

# NAT64/DNS64 環境下における トランスレーションの失敗における回避方法の設計と実装

佐原 弘太郎

伊藤 暢浩

愛知工業大学

## 1 はじめに

研究目的の ARPANET から端を発したインターネットは、商用化に伴い爆発的に普及したためインターネット上でコンピュータを識別するための IP アドレスが枯渇している。このため、現在主流のプロトコルである IPv4 に代わるものとして、IPv6 が提唱されており移行が実現すれば IP アドレスの枯渇問題は解決されるとされている。しかし、IPv4 と IPv6 には互換性がないため、IPv6 に移行が完了するまでの技術として相互通信を確保するトランスレータが必要である。その代表的な実装として、NAT64/DNS64 が挙げられる。しかし、特定の環境及び条件において NAT64/DNS64 の問題により接続元の IPv6 ネットワークから接続先の IPv4 ネットワークへの変換に失敗する可能性があることが知られている。そこで本研究では DNS64 を用いない手法としてクライアントにトランスレータを実装し評価する。

## 2 インターネットプロトコル

### 2.1 Internet Protocol Version 4 について

IPv4(Internet Protocol Version 4) は、現在のインターネットで利用されているインターネットプロトコルである。 $2^{32}$  個= $4.3 \times 10^9$  個のアドレスが使用できるが、普及が進みアドレスが枯渇することが予想された [1]。問題を解決するために CIDR(Classless Inter Domain Routing) 及びプライベートアドレス、NAT(Network Address Translation)、NAPT(Network Address and Port Translation) がそれぞれ規格化され IPv4 アドレスの節約が可能となっているが、それを上回るペースでの枯渇が進んでいる [1]。

### 2.2 Internet Protocol Version 6 について

IPv6(Internet Protocol Version 6) はインターネットプロトコルの次世代規格であり、現在主流の IPv4 に代わるものとして制定された。先述した IPv4 の枯渇問題などを踏まえて設計されており、大きな特徴としてアドレス空間は  $2^{128}$  個= $3.4 \times 10^{38}$  個ものアドレスが使用できることがあげられる。広大なアドレス空間を用いることにより、すべてのノードが IPv6 ネットワーク上で一意のアドレスを持つことが可能となった。

### 2.3 NAT64/DNS64 について

NAT64/DNS64[2][3] はトランスレータの規格であり、IPv4 から IPv6 への移行後期の技術として策定されている。よって、IPv6 ノードから IPv4 ノードに対して開始されたセッションのみ変換を行うのが特徴である。NAT64/DNS64 の概略を図 1 に示す。

クライアントノードの DNS 問合せを DNS64 機能を備えた DNS キャッシュサーバ (以下 DNS64 キャッシュサーバ) によって対象の IPv4 アドレス 32 ビットを IPv6 アドレス下位 32 ビットに組み込んだアドレス

(以下 NAT64 射影アドレス) へと上書きし、NAT64 トランスレータへと誘導する。NAT64 トランスレータは IP パケットの変換を行い IPv4 ネットワークに属するノードへと接続を行う機構である。

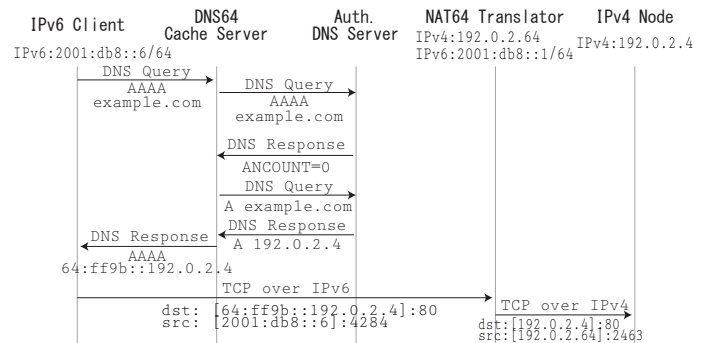


図 1: NAT64/DNS64 環境下でのクライアント接続順序 [2] [3]

### 2.4 NAT64/DNS64 による問題点

NAT64/DNS64 は 2 つの機能の組み合わせにて IPv6 ネットワークから IPv4 ネットワークへと変換する。しかし、NAT64/DNS64 は次のような問題があり変換に失敗することが知られている。

#### A. NAT64 による問題

パケットのデータ内部に IP アドレスを入れるプロトコル (FTP、IRC 等) には対応できない

#### B. DNS64 による問題

DNS の問合せを行わないプログラムや、DNS を用いない名前解決方法に対応できない

## 3 問題の解決手法の提案と検討

2.4 節で説明した問題を解決するための手段を検討した結果、クライアントノードにアドレス変換をおこなうトランスレータを実装し、NAT64 トランスレータノードと併用する方式とする。次に理由を述べる。

2.4 節 A. の解決手段として、NAT64 トランスレータに問題が生ずる各プロトコルごとの変換機能を実装する手法が考えられる。しかし、プロトコルごとに実装するのは非効率的であり、また流れるデータを走査し必要に応じて書き換える必要があると考えられるため処理コストも高い。提案方法の場合、クライアントにトランスレータを置くことからプロトコルは選ばずに変換出来るため、有利であると考えられる。また、NAT64 トランスレータに各プロトコルごとの変換機能を実装したとしても 2.4 節 B. の問題が解決されない。したがって、提案手法が最も適切であると考えられる。

## 4 クライアントへのトランスレータの実装

実装のベースプラットフォームとして、クライアント用として普及している Ubuntu や MintLinux の派生元である Debian GNU/Linux [4] を使用する。トランスレー

<sup>0</sup>A design and implementation of the evade avoid method in failure transration under NAT64/DNS64 environment.

<sup>0</sup>Koutaro SAHARA Nobuhiro ITO

<sup>0</sup>AICHI INSTITUTE OF TECHNOLOGY

タは高速化を図るためカーネルモジュールとして実装することとし、仮想的なインターフェイスを作成するためにLinuxカーネルに標準で含まれているTUN/TAPデバイスを使用する。トランスレータの構成要素として、以下の3点を実装した。

1. TUN 仮想デバイス入出力部
2. ネットワーク層変換部 (IPv4-IPv6 の変換)
3. トランスポート層変換部 (TCP/UDP の変換)

## 5 トランスレータの評価実験と考察

本研究で実装したクライアントのトランスレータの性能評価として、TCPのスループットおよびCPU処理時間、UDPのスループットによる性能評価を行う。比較対象としてNAT64/DNS64を用いた方式と比較を行う。計測に用いる環境はクライアントノード、NAT64トランスレータノード共に次の通りである。

- CPU : AMD Phenom II X6 1055T(2.8GHz x 6Core)
- メモリ:8192MByte
- OS:Debian GNU/Linux 6.0.1 (2.6.32-5-686)
- NIC : Realtek RTL8168B (GigaEthernet)

スループット計測に使用するソフトウェアには、NLANR(National Laboratory for Applied Research)が開発したTCP/UDP帯域測定ツールであるIperf [5]を使用する。計測時はinit,sshd以外のデーモンを停止する。LANはGigabit対応HUBにクライアントとサーバ、監視用ノートPCの3台のみ接続する。

### 5.1 TCP性能

クライアントノードからNAT64トランスレータノードへのTCPスループットおよびCPU処理時間を測定する。TCPウィンドウサイズは送信側、受信側共に64Kバイト、測定時間は60秒とし100回測定を行い平均値を取る。カーネルモジュールとして実装したトランスレータの所要時間を計るためカーネルランドのシステム時間をBashシェルのtimeコマンドを利用し測定する。表1は本研究の提案手法の、表2はNAT64/DNS64を用いた実験結果である。

表1: 本研究の提案手法によるTCPスループットおよびシステム時間

	平均	標準偏差
スループット	928.6Mbps	0.019Mbps
システム時間	0.611msec	0.0089msec

表2: NAT64/DNS64によるTCPスループットおよびシステム時間

	平均	標準偏差
スループット	928.7Mbps	0.015Mbps
システム時間	0.446msec	0.0060msec

表1,2よりスループットの低下は見られず、同等の性能が発揮出来ることがわかる。また、カーネルランドの処理時間であるシステム時間も平均0.164msecしか増加してないため十分実用的であると考えられる。

### 5.2 UDP性能

クライアントノードからNAT64トランスレータノードへのUDPスループットを測定した。図2は計測結果である。クライアントノードは長さが一定のUDPデータを一定のスループットで60秒間送信し、NAT64トランスレータノードは実際に受信出来たデータから受信スループットを算出している。送信するデータ長は512バイトである。

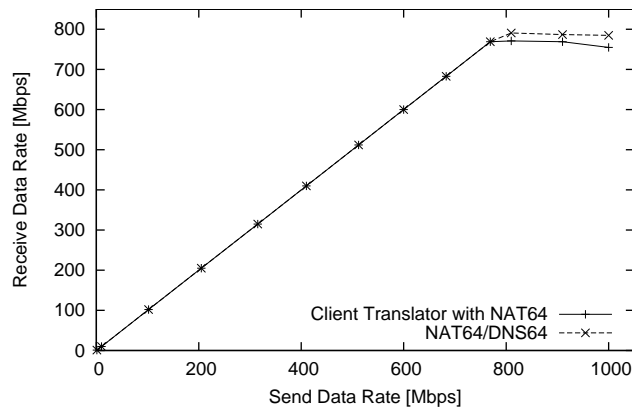


図2: UDPデータスループット

図2より送信サイズが800Mbpsを超えると性能の低下が見られたが、それ以下では同等の性能を発揮しており十分実用的であると考えられる。

## 6 まとめ

本研究では、クライアントノードにトランスレータを実装し、従来のNAT64/DNS64方式と評価を行った。クライアントノードのトランスレータを導入することによる性能低下はごくわずかであり、またDNS64を用いないことにより数多くのソフトウェア資源が使用出来ることからよりスムーズなIPv6移行が出来ることが期待出来る。

今後の課題として、今回の実験は影響を極力排除した環境にて行ったため、実際のクライアント環境で性能低下が発生しないかの検証を行う必要があると考えられる。

## 参考文献

- [1] 日本ネットワークインフォメーションセンター “IPv4アドレス在庫枯渇&基礎編” <http://www.nic.ad.jp/ja/ip/ipv4pool/qa1.html>
- [2] M. Bagnulo, P. Matthews “Stateful NAT64: Network Address and Protocol Translation from IPv6 Clients to IPv4 Servers” <http://tools.ietf.org/rfc/rfc6146.txt>, April 2011
- [3] M. Bagnulo, A. Sullivan, P. Matthews “DNS64: DNS Extensions for Network Address Translation from IPv6 Clients to IPv4 Servers” <http://tools.ietf.org/rfc/rfc6147.txt>, April 2011
- [4] Debian – The Universal Operating System, <http://www.debian.org/>
- [5] “Iperf – Free software downloads at SourceForge.net” <http://sourceforge.net/projects/iperf/>