

IPv4-to-v6 トランスレータにおけるパケット分割処理

2003MT037 加藤 慎也

指導教員 後藤 邦夫

1 はじめに

2011年4月15日、日本を含むアジア太平洋地域のIPv4アドレス在庫が枯渇した。各事業者は早い段階でIPv6への対応を進めていくことが求められている。しかしIPv4とIPv6には互換性がないため、IPv4のみのネットワークがIPv6の移行が完了したネットワークから取り残されてしまうことが予想される。そこでIPv4-to-v6トランスレータ(以下トランスレータ)が必要になる。過去の卒論(発行2010年)[1]では、IPv4からDomain Name System(以下DNS)に聞かれるFully Qualified Domain Name(以下FQDN)に対し、AAAAレコードをAレコードに変換して返答を返せること、またヘッダ変換やチェックサム計算、上位層ヘッダを変換することで、IPv4hostとIPv6hostの通信が可能であることを示した。

本研究は[1]の引き継ぎであり、未実装であったIPv4からIPv6に変換する際のパケット分割処理を追加する。実装はC++を用い、GOTO's IP Network Emulator(以下GINE)クラスを使用して、既存プログラムに追加する。GINEを使用し、実際に分割処理が出来ているか確認する。IPv6のMinimum Maximum Transmission Unit(以下MTU)である1280bytesを標準とするが、現実にはMTU1500で実験する。MTUはIPヘッダとペイロードの合計の最大値である。

2 上位層ヘッダ変換と分割処理の必要性

IPv4, IPv6それぞれから始まる通信と、分割されていないパケット、1500bytes以上のもともと分割されて発信されたパケットの扱いと、プロトコルのTCP, UDP, ICMP, ICMPv6の12通りの上位層ヘッダ変換(過去の卒論[1]と同じ)と分割処理の必要性を考えた。その様子を表1で示す。UDPやTCPはIPv4とIPv6でその違いはない。しかしICMPとICMPv6はIPv4とIPv6で構造は似ているが、割り当てられたタイプとコードがかなり異なる。そのため、ICMPとICMPv6のヘッダ変換をする必要がある。TCPのMaximum Segment Size(以下MSS)は受信可能なセグメントの最大サイズを通信相手に通知するためのオプションである。TCPはMSSオプションを使うので、トランスレータで分割が必要なのは、UDPとICMPである。チェックサム再計算は全ての場合に必要な。IPv6フラグメントパケットの2番目以降のフラグメントパケットはTCP, UDP, ICMPヘッダがパケット内にない。

表1: IPv4からIPv6への上位層ヘッダ変換と分割処理の必要性(分割されていない場合)

	変換	分割
TCP	必要なし	必要なし
UDP	必要なし	必要
ICMP	必要	必要

3 プログラムの実現

本節ではIPv4とIPv6のパケットを、相互に変換する変換ゲートウェイ(TG)プログラムについて説明する。まずフレームを1つか、2つdequeueする。メソッドIP4to6()では、フレーム情報の書き換え前のフレームfと、2分割なら送信用のフレームout1, out2を用意し、書き換えた結果をそれらに入れていくようにした。/usr/include/netinetの中のip.h, icmp6.h, ip6.hなどの構造体にキャストして、ヘッダを数だけ用意する。各ヘッダフィールドにフラグメントヘッダなどの情報を代入してデータも書き換え、2つのフレームをFrameクラスのnextポインタで繋いで、out2をreturnする。そしてout2->next, out2の順番でenqueueする。メソッドIP6to4()ではIPv6hostで再構成されて、また分割したフレームを1つずつ通過させIPv6ヘッダからIPv4ヘッダに変換し、フレームをenqueueする。

4 フラグメンテーション

本節では、フラグメンテーションについて述べる。EthernetではMTU1500である。1500bytesから、IPv6ヘッダの40bytesと上位層ヘッダの8bytesを引いた、1452bytesを超えると2つにフラグメントする必要がある。1500bytesを越えるとIPv4発で、もともと分割されて発信されて3つにフラグメントする必要がある。そしてIPv6hostで再構成されて、また2つにフラグメントされトランスレータを通過し、IPv4hostで再構成される。その様子を図1で表す。図中の番号については、5.3節で説明する。

UDPはDon't Fragment(以下DF)が多い。DFビットが1で、かつ長いIPv4パケットも、トランスレータでフラグメントすれば通過する。IPv6からIPv4へパケットを変換する場合は、データをさらにフラグメントする必要がない。ヘッダサイズが小さくなるため必ず一つのパケットに収まるからである[2]。

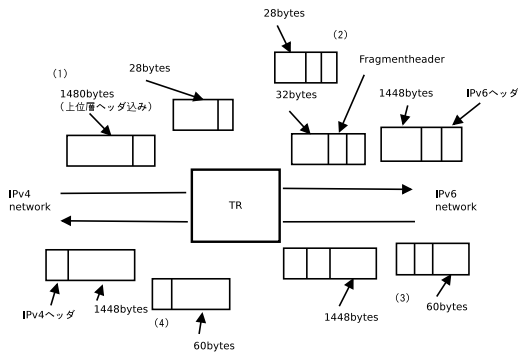


図 1: パケットサイズ 1500bytes のフラグメントの様子

5 トランスレータの実験

本節ではトランスレータの実験について述べる。

5.1 実験の構成

Network NameSpace(以下 NS) を 2 つ作成し、実機を 2 つ表現する。ひとつは IPv4 のみの host(172.16.0.2), もう一つは IPv6 の host(2002:0:2::2) である。この 2 つある NS の間にトランスレータを置く。ICMP は ping コマンドを使って実験を行い、UDP は自分で書いたプログラムで実験を行う。そしてそれぞれ tcpdump でパケットを dump してその様子を確認する。

5.2 実験環境

本研究の実験で使用した PC のスペックを表 2 に示す。

表 2: 使用した PC のスペック

PC	Let's note CF-W7
OS	Ubuntu 10.04 Jaunty Jacklop 32 bit
CPU	Intel(R) Core(TM)2 Duo CPU U7600 @ 1.20GHz
メモリ	2.0GB

5.3 実験結果

フラグメント成功例の payload の bytes 数を表 3 に示す。(1) から (4) までの番号は、図 1 の番号と対応している。ICMP、ICMPv6 の動作確認をするために、IPv4 から IPv6 の向きにパケットサイズ 1500bytes で ping コマンドを実行した様子を以下に示す。tcpdump の結果の番号は図 1 と表 3 の番号と対応している。Next header は Fragment header の 44 で、payload length は Fragment header の 8bytes を足した値になっている。アドレス (2001:0:1:ff::ac10:2) は、IPv4host から IPv6host に ping コマンドを実行して、アドレス変換ルールが決まった IPv6 アドレスである。そこから IPv6host(2002:0:2::2) に通信しているのが、トランスレータで 3 分割した echo request で、逆に通信しているのが IPv6 で再構成して、また 2 分割した echo reply

である。

表 3: フラグメント成功例 (payload)

UDP と ICMP	MTU ぎりぎり で発信 (bytes)	もともと分割 されて発信 (bytes)
(1)IPv4 発	1468	1480+28
(2) 分割後	1448+20	1448+32+28
(3) 応答	1448+20	1448+60
(4)IPv4 受	1448+20	1448+60

```
ping 10.248.66.126 -M dont -s 1500
```

```
(2)IP6 (hlim 62, next-header Fragment (44)
payload length: 1456)
2001:0:1:ff::ac10:2 > 2002:0:2::2:
frag (0x000073c4:0|1448) ICMP6,
echo request, length 1448, seq 2
(2)IP6 (hlim 62, next-header Fragment (44)
payload length: 40)
2001:0:1:ff::ac10:2 > 2002:0:2::2:
frag (0x000073c4:1448|32)
(2)IP6 (hlim 62, next-header Fragment (44)
payload length: 36)
2001:0:1:ff::ac10:2 > 2002:0:2::2:
frag (0x000073c4:1480|28)
(3)IP6 (hlim 64, next-header Fragment (44)
payload length: 1456)
2002:0:2::2 > 2001:0:1:ff::ac10:2:
frag (0x00000002:0|1448) ICMP6,
echo reply, length 1448, seq 2
(3)IP6 (hlim 64, next-header Fragment (44)
payload length: 68)
2002:0:2::2 > 2001:0:1:ff::ac10:2:
frag (0x00000002:1448|60)
```

6 おわりに

本研究では UDP と ICMP の、トランスレータで IPv4 発 1460bytes から 2924bytes までは、2 または 3 分割して IPv6 に変換できて、IPv6host からの応答が IPv4host で受信できた。2925bytes 以上の 4 分割は、使用した Linux kernel 2.6.32-22-generic では送信できないが、UDP では長いデータを通常送信しないので問題ない。

参考文献

- [1] 森 千恵, 畑佐宏輝: IPv4/IPv6 トランスレータの実現とネットワークエミュレータ上での評価, 卒業論文, 南山大学 数理情報学部 情報通信学科 (2010).
- [2] 宮田宏: 第 2 回 トランスレータの動作概念 (パケットの変換・前編) (accessed Nov. 2010). <http://itpro.nikkeibp.co.jp/article /COL-UMN/20091016/338946/>.