

第3部

特集3 自家製ネットワーク技術を活用した研究促進活動

上野 幸杜、大江 将史、空閑 洋平、島 慶一、堀場 勝広、松平 直樹、松谷 健史

第1章 概要

最先端の研究開発を進める場合、研究に必要となる設備自体が存在しない場合も多い。新しいアイデアや実装方式は、それ自体の斬新さ故に既存の製品では提供されていないことが多いためである。そのため、WIDEプロジェクトでは、内容の如何に関わらず、ハードウェア技術を活用してネットワーク技術の研究活動に役立つ為の基礎をお互いに学習する場、またソフトウェアのみの技術では解決が困難な課題をハードウェア技術を用いて解決できるようなアイデアを考えていく場、また新しいソフトウェアアーキテクチャを自らの手で考えていく場として2010年10月にHandmadeワーキンググループを設立した。

2012年度は、新しいIPルータアーキテクチャの研究、超高速ネットワーク対応PCルータの開発、IPv6移行に備えるためのソフトウェアルータの開発を実施した。研究開発の成果は、2012年6月にInterop Tokyo2012の特別企画として開催されたオープンルータ・コンペティション^{*1}で披露されている。

オープンルータ・コンペティションは、高度化、高速化が進むインターネット技術の研究開発におけるイノベーションを加速することを目的として、奈良先端科学技術大学院大学の門林氏の呼びかけで開催された。次世代インターネットの研究開発を担う技術者の掘り起こしと研究開発内容の促進を目指している。

本章では、オープンルータ・コンペティションに応募、披

露された以下の4つのトピックに関する報告を行う。なお、次節で紹介するFIB offloading NICはコンペティションにおいて「グランプリさくらインターネット賞」を受賞している。

- FIB offloading NIC(第1章)
- 手作りネットワークテストMAGUKARA(第2章)
- 高速パケット処理システムULTRA40 (第3章)
- SA46T-AT Advanced IPv6 Transition(第4章)

第2章 FIB offloading NIC

2.1 概要

ネットワークの高速化に伴いPCルータの性能が問題になってきている。PCルータは機能面において専用ルータと比べ著しく劣るような事はないが、性能面においては専用ルータが専用ASICを用いているのに対しPCルータはソフトウェア処理を行うため劣っている。特に10Gbpsを超える高速インターフェイスや最小フレームサイズをワイヤレート近くで負荷を与えるような場合、その差が顕著にあらわれる。

FIB offloading NIC (FIBNIC) はPCルータがソフトウェアで処理するパケット転送機能の一部をFPGA (Field Programmable Gateway Array) にオフローディングさせる事により、PCルータの高機能性とFPGAを用いたハードウェア処理による高性能を実現するための技術である。ハードウェアでのパケット転送処理をオフローディングしていない時などはマルチポートNIC (Network Interface Card) として動作するため、既存のPCルータソフトウェア

*1 <http://www.interop.jp/2012/orc/index.html>

の修正をせずに対応させることが容易になっている。

2.2 PCルータの制約

性能面におけるPCルータの問題には、pps (packetper second)によるものとスループットによるものがあげられる。

ppsに関して、一般に普及されているGigabit Ethernetの場合でも、64バイトのショートパケットでは最大1,488,095ppsになる。これは2ポートルータで上りと下りをワイヤーレートで処理する必要がある場合、5,952,380pps (およそ6Mpps)のパケット処理が必要になる。すなわち、3GHzで動作するCPUにおいて500サイクル内で一連のパケット処理(割り込み、ドライバ、プロトコルスタック、パケット受信、ACL、NextHop算出、パケットの書き換え、SUM再計算、パケット送出)を終える必要があることを示し、キャッシュミスなどを容認できないほどの厳しい要求といえる。

スループットに関して、40Gigabit Ethernetの場合、一方方向転送において5GB/secのメモリ間転送が発生し、これはワイヤーレートで動作する2ポートルータにおいて20GB/secのメモリ間転送能力が要求されることを示す。一方、DDR3-1600のメモリの転送能力は最大12.8GB/sec^{*2}である。メモリの転送能力はマルチチャンネルメモリアーキテクチャ^{*3}技術により向上させることも可能だが、転送処理にはパケットの複製や書き換えなどOSやアプリケーションによるメモリアクセスも生じるため、メモリーバスの帯域がボトルネックになる。

2.3 FIBNIC

PCルータの処理にはルーティングプロトコルやユーザーインターフェイス、障害対策のための冗長性、暗号化、パケットの転送など様々なものがある。FIBNICはパケットの転送機能をハードウェアでオフローディングさせることにより転送性能を向上させる。FIBNICはPCホストから見るとマルチポートNICデバイスとして認識される。

FIBNIC上にはパケット転送に必要なFIB (Forwarding Information Base)の情報を持つ。カーネル上にあるFIBと同期することができるので、カーネル上のルーティングテーブルに基づいてハードウェア処理が可能となる。FIBに合致するパケットを受信した場合は、FIBNIC上で転送処理が行われ、ホスト宛通信などでFIBと合致しない場合は通常のNICへパケットをスルーさせソフトウェアによって処理を行う。

2.4 FIBNICの設計と実装

図2.1は、FIBNICのシステム構成を示す。

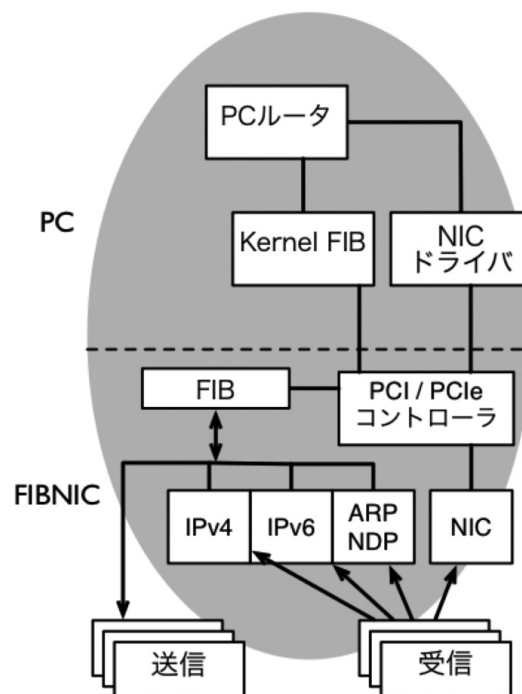


図2.1 システム構成

FIBNICはPC上のPCIeデバイスとして構成され、受信パケットのフレームタイプ、プロトコル、宛先によってNIC回路とARP/NDP回路、IPv4回路に振り分けられる。

NIC回路はFIFOとデュアルポートメモリより構成され、PCのNICドライバ経由で操作する。対象となるパケットはブロードキャストパケット、NIC宛アドレスおよび

*2 http://en.wikipedia.org/wiki/DDR3_SDRAM

*3 http://en.wikipedia.org/wiki/Dual-channel_architecture

IPv4/IPv6回路で対象から外れたものである。

ARP/NDP回路はIPv4のARPとICMP、IPv6のNDP処理を行う。このため、ホスト側のPCルータが停止しても、PCIeの電源さえ供給されていれば、ルーティングの動作を継続できるが、ルーティングプロトコルの処理が出来ないため、経路表の変更を反映することは出来ない。

IPv4回路は受信パケットの宛先IPアドレスよりNext Hopと出力ポートを調べ、宛先MACの書き換えと、TTLの減算、ACLへのマッチングを行う。これらおよそ50サイクル(@125MHz)のパイプライン処理を経た後、送信FIFOへ出力される。

2.4.1 実装環境

表2.1に本実装の開発環境を示す。

表2.1 開発環境

Dev kit	NetFPGA-1G
FPGA	Virtex-II Pro 50
Ethernet	1000BASE-T x4
Peripheral Bus	PCI
SDK	ISE 10.1SP3
Verilog sim	GPL Cver/GTKWave

FIBNICはマルチポートのイーサネットインターフェイスを搭載したFPGA開発ボードが必要である。本実装は、Xilinx社のXilinx Virtex-II Pro 50を搭載したNetFPGA-1G[11]をサポートしている。Xilinx社の開発ボードはEthernet 1000BASE-Tを4ポートとPCIスロットを搭載している。

本設計はメーカー問わず、汎用のFPGAボード上で動作することを目標としている。NetFPGAにはStanford Universityが設計、開発したHDL (Hardware Description Language) モジュールおよびソフトウェアのプラットフォームが提供されているが、それらは使用せずボード上の部品を使用する形で新規に設計した。

2.4.2 IP LookupとFIB

パケットの転送先を調べるIP Lookupはルーティングの中

でも時間のかかる処理で、極めて重要な役割を担う。

IP LookupはFIBにあるルーティングテーブルをもとに、Longest Match処理を行い転送先を探索する。

これらをハードウェアで実装する場合、ルータ専用機ではTCAMを用いる事が一般的であるが、TCAMは多量のハードウェアリソースを消費するため、ASICと比べて回路規模が小さいFPGAには適さない。

一方、IPアドレスをRAMのアドレスにマッピングする手法として、文献[12]があげられるが、今回実装するNetFPGAのSRAMは4MBしかないためそのまま適用することは出来ない。

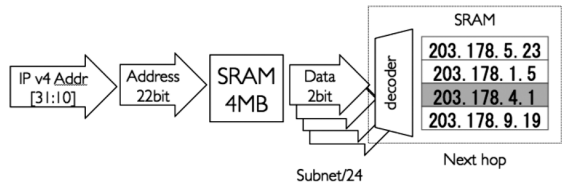


図2.1 IP Lookup

本論文では図2.1に示す方法で実装する。

4MBのSRAMのアドレスバスは22ビットになることから、IPv4アドレスの上位22ビット分までをSRAMのアドレスバスに割り当てる。次に8ビットのデータを2ビット4に分割し、サブネットマスクを2ビット拡張し、24ビットまでのサブネットマスクへ対応させる。1セグメントあたり4値となるので、3種類のNext Hop情報を持つことが出来る。この値をデコードする事により、Next HopのIPとポート番号を取得する。

SRAMはデュアルポートの同期式パイプラインなので、166MHz動作時で最大166M回/秒の参照が可能である。

今回の構成のような4ポートGigabit Ethernetルータでは、64バイトショートパケットのワイヤーレートでも約12Mppsの負荷に抑えられるため、それぞれのポートにIP Lookup用のキャッシュ機構を採用する必要はない。

2.5 評価

性能評価にはハードウェアによるトラフィック生成と遅延計測が可能なMAGUKARA[13]を用いた。

表2.2に64バイトフレームを用いた測定結果、表2.3に1518バイトフレームを用いた測定結果を示す。

表2.2 64バイトフレーム(IPv4)

対象	pps	遅延 (ns)	備考
NIC+PC ルータ	365,497	24,144	
FIBNIC+PC ルータ	1,488,095	976	IFG=12
Catalyst 3750G-24P	1,470,588	6,600	IFG=13

表2.3 1518バイトフレーム(IPv4)

対象	pps	遅延 (ns)	備考
NIC+PC ルータ	59,355	101,152	
FIBNIC+PC ルータ	81,274	976	IFG=12
Catalyst 3750G-24P	81,222	20,608	IFG=13

NIC+PCルータはIntel Core i5 2.8GHzとLinuxKernel 3.3.7を採用した構成である。FIBNIC+PCルータはIntel Celeron 2.53GHz、Linux Kernel 3.3.7、IPルーティングテーブルを41万件登録した構成である。

FIBNIC+PCルータではワイヤーレートに相当する1,488,095ppsが出ているが、Catalyst3750G-24Pではワイヤーレート時ではパケットロストが生じたため、IFG(Inter Frame Gap)を13 バイトにして計測し直したものである。ただし、MAGUKARAは送信ppsをIFG単位でしか調整できないため、実際には3750G-24Pの転送能力がこれより高い可能性がある。

いずれもFIBNIC+PCルータはワイヤースピード時のpps値でかつ、フレームサイズに関わらず遅延が一定である。これはパケットをRAMにバッファリングせずパイプライン処理で全行程を終えている事により、L3でのカットスルーが行われているためである。

次にフレームサイズ毎の遅延評価を図2.3に示す。GS908E、LSW4-GT-8EP、LSW3-GT-5EPは家庭向けL2スイッチであるが、参考値として計測した。

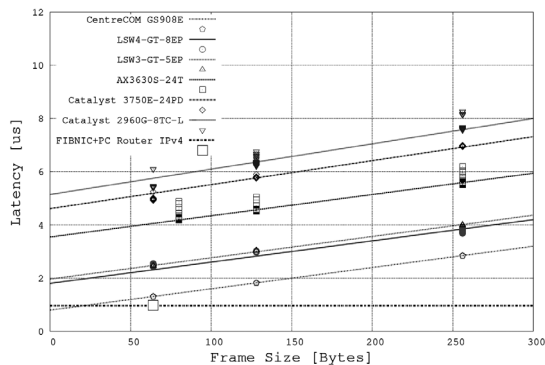


図2.3 フレームサイズ毎の遅延

2.6 今後について

FIBNICはPCルータ上でのフルルート、10Gワイヤースピードを目指し設計、実装したものである。

今回はリソースの問題で2007年に設計されたFPGAボード上で実装したため、インターフェイスの速度やメモリーの容量などに大きな制約があった。

今後の方針としては、10GFPGAへの対応と、入手がしやすい価格帯での8ポート以上のGigabit FPGAボードを設計し、オープンソースのFPGAによるL2/L3スイッチを広く普及させることを目指す。

第3章 手作りネットワークテストMAGUKARAの開発

3.1 概要

ルータ機能の開発に際しては、開発するハードウェアの動作と性能計測が可能な検証環境が必要になる。例えば、ワイヤーレートで動作するネットワークハードウェアの開発であれば、開発対象がワイヤーレートで動作しているか検証するツールが必要である。今回、オープンルータ・コンペティションに参加し、FPGAを使ってソフトウェアルータの転送機能をハードウェアにオフロードするNIC(1節参照)を制作したが、その性能計測は大きな課題だった。一般的な性能検証環境は比較的高額であり、今回のような事例に応用することは困難である。そのため、今回のルータ開発は計測機器の作成からはじまっている。

開発した計測機器には、作成当時約9,000円で購入可能(現在は3万円程度)だったPCI Expressインタフェースを持つFPGA開発キットが用いられている。ハードウェアのFPGA回路とソフトウェアのGUIフロントエンド両方共にオープンソースで公開しているため、同じFPGAボードを購入すれば誰でも本ネットワークテストを利用可能である[13]。

3.2 ネットワークテスト環境と手作りハードウェア開発

ネットワーク機器の性能検証には、一般的にネットワークテストが利用される。ネットワークテストは、汎用PCとテストソフトウェアを用いた方法と、専用計測機を利用した方法がある。ソフトウェアによるテスト環境は、ソフトウェアを改変することによってテストシナリオに応じた、柔軟な検証が可能である。しかし、現在バックボーンで利用されているような10Gigabit Ethernetや40Gigabit Ethernetインタフェースを持つネットワーク機器の高精度な検証が困難である。汎用PC環境による10Gigabit Ethernet環境の整備は未だ発展途上にであり、ショートパケットなどのPPS処理性能の限界がある。また、OSのタイマ精度には限界があるため、1Gigabit Ethernetであってもショートパケットやワイヤーレートによる検査には制約がある。

そこで、ネットワークハードウェアの検証には、テスト専用機が利用されている。これらは、パケットのジェネレートと解析機能をハードウェアで処理することによって、ショートパケットによる検査や高精度な検査を実現している。しかし、専用機は検証ソフトウェアと検証機のハードウェアが一体となって提供されるため、柔軟な検査シナリオの改変が困難である。近年のSDNの登場やデータセンターにおけるInfinibandの普及など、ネットワーク機器のインタフェースは多様化しており、これらに柔軟に対応するためには、ハードウェアの性能とソフトウェアの柔軟性の両方が必要になる。

一方、近年の半導体プロセスの微細化によって、FPGAに集積されるトランジスタ数は劇的に増加しており、ソフトウェアによってプログラム可能なFPGAは、様々なネットワーク機器で採用されている。また、PCI Expressのような高速インターフェース経由でFPGAを接続することによって、CPUとFPGAによるヘテロジニアスな計算機環境が実

現可能になっている。

しかし、現在のFPGAは企業によるASIC開発のプロトタイプや製品開発が主な利用場面である。世界に多くのソフトウェア開発者が存在することと比較すると、一般ユーザーによるFPGAを用いた開発は稀である。CPUとFPGAを利用したアプリケーション開発には、ソフトウェア側のハードウェアとのインタフェース、コントロールプログラム、GUI、ハードウェア側のインタフェース、各種回路といった両方の開発が必要であり、多くの知識と経験が必要になることがその理由の一つであると考えられる。

そこで今回、ネットワークテストを作成するにあたり、WebSocketとブラウザを利用することでソフトウェア側の開発コスト低下を目指した。一般的に、ハードウェア回路設計の検証ツールは高価だが、オープンソースによる豊富なJavaScriptの可視化ライブラリを利用することで、テスト作成の手間を削減できる。本実装では、ブラウザ側のHTMLのフォームから直接ハードウェアレジスタを操作可能にすることで、ネットワークテストの検証シナリオをブラウザから操作可能としている。これにより、作成したいネットワークデバイスに合わせたプログラマブルなネットワーク検証環境を実現しており、今後登場する新しい機能にも柔軟に対応可能である。

3.3 試験内容とサポートFPGA環境

図3.1に本テストの計測環境を示す。

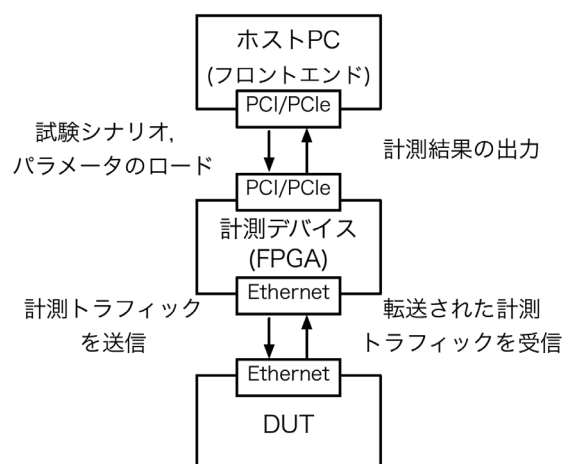


図3.1 計測環境

一般的なネットワーク機器のテスト方法は、RFC1242、RFC2544に記載されている[14, 15]。また、IPv6対応機器のテストについては、RFC5180[16]に記載されている。特に、RFC2544は、基本的なネットワーク機器のテスト方法が記載されている。

検証項目は、スループット、レイテンシ、フレームロス率、連続フレーム数、システムリカバリ、リセットシグナルからのリカバリなどがある。リセットシグナルからのリカバリテストについては、RFC6201で更新されている[17]。本論文では、RFC2544の用語定義より、試験対象のネットワーク機器をDUT(Device Under Test)と呼ぶ。

3.4 実装環境

表3.1に本実装の開発環境を示す。

表3.1 開発環境

Dev kit	ECP3 versa kit	NetFPGA-1G
FPGA	LatticeECP3	Virtex-II Pro 50
Ethernet	1000BASE-T x2	1000BASE-T x4
PCI/PCIe	PCI Express 1.1	PCI
SDK	Diamond 1.4SP2	ISE 10.1SP3
Verilog sim	GPL Cver/GTKWave	

ネットワークテストは、試験トラフィック生成用のポートと受信ポートの2ポート以上を使用するため、マルチポートを搭載したFPGA開発ボードが必要である。本実装は、Lattice社のLattice ECP3 Versa Development Kit^{*4}とXilinx社のXilinx Virtex-II Pro50を搭載したNetFPGA-1G[11]をサポートしている。Lattice社の開発ボードは、Ethernet 1000BASE-Tを2ポートとPCI Express 1.1を搭載しており、Xilinx社の開発ボードはEthernet 1000BASE-Tを4ポートとPCIスロットを搭載している。本実装では、将来的な10G/40G Ethernetのサポートのために、マルチベンダのFPGAチップをサポートできる構成になっている。

3.5 試験パケット構成

図3.2と図3.3を使って、本テストで使用する試験パケットフォーマットを説明する。

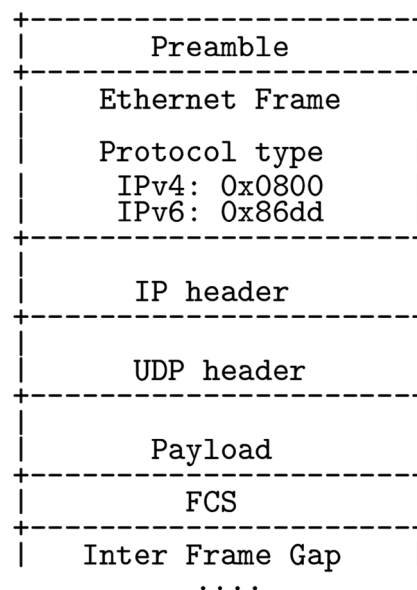


図3.2 計測用パケットフォーマット

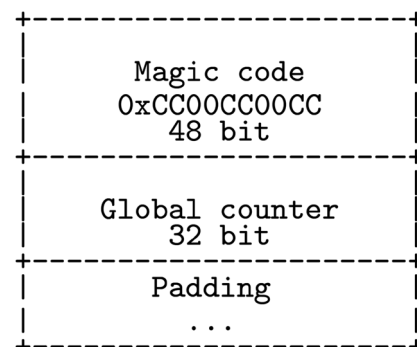


図3.3 計測用パケットのペイロードフォーマット

試験パケットはIPv4、IPv6に対応していて、UI側からIFG (Inter Frame Gap)、パケットサイズが変更可能になっている。IFGを変更することで試験パケット数を調整できる。また試験パケットのペイロードのPaddingサイズを変更することによって、パケットサイズを調整できる。

試験パケットのペイロードの先頭48bitは固定のマジックコードになっており、受信ポートでの試験パケットの判定に使用する。32bitのGlobal counterは、送信ポートのタイムスタンプとして利用しており、受信ポートで試験パケッ

^{*4} <http://www.latticesemi.com/products/developmenthardware/developmentkits/ecp3versadevelopmentkit/index.cfm>

トを受信した際にペイロードのタイムスタンプとの差を見ることによって、伝送遅延を測定している。

3.6 実装

図3.4 に本実装のシステム構成を示す。

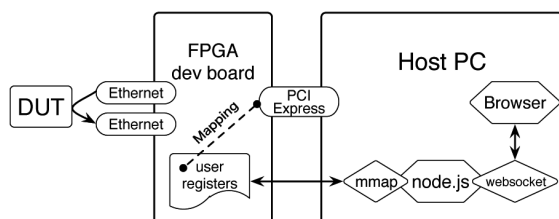


図3.4 システム構成

3.6.1 ハードウェア構成

FPGAボードの送信用ポートから送信した試験パケットは、DUTを経由して、受信ポートに返信される。その際計測したスループット、遅延、PPS (Packets per second)をFPGA内のレジスタで保持する。計測結果を保持したレジスタは、PCI/PCIe回路側から読み書きできるように配置され、mmap()システムコールなどを用いてOS側からPCIレジスタを直接操作できる。

3.6.2 ソフトウェア構成

フロントエンドは、サーバサイドJavascript実行環境であるnode.jsを用いて作成した。node.jsのnodemmapモジュールを利用することで、node.jsを使用してPCI/PCIeのメモリ空間を操作する。次に、node.jsのWebSocketモジュールであるwcを利用することでメモリマップした空間をブラウザに送信する。これにより、ブラウザからPCI/PCIeを経由してFPGA内のレジスタが操作可能になる。

3.7 今後の機能拡張

今後は、ホストPCからのpcapフォーマットによるシナリオ読み込みをサポートすることによって、より容易なネットワークテスト環境の実現を目指す。具体的には、FPGAボードに搭載されているDRAM上に、フロントエンドからpcapファイルを読み込むことによって、利用者の好きなトラフィック生成をサポートする。それにより、HTTPやTCPといった擬似トラフィックを使って、DUTの性能計測が可能になる。

第4章 高速パケット処理システムULTRA40の研究開発

国立天文台と情報通信研究機構は、2013年度から水沢観測所において、国立天文台CFCAプロジェクトが運用するHPCシステム、VLBIシステム等と情報通信研究機構が運用する10ギガビット毎秒の通信回線(2012年度に誘致)を組み合わせ、新世代ネットワークの共同研究を開始する。

この研究では、HPCシステムにおける計算ノードから出力されるシミュレーションデータや、VLBI観測データなどのデータストリームの属性に応じて、広域ネットワークを制御し、遠隔地へ効率良く伝送できる仕組みの実現に取り組んでいる。属性とは、各データストリームが有する伝送要件である。たとえば、シミュレーションデータの場合、量は膨大だが決められた時間までにエラーフリーで伝送できればよい。また、VLBIの観測データの場合は、パケットロスに肝要だが、時間は厳守しなければならない点等がある。具体的には、各ストリーム特性のデマンドに基づき、広域網上やり取りされるトラフィックのスケジューリング・最適な伝送路の策定・伝送方式を折衝し、伝送するしくみを実運用のストリームを元に研究開発を行っている。

その一環として、本活動では、ストリーム処理やストリームのキャッシング処理をおこなうシステムとして、IA (Intel Architecture)を利用した100ギガビットの処理能力を有するPCルータ(図4.1)と4ギガバイト毎秒の読み書き性能有するストレージシステムを開発した(オープンルータ・コンペティション、2012年インタロップに出展)。

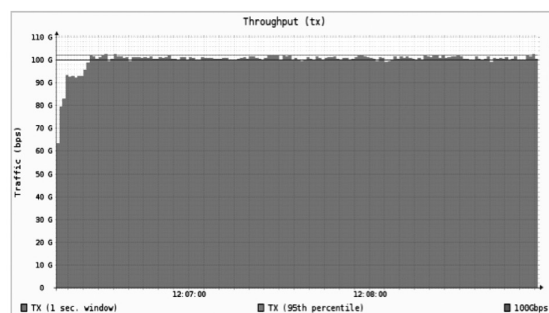


図4.1 100Gbps処理例(IXIAとSpirent計測器によるUDPトラフィック計測結果)

このシステムは、コストパフォーマンスに優れたIA環境に加えて、新世代の10ギガビットイーサネット対応のネットワークインターフェイス、超高速SSD、そして、これらを構成するハードウェアとソフトウェアにチューニングを施すことにより、割り込みタイプのパケット処理として、約100ギガビット、ストレージとして、4ギガビット毎秒の書き込み・読み込みの性能を達成した。現在、国立天文台では、より高い性能向上を目指し、現在、Intel社のDPDK(Data Plane Packet Processing on Embedded Intel®)および、ザイリンクス社やアルテラ社の40ギガ対応FPGAシステムによる応用システムの研究開発を進めている。

第5章 SA46T-AT Advanced IPv6 Transition

5.1 IPv4アドレス枯渇とSA46T

IPv4アドレス在庫枯渇が現実となり、通信事業者はIPv4の延命とIPv6へのシームレスな移行を平行して行っている。IPv4の延命としてはアドレス移転などを用いてIPv4アドレスの確保、LSN (Large Scale NAT)などIPv4アドレスの利用効率の向上、バックボーンネットワークをIPv6化しスタブネットワークのみ既存のIPv4を維持する方式など様々な手法がある。本研究はこれらの移行方針の中でも、IPv6化されたバックボーン上に既存のIPv4ネットワークを運用する方式に着目する。

SA46T[18]はIPv6化されたバックボーン上に、IPv4ネットワークをスタブとして展開する技術である。SA46Tの特徴としては、各スタブIPv4ネットワークをPlane IDと呼ばれる識別子によって認識し、運用しているバックボーンのIPv6 Prefixと組み合わせる事で、同じIPv4アドレスを用いたスタブネットワークも識別可能なレイヤ3VPN機能を実現する。すなわち複数の同一IPv4プライベートアドレスを利用するユーザが同居可能なIP-VPN網が提供できる。SA46Tは大規模なMPLS等と異なり、アドレスの割当設計と一般的なIGP経路制御によってこれらの機能が実装される、いわば運用とプロトコルの組み合わせによってステートレスなパケットエンキャプスレーション技術である。SA46Tは一般的なアドレス(プロトコル)変換技術を自由に選択可能なため、シンプルかつ他の技術で発生する利用可能なプロトコル制限等の懸念が小さい。

5.2 SA46T-AT

IPv6が普及しない理由の一つに、IPv4にのみ対応した機器を持つユーザに対してIPv6への移行が何らインセンティブを生んでいない点、IPv4にのみ対応した機器の存在が挙げられて来た。本研究はこれらの問題に対して、SA46Tの特徴を生かしてIPv6バックボーン上に平行して存在するプライベートIPv4アドレスを用いたスタブネットワークをIPv6アドレスによって個体認識しグローバルに到達可能にし、その上でプロトコル変換によってIPv4アドレスのみに対応した機器とIPv6アドレスを持つ機器の通信を可能にする。すなわち、SA46TルータにIPv6からIPv4へのトランスレータを組み合わせる事で、グローバルIPv6アドレスを持つ機器が、プライベートIPv4アドレスのみを持つ機器に対する接続性を提供する。本研究ではこの機能を、SA46T-AT(Address Translator)[19]と呼ぶ。

図5.1にSA46TとSA46T-ATを用いたIPv6バックボーンにおけるIPv4ネットワークの存続と、IPv6を用いたインターネットからIPv4へアクセスする環境の概要を示す。

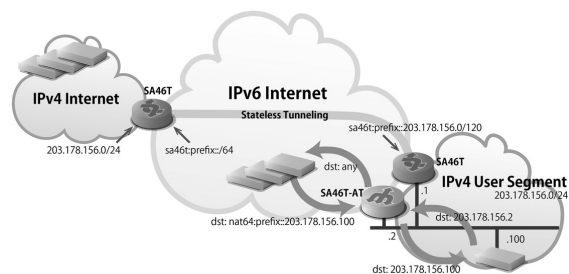


図5.1 SA46TとSA46T-ATによるIPv4ネットワーク存続シナリオ

SA46TルータはIPv4 User Segment(図中203.178.156.0/24)のPrefixとPlane-IDから生成したSA46T IPv6 Prefix (図中sa46t:prefix::203.178.156.0/120)を経路広告する。IPv4インターネットと接続されたSA46TルータはIPv4ネットワーク宛のパケットをSA46T IPv6 Prefix宛としてエンキャプスレーションし、IPv4 User Segmentへ転送する。IPv4 User SegmentのSA46Tルータはエンキャプスレーションされたパケットを取り出し、通常のIPv4パケットとしてIPv4 UserSegmentに転送する。

SA46T-ATルータはSA46Tと同様のルールで生成したSA46T IPv6 Prefixを(図中nat64.prefix:203.178.156.0/120)を経路広告する。IPv6インターネットからSA46T IPv6 Prefixに通信先IPv4アドレスをSuffixとしたIPv6アドレス宛に送信されたパケットは、SA46T-ATルータまで通常のIPv6経路制御によって配送され、IPv6アドレスからIPv4アドレス部分を抜き出した上でプロトコル変換を行い、IPv4 User SegmentのIPv4ホストへパケットを転送する。

SA46T-ATはIPv4のみに対応した旧型のシステムとIPv6にのみ対応した新型のシステムを融合可能な通信基盤を提供できる。例えば、後付けでIPv6対応困難なハードウェアチップ上に通信スタックが実装された旧型のセンサーと、新たにIPv4アドレスを取得するのが困難になっていくためIPv6のみに対応した新設のセンサーならびにセンサー情報収集アプリケーションを意識する事無く融合するためには、IPv6を起点とした通信をIPv4機器に対して通信可能なネットワーク基盤は利便性が高い。

5.3 SA46T-ATルータの設計と実装

SA46T-ATルータはIPv6側のネットワークインターフェースからのパケット入力をトリガーにIPv6とIPv4のプロトコル変換テーブルを作成する。設計上で留意した点として、変換テーブルのエントリをIPアドレスによる1対1変換を採用した。一般的なNAT技術と同様に、レイヤ4の情報まで含めた変換テーブルを構築する場合も検討したが、最終的には動作するアプリケーションにできる限り制限を与えない方法として、低レイヤでの変換を採用した。図5.2にSA46T-ATルータの動作概要を示す。

5.3.1 IPv6からIPv4向きの処理

1. 変換テーブルにIPv6ホストのエントリがあるか調べる。

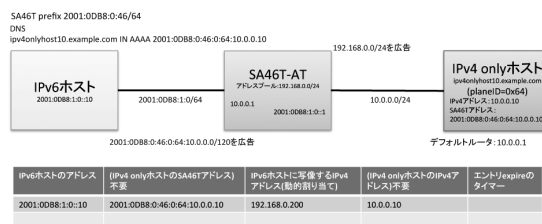


図5.2 SA46T-ATの設計仕様

2. 無ければ、作成して、写像するIPv4アドレスを動的に割り当てる。
3. 変換テーブルを発信IPv6アドレスをキーにして検索し、写像するIPv4アドレスを求める。宛先はSA46TアドレスのIPv4部を用いる。

5.3.2 IPv4からIPv6向きの処理

1. 変換テーブルを宛先IPv4アドレスをキーにして検索し、該当するIPv6ホストのアドレスを求める。発信アドレスはSA46Tアドレスの変換ルールで求める。

5.3.3 その他処理

1. タイマー起動でエントリを消去^{*5}する。
2. IPv6ホストに写像するIPv4アドレスのProxy ARP: 異なるサブネットのアドレス空間をプールにすれば不要。Proxy ARP面倒だから異サブネットアドレスをプールするように条件を設ける。

5.3.4 設定

1. 広告する経路(2001:0DB8:0:46:0:64:10:0:0/120)、なお、ここからplane IDがわかる。
2. 写像するIPv4アドレスプール。
3. エントリ消去タイマー時間(デフォルトは180秒)

5.3.5 SA46T-ATの実装

図5.3にSA46T-ATのLinuxにおける実装概要図を示す。

SA46T-ATの実装はLinux上でデーモンとして動作し、パケットの送受信はtun/tapを用いる。SA46T-ATデーモンは

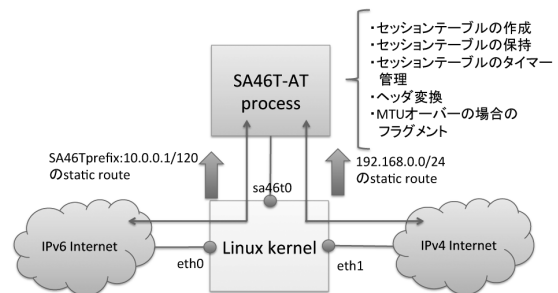


図5.3 SA46T-ATの実装概要ダイアグラム

*5 パケット変換が行われる際にタイマーが再設定される。

起動されると同時にtun/tapを用いて仮想インターフェースを作成し、仮想インターフェースに着信したパケットに対してヘッダ変換処理を行う。ユーザはSA46T-ATデモンの起動後に仮想インターフェースに対してSA46T PrefixとIPv4写像アドレスプールのスタティックルートを設定する必要がある。

IPv6パケットをIPv4パケットに変換する場合、SA46T-ATはIPv6パケットのソースアドレスをキーにして自身が持つセッションテーブルを検索する。セッションが存在した場合はそこに記録されたIPv4写像アドレスを変換後のパケットのソースアドレスとする。また、IPv6パケットの宛先アドレスの末尾32ビットを変換後のパケットの宛先アドレスとする。IPv6パケットのソースアドレスに対応するセッションが存在しない場合は、設定されたIPv4写像アドレスのプールから新たなIPv4写像アドレスを割当て、セッションを作成した後にIPv4パケットを生成する。セッションは180秒間通信がない場合にのみ消去される。

IPv4パケットをIPv6パケットに変換する場合、SA46T-ATはIPv4パケットの宛先アドレスをIPv4写像アドレスとしてセッションテーブルを検索する。セッションが存在した場合は記録されたIPv6アドレスをIPv6パケットの宛先アドレスとする。また、IPv4パケットのソースアドレスから生成したSA46T Global IPv6 AddressをIPv6パケットのソースアドレスとする。

なお、IPv4パケットをIPv6パケットに変換する場合はネットワーク層のヘッダ長が大きくなるため、変換先のリンクのMTUを超える場合が考えられる。SA46T-ATは、変換後のパケット長が設定されたMTUを超える場合、パケットをフラグメントした上で送信する。相互の変換に共通の動作として、TCPまたはUDPパケットの変換を行う場合はトランスポート層のチェックサムの再計算を行う。これは、ネットワーク層のヘッダがトランスポート層のチェックサムの計算元に含まれているためである。また、ICMPとICMPv6の間に相互互換性がないため、ICMPを変換する場合はICMP codeなどを個別に変換する。

5.4 評価

5.4.1 他技術との比較

SA46T-ATと他の技術との比較について、表5.1に示す。動作レイヤについては、SA46T-ATはネットワーク層のみで動作するが、NAT64及びTRTについてはトランスポート層の変換についても定義されている。特に、TRTはTCPの再送制御等についてもトランスレータが行うため、通信効率は高くなるが、変換の際の負担が大きくなるという特徴がある。また、IPv6空間からIPv4空間へのアクセスが可能である点、IPv4空間からIPv6空間へのアクセスが不可能である点は共通である。なお、この場合のアクセスはセッションが存在しない時点での通信の可否を指す。IPv4空間の多重化については、SA46T-ATではプロトコルスペックで定義されている。NAT64及びTRTは仕組み上可能ではあるものの、プロトコルスペックでは定義されていない。SIITはアドレッシングの際にホスト部の末尾32ビット以外を予約するため、多重化は不可能である。想定用途としては、SA46T-ATのみがIPv4のみのエッジネットワークに設置することを想定し、他の技術はIPv6バックボーンとIPv4バックボーンの境界に設置することを想定している。

表5.1 SA46T-ATと他のプロトコル変換技術の比較

	動作レイヤ	v4⇒v6	v6⇒v4	IPv4 空間の多重化	想定用途
SA46T-AT	ネットワーク層	不可	可	可	IPv4 only host 側
NAT64	ネットワーク層、トランスポート層	不可	可	可	IPv6 Internet 側
TRT(Transport Relay Transition)	トランスポート層	不可	可	可	IPv6 Internet 側
SIIT	ネットワーク層	不可	可	不可	IPv6 Internet 側

5.4.2 パフォーマンスの計測

SA46T-ATを設置した場合、トランスレーション対象へのIPv6のトラフィックはすべてSA46T-ATを経由するため、最大スループットは重要なメトリックである。実際にどの程度のトラフィックを変換することが可能か検証し、パフォーマンスの傾向を明らかにするため、以下の対象についてIperf^{*6}を用いてスループットの計測を行った。

1. 直接接続(Direct)
2. Linuxカーネルが提供するIPv6ルーティングエンジン

*6 <http://iperf.sourceforge.net/>

(IPv6 routing)

3. NAT64実装の一つであるTAYGA(TAYGA)

4. 本研究で実装したSA46T-AT(SA46T-AT)

スループットを計測する環境を構築するために、実装したSA46T-ATを動作させるためのサーバを1台、Iperfのサーバ及びクライアントとして動作させるためのサーバを2台、計3台のサーバを用いた。この3台のサーバのスペックを以下に示す。

● トランスレータ

OS: Ubuntu 10.04 x86 64

CPU: Intel Xeon 5160@3.00Ghz 2 cores/2 threads

Memory: 4GB

NIC: Intel 80003ES2LAN x 2

● Iperfサーバ

OS: Ubuntu 10.04 x86 64

CPU: Intel Xeon 5160@3.00Ghz 2 cores/2 threads

Memory: 4GB

NIC: Intel 80003ES2LAN

● Iperfクライアント

OS: Ubuntu 10.04 x86 64

CPU: Intel Core i7 975@3.33Ghz 4 cores/8 threads

Memory: 12GB

NIC: Realtek RTL8111/8168B

この条件において、Direct、IPv6 routing、TAYGA、SA46T-ATのそれぞれについて、TCP及びUDPでパケットサイズを1500byteと250byteとした場合の計16通りを3回ずつ10秒間計測し、各場合において3回の計測の平均値を計測結果とした。本計測のBPSでの結果を図5.4に、PPSでの結果を図5.5に示す。

パケットサイズが1500byteの場合、BPSではSA46T-AT、TAYGA (以下トランスレータ)とDirect、IPv6 routing (以下非トランスレータ)の間に目立ったパフォーマンスの差は現れなかった。パケットサイズが1500byteでTCPの場合にトランスレータのPPSが大きくなるという現象が発生した。この現象の原因については判明していないが、TCP特

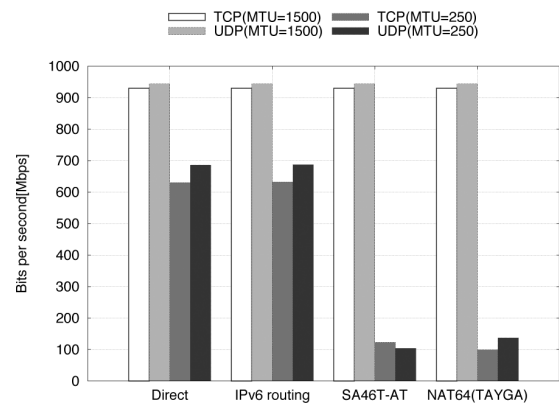


図5.4 BPSでの計測結果

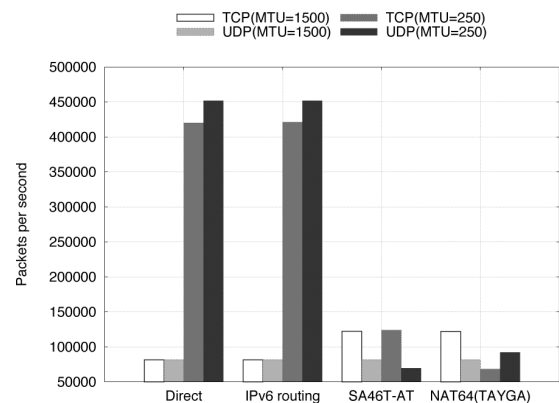


図5.5 PPSでの計測結果

有の現象でありBPSに変化がないことから、何らかの原因によりパケットの再送がおきていると推測できる。パケットサイズが250byteの場合は、BPS及びPPSの計測結果から、トランスレータのパフォーマンスが低下する傾向があることが判明した。以上から、トランスレータは時間あたりのパケット処理可能数が非トランスレータと比べて少ないが、NAT64とSA46T-ATの間に目立ったパフォーマンスの差はほぼ存在しないと言える。

第6章 まとめ

本研究ではSA46Tのアドレス設計ならびに経路広告を利用するアイデアに、IPv6からの通信をトリガーとしたIPv6-IPv4プロトコル変換機能を組み合わせたSA46T-ATを提案しLinux上に実装した。SA46T-ATによって、プ

ライバートIPv4ノードに対してIPv6からの接続性が提供できる。SA46T-ATは本章著者の松平によってIETFにInternet-Draftとして提案されている[19]。また、Linux上
に実装しInterop Tokyo 2012において併設されたオープン
ルータ・コンペティションに出展した。

6.1 謝辞

本研究活動は富士通株式会社とWIDEプロジェクトの共同研究の一環である。本研究に発表の場を提供していただいた、オープンルータ・コンペティション開催に尽力された方々に感謝する。