詳細は省略するが、IPv6 ホストから IPv4 ホストへの通信も同様に可能である。

5 トランスレータの試作

我々が提案したトランスレータを試作し動作を確認した。アドレス・マップパーはスクラッチから書き、通信プロトコルとしてはカーネルが扱いやすい ICMP/ICMPv6 を拡張して用いた。ホスト名サーバには、慶應義塾大学の土井裕介氏が開発したネームサーバ *yans* を拡張することで実装した。変換ゲートウェイとしては、我々が BSD/OS を IPv6 用に拡張しているコードに、ヘッダ変換の機能を追加して作成した。動作させた結果、期待通り IPv4 と IPv6 の通信が実現できた。

また、変換ゲートウェイの速度を評価するために、TCP を用いて 1000 バイトのパケットを 10000 個転送にかかる時間をカーネル内で計測した。結果を表 1 に示す。第 1 行と第 2 行は、それぞれ IPv4 パケットと IPv6 パケットを単純に転送する際に要する時間である。第 3 行はヘッダ変換方式により IPv4 パケットを IPv6 パケットに変換して転送する際にかかる時間である。列はユーザ空間で動いているプロセス数、つまり、ゲートウェイの負荷を示している。表 1 から分かるように、ヘッダ変換方式は単純な

表 1: 転送速度 (単位は秒)

	プロセス無し	プロセス1つ	プロセス2つ
IPv4	15.71	16.21	16.26
IPv6	16.28	16.97	17.19
ヘッダ変換	17.30	17.73	17.88

転送と遜色無い速度が出ている。

6 おわりに

IPv4 から IPv6 への移行後期には IPv4 ホストと IPv6 ホストが混在するため、現在の技術では両者の通信が実現できない。そこで、IPv4 ホストと IPv6 ホストの通信を実現するためのトランスレータについて考察した。

トランスレータの本質的な構成要素として、アドレス・マッパー、ホスト名サーバ、および、変換ゲートウェイが必要であることを示した。アドレス・マッパーは、さまざまな組織にトランスレータが導入可能となるよう独立して実装することを提案した。また、ホスト名サーバはネームサーバを拡張して用いることを提案した。変換ゲートウェイにはヘッダ変換ゲートウェイを仮に利用し、十分な高速性を得られた。

トランスレータを試作して実験したところ、IPv4 と IPv6 の通信が実現できることが確かめられた。しかし、ヘッダ変換方式ではネットワーク・プロトコルに依存したデータをやりとりするアプリケーション・プロトコルに対応できない。今後は、この問題を解決するためにトランスポート・リレーをアプリケーション・プロトコルまで拡張する方法に取り組んでいく予定である。

謝辞

本研究に関し有意義な議論をして頂いた陣崎明氏と小林伸治氏に感謝する。

参考文献

- [1] J. Postel, "Internet Protocol", RFC 791, 1981.
- [2] S. Deering and R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification", RFC 1883, 1996.
- [3] R. Gilligan and E. Nordmark, "Transition Mechanisms for IPv6 Hosts and Routers", RFC 1933, 1996.
- [4] P. Francis and K. Egevang, "The IP Network Address Translator (Nat)", RFC 1631, 1994.
- [5] J. Postel, "Internet Control Message Protocol", RFC 792, 1981.
- [6] A. Conta and S. Deering, "Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (IPv6)", RFC 1885, 1996.
- [7] J. Postel and J. Reynolds, "File Transfer Protocol", RFC 959, 1985.
- [8] J. Postel, "Transmission Control Protocol", RFC 793, 1981.
- [9] R. Fielding, J. Gettys, J. Mogul, H. Frystyk and T. Berners-Lee, "Hypertext Transfer Protocol - HTTP/1.1", RFC 2068, 1997.
- [10] Y. Rekhter, R. Moskowitz, D. Karrenberg, G. de Groot and E. Lear, "Address Allocation for Private Internets", RFC 1918, 1996.
- [11] P. Mockapetris, "Domain names - implementation and specification", RFC 1035, 1987.
- [12] S. Thomson and C. Huitema, "DNS Extensions to support IP version 6", RFC 1886, 1996.
- [13] M. Leech, M. Ganis, Y. Lee, R. Kuris, D. Koblas and L. Jones, "SOCKS Protocol Version 5", RFC 1928, 1996.