

# 仮想計算機におけるソケットアウトソーシングによる IPv4/IPv6 トランスレータ

大橋 宏樹<sup>†</sup> 齊 藤 剛<sup>††</sup> 新 城 靖<sup>††</sup>  
佐 藤 聡<sup>††</sup> 中 井 央<sup>†††</sup> 板野 肯三<sup>††</sup>

## 1. はじめに

近年、IPv4 アドレスの枯渇問題の深刻さが増しており、枯渇する事態は近く避けられない。この問題解決のため IPv6 の導入が急がれている。しかし、現在の一般的なネットワークプログラミングにおいて、IPv6 普及の遅れと IPv4/IPv6 コンパチブルな実装・テストにかかるコストの高さから、IPv4 のみを前提に開発がなされることが多い。将来的に IPv4 ネットワークから IPv6 ネットワークへ移行した場合、大量のソフトウェアがメンテナンスされないまま残されることが考えられる。そのような移植がなされないレガシーなソフトウェアを活用し続ける方法があれば過去の資産を有効利用できる。

そこで本研究ではアウトソーシングという手法を用いてこの問題を解決する。アウトソーシングとは、仮想計算機の処理を実機上の OS に対して委譲することにより処理の高速化と機能拡張を行える手法である。本研究では、この手法を仮想計算機にモジュールを追加して機能拡張を行うために用いる。このモジュールとして、IPv4 と IPv6 の相互変換を行う機能を実装することにより、レガシーなソフトウェアにアプリケーションレイヤでの修正を行うことなく仮想計算機上で動作させることができる仕組みを提案する。

## 2. 仮想計算機におけるソケットアウトソーシング

本研究ではホスト型仮想計算機を用いる。実機上で動作する OS をホスト OS と呼び、ホスト OS の機能を用いてホスト型仮想計算機が動作する。この仮想計算機上で動作する OS をゲスト OS と呼ぶ。

このホスト型仮想計算機ではホスト OS とゲスト OS で機能の重複が起こるといった問題がある。このため、ホスト OS とゲスト OS で同様の処理を 2 度行うのでオーバーヘッドが大きくなる。この問題を解決するために本研究室ではアウトソーシングという手法を提案している。アウトソーシングでは、ゲスト OS の高レイヤのモジュールに修正を行い、その処理をホスト OS に任せる。アウトソーシングの適用対象としてネットワーク処理があり、これをソケットアウトソーシングという<sup>1),2)</sup>。ソケットアウトソーシングでは、ゲスト OS のソケット API レベルでの高水準な処理をホスト OS に委譲することでネットワーク I/O の高速化を実現している。

ソケットアウトソーシングは、ホスト OS はゲスト OS の通信をソケット API レベルで容易に監視できるので、ゲスト OS が明示的にソケットに対して行う操作内容を知ることができる。たとえば、socket システムコールや connect システムコールの引数を得ることができる。本研究では、この機能を利用し、ゲスト OS の通信に対して変換を施す IPv4/IPv6 トランスレータの機能を有するモジュールを実装する。

## 3. IPv4/IPv6 トランスレータ

本研究では IPv6 非対応の IPv4 ソフトウェアを IPv6 ネットワーク上の計算機で使用することを想定する。仮想計算機上で動作するゲスト OS は IPv4 でのみ通信が可能であるとする。ゲスト OS が外部の IPv6 ホストと通信するために、ソケットアウトソーシングを利用するモジュールとして IPv4/IPv6 トランスレータの実装を行い、IPv4-IPv6 のアドレス変換を可能にする (図 1)。IPv4/IPv6 トランスレータは仮想計算機からの通信を受け付けるホストサーバモジュールとアドレス解決を行う DNS キャッシュサーバからなる。

ゲスト OS 内でクライアントを実行する場合は、クライアントはドメイン名から IPv4 アドレスを取得するために A レコードを含む DNS クエリを発行する。しかし、外部の DNS サーバは AAAA レコードの IPv6 アドレスのみを保持しており、IPv6 のアドレスのみ名前解決を行える。本研究では、IPv4/IPv6 トランスレータと協調して動作する DNS キャッシュサーバに A レコードを含む DNS クエリを送る。

DNS キャッシュサーバは外部の DNS サーバに AAAA レコードを含む DNS クエリを送り、IPv6 アドレスを取得する。次に、ダミーの IPv4 アドレスを生成し、それとのマッピングをした後キャッシュを行う。その後、ゲスト OS 上のクライアントにダミーの IPv4 アドレスを返す。ゲスト OS 上のクライアントはこのダミーアドレスに対して接続するためのソケットを作成しようとするが、アウトソーシングによりゲスト OS 上にソケットは生成されず、ホストサーバモジュールに制御が移る。ホストサーバモジュールがダミーの IPv4 アドレスから IPv6 アドレスを得て IPv6 ソケットを作成する。

ゲスト OS でサーバを実行する場合、ゲスト OS のサーバがサーバ用ソケットを作成する要求を出すと、ホストサーバモジュールが要求を受け取る。ホストサーバモジュールは

<sup>†</sup> 筑波大学第三学群情報学類

<sup>††</sup> 筑波大学システム情報工学研究科

<sup>†††</sup> 筑波大学図書館情報メディア研究科

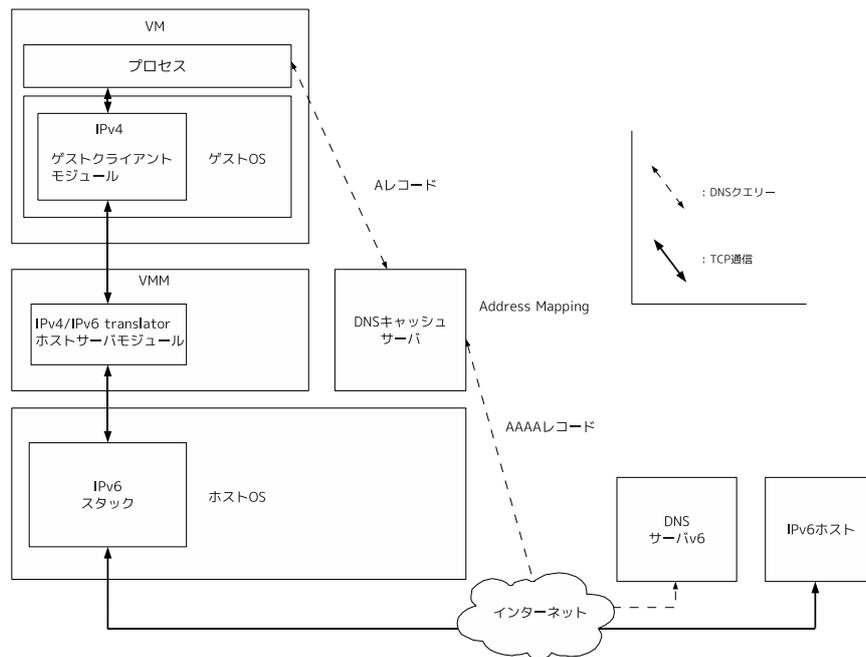


図 1 IPv4/IPv6 トランスレータ

IPv6 サーバ用ソケットを作成し、ゲスト OS 上のサーバの代わりに待ち受けを開始する。外部からホストサーバモジュールに対し接続要求が来ると待ち、ゲスト OS に通信する。

#### 4. 関連研究

Transport Realy Translator (TRT) 方式の IPv6-to-IPv4 トランスレータとして、BSD 系 OS 固有のインターフェースである *faith* デバイス、*faithd* サーバ、及び DNS プロキシ *totd* の協調動作により変換を行う手法がある<sup>3)</sup>。この研究では、*faithd* 用に予約された 96 ビットアドレスプリフィックスにより、*faithd* が IPv6 TCP パケットから IPv4 TCP パケットへの変換を行う。これに対して、本研究では、ホスト OS に割り当てた IPv6 アドレスを用いることが異なる。

#### 5. おわりに

仮想計算機におけるソケットアウトソーシングによる IPv4/IPv6 トランスレータについて述べた。今後は、Unbound、DNSMasq、*totd* などの DNS キャッシュサーバについて評価を行い、これを用いてアドレスマッピングの機能を実装する。また、ホストサーバモジュールの実装を行い、性能評価を行いたい。

#### 参考文献

- 1) Hideki Eiraku, Yasushi Shinjo, Calton Pu, Younggyun Koh, and Kazuhiko Kato: "Fast networking with socket-outsourcing in hosted virtual machine environments", 24th ACM Symposium on Applied Computing (SAC2009), pp.310-317, 2009.
- 2) 齊藤 剛, 新城 靖, 榮樂 英樹, 佐藤 聡, 中井 央, 板野 肯三: "仮想計算機におけるアウトソーシングのためのゲスト-ホスト間 RPC", 第 20 回コンピュータシステムシンポジウム, ポスター・デモセッション, 2008.
- 3) J. Hagino, K. Yamamoto: "An IPv6-to-IPv4 Trans-

port Relay Translator", RFC3142, 2001.