特別研究報告

題目

マルチホーム環境実現のための IPv6 プレフィクス変換機構の開発

指導教官

尾家 祐二 教授

報告者

鬼丸 敬輔

平成 15 年 2 月 25 日

九州工業大学 情報工学部 電子情報工学科

平成 14 年度 特別研究報告

マルチホーム環境実現のための IPv6 プレフィクス変換機構の開発

鬼丸 敬輔

内容梗概

近年インターネットは急速に拡大し，利用者数も増加している．それにともなってインターネットの重要性は高まっており，電気や電話などと同様な社会基盤として認知されるようになった．しかし現状では，インターネットで利用する通信システムは電気や電話と比較すると不安定であり，今後さらに可用性（availability）を向上させることが望まれている．

一方，近年 ADSL や FTTH といった安価なアクセス回線が普及しはじめており，容易に一つの組織が複数のアクセス回線を保有可能となってきている．そこで，これらの安価な回線を複数同時に用いることで可用性を向上させるための「マルチホーミング（multi-homing）技術」が注目されている．

しかし現状のマルチホーミング技術には、回線選択の方法や経路情報の扱いが複雑であるという問題がある。現状ではこれらの問題のために、必ずしも効率のよいマルチホーム環境が実現されているとはいえない。

そこで本研究では効率のよいマルチホーム環境を実現するために，次世代の基幹プロトコル IPv6 の特徴を利用して，始点アドレスのネットワーク ID（プレフィクス）を回線の状態に応じて書き換える機構を提案する．またその機構を実装し，実験によってその効果を明らかにする．

主な用語

IPv6、マルチホーム、経路選択、アドレス変換

目 次

1　はじめに

近年インターネットは急速に拡大し，利用者数も増加している．それにともなってインターネットの重要性は高まっており，電気や電話などと同様な社会基盤として認知されるようになった．しかし現状では，インターネットで利用する通信システムは電気や電話と比較すると不安定であり，今後さらに可用性（availability）を向上させることが望まれている．

一方，近年 ADSL や FTTH といった安価なアクセス回線が普及しはじめており，容易に一つの組織が複数のアクセス回線を保有可能となってきている．そこで，これらの安価な回線を複数同時に用いることで可用性を向上させるための「マルチホーミング（multi-homing）技術」が注目されている．

しかし現状のマルチホーミング技術には、回線選択の方法や経路情報の扱いが複雑であるという問題がある。現状ではこれらの問題のために、必ずしも効率のよいマルチホーム環境が実現されているとはいえない。

そこで本研究では効率のよいマルチホーム環境を実現するために，次世代の基幹プロトコル IPv6 を用いた、始点アドレスのネットワーク ID（プレフィクス）を回線の状態に応じて書き換える機構を提案する．

本章に次ぐ 2 章では、IPv6 の必要性とマルチホーミングの現状、および関連研究を示す。3 章では、本研究で提案するプレフィクス書き換えの手法、および実際に作成したプレフィクス変換機構について示し、4 章では当該の変換機構を用いた実験の内容について説明する。5 章でその結果を示し、それをもとに最後の 6 章でまとめを行う。

2　IPv6 とマルチホーム

本章では、本研究で用いる IPv6 について説明し、その後マルチホームについての関連研究について説明する。

2.1　IPv6（Internet Protocol Version 6）

IPv6 は、現在多くのネットワーク、およびインターネットで利用されている IPv4（Internet Protocol Version 4）の後継プロトコルとして、1995 年に IETF（Internet Engineering Task Force）が仕様を策定した。

現行の IPv4 は、1974 年に Vinton Cerf、Robert Kahn の両氏によって提案されて以来、29 年にわたってインターネットの基幹プロトコルとして使われている。その間にインターネットは商用化を含む大きな変化をとげたが、そのことによって IPv4 の提案当時には考えられなかった問題が多く現れてきた。

IPv6 では、それらの問題を解決するために様々な仕様変更、拡張が加えられている。

2.1.1　IPv4 の問題点

現在浮き彫りになっている IPv4 の問題点は、主に次の通りである。

インターネットに接続するネットワーク、端末数の増大によるアドレス空間の枯渇　インターネットの前身である ARPANET（Advanced Research Projects Agency Network, 米国防総省による全米規模のネットワーク）の登場以降、多数のネットワークが接続するようになった。しかし当時のインターネットは研究用であり、気心の知れた仲間による性善説的な運用がなされていた面もあり、接続したネットワークへのアドレス割り当てについても大雑把で、必要以上のアドレスが大量に割り当てられることとなった。

また、1990 年代前半にインターネットが商用化されて以降、企業でのビジネスツールとしての利用が飛躍的に増え、また 1990 年代後半には、個人が World Wide Web (WWW) の閲覧などを目的として家庭から電話回線を利用して接続することも増えた。

初期の頃の大雑把な割り当て、および現在の爆発的なユーザ数の増加により、アドレス空間は徐々に枯渇しつつある。

アドレス空間の枯渇に伴うプライベートアドレスの導入による、End-to-End 接続性の喪失
ユーザ数の増加に伴いアドレス空間が枯渇しつつある、という認識が広まった 1990 年前半ころより、アドレス空間の消費を押さえるための技術が様々提案され、実用化された。その一つが 1994 年に IETF によって策定されたプライベートアドレス [RFC1591] と、プライベートアドレスとグローバルアドレスを変換する、NAPT（Network Address Port Translation, アドレス・ポート変換）の機構である。

これらの導入によってアドレス空間の消費ペースはやや緩やかになった。しかし代償として、インターネットの特徴でもあった E2E（End-to-End, 末端の端末間）の接続性が失われることとなった。現在、IP 電話、ビデオ会議システムや、ファイル共有システムなどの E2E での通信を必要とするアプリケーションのほとんどがトリッキーな実装を強いられ、またあるいは NAPT を利用した環境では使えないようになっている。

アドレス空間と物理的な配置が無関係なことによる経路情報の増大　IPv4 ネットワークは、隣あったアドレスのブロックで経路が全く異なることも多く、細かい経路情報が無数に存在している。インターネットに接続するネットワークの増大につれ、接続するルータ間で経路情報をやりとりする際の負荷も急激に増加している。

アドレス空間の消費を押さえるための、もう一つの仕掛けである CIDR（Classless Inter-

Domain Routing）によって、経路情報の増大にもやや歯止めがかかっているが、根本的な解決には至っていない。

2.1.2 IPv6 による解決策

これらの問題を解決するため、IPv6 では以下の特徴が取り入れられた。

128ビットのアドレス空間 アドレス空間が現在の 32 ビット、総数で約 4.2×10^9（42 億）個から、128 ビット、総数で約 3.4×10^{38}（340 澗）個に増えた（表 2.1）。これは問題 (1) を解決し、それにともなって (2)、(3) を解決する。

表 2.1: IPv4 と IPv6 のアドレス空間のビット数とアドレス個数

	ビット数	アドレスの個数
IPv4	32	4,294,967,296
IPv6	128	340,282,366,920,938,463,463,374,607,431,768,211,456

アドレスは便宜的に、ネットワーク ID とインタフェース ID に分けられる。これらは可変長で、ネットワーク ID が n ビットの場合、インタフェース ID は $128 - n$ ビットとなる。ネットワーク ID は IPv6 では「プレフィクス (Prefix)」と呼ぶことが多い。n は「プレフィクス長 (Prefix length)」であり、CIDR の導入された IPv4 と同様に、必要に応じてアドレスの末尾に /n と表記する。

階層化され、集約可能なグローバルアドレス IPv6 の 128 ビットのアドレスの構造は (図) の通りである。先頭 3 ビットが 001b のアドレスは「集約可能なグローバル・ユニキャストアドレス (Aggregatable Global Unicast Address)」といい、階層的な構造を持つ。

先頭 3 ビットに続く 13 ビットを TLA ID(Top-Level Aggregation Identifier, 最上位階層集約識別子) といい、APNIC や ARIN などの RIR(Regional Internet Registry) が、世界的、国レベルの大規模 ISP の申請に基づき割り当てる。

続く 8 ビットは、将来のために予約されている。その次の 24 ビットを NLA ID(Next-Level Aggregation Identifier) といい、一般の ISP が、TLA ID を持つ ISP に申請して、TLA ID を持つ ISP が割り当てる。

続く 16 ビットは SLA ID(Site-Level Aggregation Identifier) といい、ISP が各利用者に割り当てる。

このように階層化されていることで、経路情報を集約することができる。これは問題 (3) を解決する。

2.1.3 IPv6 の新機能

その他にも、以下のような特徴が取り入れられた。

マルチキャストの標準化 マルチキャストが標準化され、IPv4 のブロードキャストは、特殊な形のマルチキャストで置き換わった。IPv6 ホストは、ネットワークに接続した時点で自動的に「全ノード・マルチキャストアドレス (`ff02::1`)」に参加し、これが IPv4 でいうブロードキャストアドレスのかわりとなる。

アドレス自動設定機能 現在の IPv4 では、ホストをネットワークに接続するだけでは原則として通信ができず、自分の IP アドレスを設定する必要がある。またこの設定を省くには、ネットワークに Dynamic Host Configuration Protocol (DHCP) サーバが必要となる。

IPv6 では、DHCP サーバのないネットワークでも、ネットワーク内で重複しない IP ア

ドレスを、端末自身が自動生成することで、IP アドレスを手動設定することなく通信を行うことができる。

具体的には、上流回線に接続するルータが、上流回線のプレフィクスとデフォルトゲートウェイを、定期的にクライアントネットワークに向けて全ノード・マルチキャストで通知する。あるいはクライアントネットワークに接続する端末の要求に従って、ルータが端末に通知する。

端末は通知されたプレフィクスと、自らが持つインタフェースの MAC アドレスを 64 ビットに拡張して、アドレスを生成する。

プレフィクスが 3ffe:501:2c36:100::/64 で、端末の持つインタフェースの MAC アドレスが 00:02:b3:aa:bb:cc の場合、生成される IPv6 アドレスは、3ffe:501:2c36:100:202:b3ff:feaa:bbcc となる。48 ビットの MAC アドレスを 64 ビットに拡張するために、上位 24 ビットと下位 24 ビットの間に ff、fe というコードが入っているのが分かる。

エニーキャストアドレスの導入。(XXX ret.) CDN や Web サーバなど、同じデータを複数の端末が持っているとき、そのうちいずれか一つの端末と通信ができればよい、という目的のために、エニーキャストアドレスが導入された。複数の端末に同じアドレスを割り当て、ルータが最も近い端末を判断し、通信を行う。

2.1.4 IPv6 のアドレス表記法

加えて、特徴ではないが、IPv6 のアドレスの表記法も IPv4 に比べて大きく変わった。

128 ビットのアドレスは 16 ビット単位で 8 つのブロックにわけ、それぞれのブロックの区切り文字を：として 16 進数で示す（例 1 参照）。

例 1） `3ffe:501:2c36:387e:0000:0000:0000:0001`

ただしこのように表記が長くなってしまうため、規則に従って省略することができる。

各ブロックの上位の 0 は省略することができる。ただし 1 桁以下には省略できない（例 2 参照)。また、連続した 0 のブロックは、アドレス中 1 カ所に限って、:: と省略することができる（例 3 参照)。

連続した 0 のブロックが 2 カ所以上ある場合は、その片方だけ省略できる（例 4 参照)。

例 2）	3ffe:501:2c36:387e:0:0:0:1
例 3）	3ffe:501:2c36:387e::1
例 4）	2001:200:0:0:101:0:0:0:1
	→ 2001:200::101:0:0:0:1 ……○
	→ 2001:200:0:0:101::1 ……○
	→ 2001:200::101::1 ……×

以下に、いくつかの省略の例を示す。

例）	ff02:0:0:0:0:0:0:1	→	ff02::1
	3ffe:501:2c36:0:0:0:0:0	→	3ffe:501:2c36::
	0:0:0:0:0:0:0:1	→	::1
	0:0:0:0:0:0:0:0	→	::

2.2 マルチホーム

3　プレフィクス変換と経路選択

IPv6 におけるマルチホーミングは、各端末に割り当てられた複数の始点アドレスを選択することで実現可能であると前章で触れた。しかしその実現には問題がある。

一つは、始点アドレスを変えるだけでは実現できない、という問題である。始点アドレスを回線に応じて書き換えることで、通信相手から戻ってくるときの経路を指定することはできるが、通信相手に行くまでの経路を指定することができない。

この問題を解決するために、一般に使われている「宛先ごとの経路表」とは別に、「始点アドレスごとの経路表」をもうける、という手法が提案されている。[1]　しかしこの方法を用いるにはルータの設計を変える必要があり、実現は容易ではない。

もう一つは、始点アドレスをどのように選択するか、という問題である。適切な上流回線を選ぶには回線の状態を知り、使用する回線を判断しなければならない。しかし、ネットワークに接続された端末のすべてが上流回線の状態を知り、それに基づいて端末自身で個別に判断することは、端末にかかる負担が大きい。

そこで本研究では、個々の端末にかわって、上流回線に接続するルータがそれらの作業を受け持つ仕組みを提案する。この場合ルータは、それぞれの回線の状態を計測して蓄積し、どの上流回線を利用すれば効率よく通信できるかの判断を行い、通信相手からの戻りの経路を決めるためにパケットの始点アドレスのプレフィクスの書き換えを行う。

この仕組みを用いることで、端末は利用する上流回線を選択する必要がなくなり、負担は軽減される。またマルチホームであることを意識せずに、マルチホームであることの利益を享受できるようになる。

本章では、本研究で提案するプレフィクスの書き換えによるマルチホーミングの実現手法と、その手法を実現するプレフィクス変換機構の実装概要を示す。

3.1 プレフィクス変換の構造

本研究では、IPv6 におけるプレフィクスの固定長割り当て、およびアドレスの自動生成の機能を用いて、プレフィクス変換を行う。プレフィクス変換が可能な原理について以下に示す。

２章で説明したように、現在割り振られている IPv6 アドレスは、は階層構造をしている。ある組織が IPv6 の接続を希望した場合、上流の ISP は自分の持つアドレスブロックの中からアドレスを割り当てる。このときの単位は、末端の組織であれば一般的に /48 である。

組織はこの /48 のアドレスブロック、つまりアドレスの下位 80 ビットの部分を自由に使うことができる。多くの場合、その 80 ビット中の上位 16 ビットを「サブネット」の識別子として、下位 64 ビットを、各端末のインタフェースに割り当てる「インタフェース ID」として利用する。

複数の ISP と契約した場合、それぞれ割り当てられるアドレスブロックも同様に /48 の固定長である。

たとえば、ISP ではないが、九州工業大学が接続する 3 つの IPv6 上流回線のプレフィクスは次の通りである。

表 3.1: 九州工業大学の上流回線とそのプレフィクス

上流回線名	割り当てられているプレフィクス
WIDE Project	3ffe:501:2c36::/48
Japan Gigabit Network (JGN) v6	3ffe:516:8180::/48
九州ギガポップ (QGPOP)	2001:248:185::/48

一方、IPv6 では２章で説明したように、IP アドレスの自動設定機能が利用できる。

この機能を利用すると、各上流回線のプレフィクスのついたアドレスは、インタフェース ID は MAC アドレスから生成されるため、すべて同一となる。

サブネットを 100 として、MAC アドレスが 00:02:b3:aa:bb:cc とすると、各上流回線に対応したアドレスは以下のようになる。

表 3.2: 上流回線ごとの自動生成アドレス

上流回線名	各回線に対応したアドレス
WIDE Project	3ffe:501:2c36:100:202:b3ff:feaa:bbcc
Japan Gigabit Network (JGN) v6	3ffe:516:8180:100:202:b3ff:feaa:bbcc
九州ギガポップ (QGPOP)	2001:248:185:100:202:b3ff:feaa:bbcc

このことから、端末に１つのアドレスしか割り当てられていない場合であっても、プレフィクスを通信を行いたい上流回線に応じたプレフィクスに書き換えることで通信が行えることがわかる。

3.2 プレフィクス変換機構の実装

マルチホームのためのプレフィクス変換機構に必要なのは、次の機能である。

1. 上位回線の状態を把握する機能
2. 通信の相手先に応じて利用する回線を決定する機能
3. 帰りの通信のために、利用する回線に応じてパケットの始点アドレスを書き換える機能
4. 行きの通信のために、利用する回線に応じて静的経路を設定する機能

これらの機能のうち３番は、すべてのパケットの始点アドレスを書き換える必要があるため、基本ソフトウェア 'OpenBSD 3.2' のカーネルと、それに付属する NAT 機能付きのパケットフィルタ 'pf' に機能を追加することで実装した。それ以外の機能は perl スクリプトで実装した。

pf は、カーネルの動作レベルで実行されるカーネルモジュールであり、原則としてファイルの読み書きや子プロセスの起動ができず、またそれ以外にも様々な関数が利用できないため、行える操作が強く制限される。

またカーネルモジュール内でエラーが発生して実行を継続できなくなった場合、システム全体が停止してしまうため、可能な操作であっても、失敗するおそれのある操作は極力控えるべきである。

以上より、カーネル内部でしなければならない操作である 3 番のみ pf に実装した。それ以外の機能は、一般のプログラムが動作する環境（ユーザランド）で動くプログラムとして実装した。

また、pf と情報をやりとりするための制御ツールである pfctl にも手を加えた。

本節では、本研究で機能追加を行った、pf、pfctl、および新たに作成した、プレフィクス変換制御および経路制御の機構 ptm について説明する。

3.2.1 pf

pf は、OpenBSD に備わっているカーネルモジュールで、もともとパケットフィルタとしての機能と、NAT としての機能をあわせもつ。pf を有効にした場合、OS が操作するすべてのパケットは pf を通過する。pf はそれらのパケットを、あらかじめ設定されたルールに基づいて破棄（フィルタ）したり、中身を書き換える（NAT）ことができる。

これに加えて本研究のために、あらかじめ設定されたルールに基づいて始点アドレスのプレフィクスを書き換える機能と、通信相手の終点アドレスを蓄積する機能、および蓄積したアドレスを提供する機能を加えた。

- 設定されたルールに基づいて始点アドレスのプレフィクスを書き換える機能

  pfctl を用いて設定されたルールにもとづいて、出力パケットの始点アドレスを書き換える。ルールは「始点アドレスのプレフィクス」「終点アドレスのプレフィクス」「書き換え後のプレフィクス」となっており、パケットの始点アドレスのプレフィクスがルールの「始点アドレスのプレフィクス」と適合し、かつ終点アドレスのプレフィクスがルールの「終点アドレスのプレフィクス」と適合した場合、始点アドレスのプレフィクスを「書き換え後のプレフィクス」に書き換える。

入力パケットの場合はルールを逆に適用して、終点アドレスを書き換える。つまりパケットの始点アドレスのプレフィクスとルールの「終点アドレスのプレフィクス」が適合し、かつパケットの終点アドレスのプレフィクスとルールの「書き換え後のプレフィクス」が適合した場合、終点アドレスのプレフィクスを「始点アドレスのプレフィクス」に書き換える。

- 通信相手の終点アドレスを蓄積する機能

  通信相手に応じて効率のよい回線を選ぶためには、どのような相手と通信をしているかを知る必要がある。そのために、pf で、フィルタを通るパケットの終点アドレスを蓄積しておく。蓄積したアドレスは、次項で説明するユーザランドのプログラム pfctl の要求によって出力し、また削除する。

3.2.2 pfctl

pfctl は、pf がユーザランドからのアクセスのために提供している /dev/pf という仮想デバイス（入出力インタフェース）を介して、pf とのデータのやりとりを行うための、pf 付属のプログラムである。

pfctl は /etc/pf.conf という pf のルールが書かれた設定ファイルを読み込み、それを pf が解釈できる形式に変換して、/dev/pf を通じて設定する。

今回、pf にプレフィクス変換のための新しいルールを追加したので、それにあわせて pfctl でもそのルールを認識できるように機能を追加した。

3.2.3 ptm

ptm は、プレフィクス変換に関するほとんどの機能を実装した、ユーザランドのプログラムである。本研究のために新たに作成した。

ptm は、上位回線の状態を把握する機能、通信の相手先に応じて利用する回線を決定する機能、および行きの通信のために、利用する回線に応じて静的経路を設定する機能を持つ。

通信の相手先と、その相手先に応じて決められた上流回線の対応づけを、本論文の中では「PT 経路」と呼ぶ。PT 経路に含まれる情報は次の通りである。

- 通信相手先ネットワーク

  この PT 経路が司る通信相手先ネットワークのアドレス。　終点アドレスごとに最適な上流回線を選択する方法では PT 経路が飛躍的に増大する可能性があるため、経路表にある経路単位で最適な上流回線を選択する。そのために、通信を行った終点アドレスが、経路表上のどの経路に適合するかを見る。

- 代表アドレス

  この PT 経路の状態を測定するために ping を打つ端末の IP アドレス。

- 使用インタフェース

  この PT 経路でが現在利用している上流回線に接続するインタフェース。

- 最終通信時刻

  この PT 経路を用いて、最後に通信相手先ネットワークと通信を行った時間。　設定された PT 経路は、設定したままでは時間の経過に伴い増大するばかりである。そのため上流回線決定のために行う ping の回数も増大し、効率が低下する。そのため何らかのタイミングで PT 経路の削除を行わなければならない。

  そのために本機構では最終通信時間を記録しておき、この時間から、利用者の指定する時間が経過した場合、通信が行われていないので PT 経路を維持する必要がないものとみなし、PT 経路を削除する仕組みを組み込んだ。

- 最終 PING 時刻

  この PT 経路に対して、最後に ping を行って最適な回線の選択を行った時間。この時間から、利用者の指定する時間が経過した場合、ptm は回線の選択するために ping を行う。

- 回線遷移数

  各上流回線において ping を行い、その遅延を測定した結果、現在選択されている回線と異なる回線の方が遅延が小さかった場合に加算される数。この数が利用者の設定する数に達するまでは、回線を切り替えない。　ただし、現在選択されている回線と、遅延のもっとも小さかった回線が同一の場合は０に戻される。連続してｎ回以上、他の回線の方が遅延が小さかった場合にのみ切り替えることで、瞬間的な回線状態によって使用する回線が切り替わることを防止するために用いられる。

ptm は cron によって定期的に呼び出され、処理を行う。処理の流れは次の通りである。

1. pf から、蓄積された終点アドレスの取得
2. 現在設定されている PT 経路を確認。
3. 情報更新が必要な PT 経路に対して ping を実施
4. PT 経路を更新する
5. pf.conf と pfctl を使って pf にルール設定
6. route add/delete を使って、カーネルに静的経路の設定

以下で、それぞれの処理の流れの概略を説明する。

- pf から、蓄積された終点アドレスの取得

  ptm は pfctl を用いて、pf が蓄えている終点アドレスのリストを取得する。

- 現在設定されている PT 経路を確認。

  pf から取得した各終点アドレスに対し、以下の操作を行う。

  1. 取得したアドレスに対応する経路を経路表より見つけ出し、アドレスと対応づける。

     取得したアドレスに対応する経路がない場合、あるいはデフォルトゲートウェイの場合、取得したアドレスを /128 の経路とみなして設定を行う。

2. 対応する経路に、PT 経路がすでに定義されているかどうかを確認する。

   1 番で見つけた、取得したアドレスに対応する経路に PT 経路がすでに定義されている場合、その PT 経路の「最終通信時刻」を今の時間に更新し、かつ取得したアドレスをその PT 経路の「代表アドレス」として設定しなおす（より最近に通信した端末の方が、一番最初に通信を行った端末よりも生存確率が高く、かつ現状を示しやすいはずである）。

   PT 経路が定義されていない場合、新たな PT 経路であることを示すフラグをセットして、PT 経路を設定する。

3. PT 経路の「最終 PING 時刻」が、ある一定の時間を経過しているものを、再 PING を行うために抜き出す。

   ある一定の時間は利用者が自由に設定できる。設定がない場合は 7,200 秒（2 時間）とみなす。最終 PING 時刻から設定された時間以上経過している場合、通信状況を確認するために PING を行うものとし、そのために PT 経路を抜き出す。

   また、新たな PT 経路であることを示すフラグがセットされているものも同様に抜き出す。

4. PT 経路の「最終通信時刻」がある一定の時間を経過しているものを、設定削除のために抜き出す。

   ある一定の時間は利用者が自由に設定できる。設定がない場合は 86,400 秒（24 時間）とみなす。最終通信時刻から設定された時間以上経過している場合、通信が行われなくなったものとみなして、設定を削除するために PT 経路を抜き出す。

- 更新が必要な PT 経路に対して ping を実施

  更新が必要だとして抜き出された PT 経路に対して、ping を実施する。

PT 経路の代表アドレスに対して、各上流回線においてまず 1 回 ping を実行し、その結果を見る。ping が到達し、RTT（Round Trip Time, 往復遅延時間）が計測でき

た回線については、さらに４度の ping を実行し、その平均を回線の往復遅延とする。

- PT 経路を更新する

  計測できた回線の中からもっとも遅延の小さい回線を選び出す。

  その回線が、現在 PT 経路で選択されている上流回線でなかった場合、PT 経路の「回線遷移数」を１だけ加算する。さらに「回線遷移数」が利用者の指定した値を超えた場合、選び出した回線を、使用する回線として設定する。

  そうでない場合は、現在 PT 経路で選択されている回線を、そのまま上流回線として用いる。

  （模式図）

- pf.conf と pfctl を使って pf にルール設定

- route add/delete を使って、カーネルに静的経路の設定

  PT 経路の情報を pf.conf の形式にあわせて変換し、pf.conf に加えて pfctl を実行してプレフィクス変換の状態を更新する。

  加えて、route コマンドの形式にあわせて変換し、route add/delete を実行して静的経路の状態を更新する。

4 実験

実装した機構を用いて、効果を示すために実験を行った。

4.1 実験環境

本節では、本章の実験を行った環境について説明する。

用いた上流回線は次の通りである。

WIDE: WIDE Project （プレフィクス 3ffe:501:2c36::/48）

JGN: Japan Gigabit Network v6（プレフィクス 3ffe:516:8180::/48）

QGPOP: 九州ギガポップ （プレフィクス 2001:248:185::/48）

これらの上流回線と接続するルータを各1台ずつ設置した。またこれらの各ルータと、本機構を導入した「PT ルータ」を接続した。さらに PT ルータから、クライアントネットワークへの Hub に接続した。Hub に、クライアント用の PC を一台接続した。

各上流回線のプレフィクスは上で示した通りである。

各上流回線のルータと PT ルータを接続するセグメントのネットワークアドレスは、それぞれ 3ffe:501:2c36:3871::/64、3ffe:516:8180:3872::/64、2001:248:185:3873::/64 とした。各接続セグメントでは、ルータ側のインタフェースの IP アドレスを ::1（3ffe:501:2c36:3871::1 など）とし、PT ルータ側のインタフェースの IP アドレスを ::2（3ffe:501:2c36:3871::2 など）とした。

またクライアントネットワークアドレスは 3ffe:501:2c36:387e::/64 とし、クライアントネットワークに接続する PT ルータのインタフェースの IP アドレスを 3ffe:501:2c36:387e::1 とした。PT ルータはクライアントネットワークにおけるアドレスの自動設定を有効にするため、クライアントネットワークに対して「ルータ広告（Router Advertise）」を行う。ク

ライアント用の PC の IP アドレスは自動生成される。

以下に接続の模式図を示す。

（実験環境の図）

4.2 使用する回線の決定

本研究では，複数ある回線の中から最適な回線を選択するための情報として，通信相手までの往復遅延を用いた．具体的には，利用可能な各回線において定期的に通信相手までの往復遅延を計測し，もっとも遅延の小さな回線を選択し，切り替える．

回線の切り替えを行うと始点アドレスが変わるため，その時点で行っている通信は途切れてしまう．そのため頻繁に切り替えるのは好ましくない．しかし単純にもっとも遅延の小さな回線を選択すると，回線の切り替えが頻発し，通信の切断回数を増やすことになる．

そこで本研究で実装する機構では，使用する回線を決定するために以下のパラメータを導入した．

n: 回線切り替えのための計測回数．現在選択されている回線の遅延よりも他の回線の遅延の方が小さい状態が n 回連続した場合に回線を切り替える．この変数により短期的な変動によって回線が切り替わることを防止する．

m: 回線間の遅延に有意な差があると見なすしきい値．回線間の遅延の差が m ミリ秒以上であった場合，差があるものとみなす．この変数により，回線間の遅延の差が小さいときに，遅延の微小な変化によって回線が切り替わることを防止する．

これらのパラメータ n, m の最適値を求めるために、以下の実験を行った。

複数の端末を対象として、実際のプレフィクス変換機構を動作させた時と同様の方法（各上流回線において５分に１回、１回ごとに４度の ping を行う）で、対象となる端末までの往復遅延を計測し、その結果の平均をそれぞれの上流回線ごとに蓄積する。

対象とする端末は次の通りである。

- 国内
  - kddilab.6to4.jp
  - www.wide.ad.jp
  - www.hitachi.co.jp
  - www.imnet.ad.jp
  - www.iij.ad.jp
  - www.kame.net
- イギリス
  - 6to4.ipv6.bt.com
- アメリカ
  - www.netbsd.org
  - playground.sun.com

結果の見方は以下の通りである。

- 各先頭行が、試行を行った対象端末のホスト名、あるいは IP アドレス
- その次の行から３行は、WIDE、JGN、QGPOP の各回線の状態である。全体の計測のうち、もっとも遅延が小さいと判断された回数、およびその割合である。その右側の数値はそれぞれ ping の結果の最小値、平均値、最大値、およびその標準偏差である。failed の項目は、３回線すべてで ping が失敗した回数を示す。

- その下の数字列は、各回の試行においていずれの回線が選択されたかを示す。1 は WIDE、2 は JGN、3 は QGPOP である。ただし * は上流回線のいずれも試行に失敗したことを示す。

まず、国内の IPv6 端末における試行の結果を表に示す。この表より、国内の IPv6 端末においては安定的に JGN の遅延が短いことがわかる。

表 4.1: kddilab.6to4.jp

kddilab.6to4.jp			
wide:	0	(0.0%)	min/avg/max/std-dev = 24.78/25.06/33.48/0.59
jgn:	268	(99.3%)	min/avg/max/std-dev = 19.15/19.56/21.52/0.16
qgpop:	2	(0.7%)	min/avg/max/std-dev = 19.77/20.34/37.57/1.15
failed:	0	(0.0%)	
2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 3 3 2			

次に海外の IPv6 端末における試行の結果を表に示す。6to4.ipv6.bt.com はイギリス、残る 2 つはアメリカの端末である。英国に関しては JGN と QGPOP がよく利用されているが、頻繁に最小遅延の回線が変わっている。

米国の端末はほぼ WIDE で安定である。

この結果より、国内は JGN、アメリカは WIDE でほとんど安定していることが分かった。しかしイギリスに関しては最小遅延の揺らぎが大きいことも分かった。

表 4.2: www.wide.ad.jp

www.wide.ad.jp			
wide:	0	(0.0%)	min/avg/max/std-dev = 30.35/30.90/111.78/4.95
jgn:	270	(100.0%)	min/avg/max/std-dev = 19.74/20.14/21.84/0.37
qgpop:	0	(0.0%)	min/avg/max/std-dev = 28.08/28.51/34.32/0.57
failed:	0	(0.0%)	
2 2			

表 4.3: www.hitachi.co.jp

www.hitachi.co.jp			
wide:	0	(0.0%)	min/avg/max/std-dev = 47.78/48.05/56.73/0.92
jgn:	270	(100.0%)	min/avg/max/std-dev = 37.34/37.59/40.73/0.28
qgpop:	0	(0.0%)	min/avg/max/std-dev = 44.31/57.63/129.73/13.35
failed:	0	(0.0%)	
2 2			

表 4.4: www.imnet.ad.jp

www.imnet.ad.jp			
wide:	0	(0.0%)	min/avg/max/std-dev = 29.67/29.84/32.23/0.27
jgn:	270	(100.0%)	min/avg/max/std-dev = 19.21/19.44/21.95/0.22
qgpop:	0	(0.0%)	min/avg/max/std-dev = 33.34/39.93/162.48/17.65
failed:	0	(0.0%)	

2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
2 2 2 2 2 2 2 2 2 2

表 4.5: www.iij.ad.jp

www.iij.ad.jp			
wide:	7	(2.6%)	min/avg/max/std-dev = 29.78/30.29/47.71/2.10
jgn:	263	(97.4%)	min/avg/max/std-dev = 19.26/20.03/41.29/2.40
qgpop:	0	(0.0%)	min/avg/max/std-dev = 32.79/39.79/128.95/13.87
failed:	0	(0.0%)	

2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 2 2
2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1
2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
2 2 2 2 2 2 2 2 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
2 2 2 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2
2 2 2 2 2 2 2 2 2 2

表 4.6: www.kame.net

www.kame.net			
wide:	2	(0.7%)	min/avg/max/std-dev = 30.14/30.30/32.23/0.17
jgn:	253	(93.7%)	min/avg/max/std-dev = 20.79/22.75/71.58/7.23
qgpop:	15	(5.6%)	min/avg/max/std-dev = 22.84/31.11/77.63/7.04
failed:	0	(0.0%)	

2 2 2 2 2 2 2 2 2 2 1 2 2 2 2 2 2 2 2 3 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 3 2 2 2 2
2 2 2 2 3 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 3 2 2 2 2 2 2 2 3 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
2 2 2 2 2 3 2 2 3 2 2 2 2 2 2 2 2 3 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 3 2 2 2 2 2
2 2 3 2 2 2 2 2 2 2 2 3 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 3 2 2 2 2 2 2 2 3 2 2 2
2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 3 2 2 2 2 2 2 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
3 2 2 2 2 2 2 2 2 2

表 4.7: 6to4.ipv6.bt.com

6to4.ipv6.bt.com			
wide:	10	(3.7%)	min/avg/max/std-dev = 300.35/307.15/420.80/12.18
jgn:	162	(60.0%)	min/avg/max/std-dev = 294.20/301.96/390.31/14.22
qgpop:	98	(36.3%)	min/avg/max/std-dev = 294.77/301.30/378.46/12.20
failed:	0	(0.0%)	

2 3 2 2 3 3 2 1 2 2 3 2 3 3 1 2 2 3 3 2 3 1 3 3 1 3 1 3 2 3 2 3 3 1 1 3 3 2 2 2 2 2 2 3 3 3 3 2 2 3 3 3
2 2 1 2 2 2 2 3 3 3 2 2 2 3 2 2 3 3 3 3 2 3 3 2 2 2 2 3 3 2 2 3 2 3 2 2 2 2 2 2 2 2 2 2 3 2 2 2 2 2 2
1 2 1 2 2 3 3 3 2 3 2 2 3 3 2 3 3 2 2 3 3 2 3 3 3 2 2 2 3 2 2 3 2 2 3 2 2 2 2 2 3 2 2 2 3 3 3 2 3 2 3 3
3 3 2 3 3 2 2 2 2 3 2 3 2 2 2 3 3 2 3 3 2 3 2 2 3 2 3 3 3 2 2 2 2 3 3 2 3 2 2 3 2 2 2 2 2 2 3 2 2 3 2 3
3 2 2 3 2 2 2 2 2 3 3 2 2 2 2 2 2 2 2 3 3 2 2 2 2 2 3 2 2 2 2 2 2 2 3 2 3 2 2 2 2 2 2 2 2 2 2 3 2 2 2 2 2
2 3 2 2 2 3 2 2 2 2

表 4.8: www.netbsd.org

www.netbsd.org			
wide:	261	(96.7%)	min/avg/max/std-dev = 133.41/134.89/186.61/5.43
jgn:	2	(0.7%)	min/avg/max/std-dev = 244.86/246.37/290.69/3.62
qgpop:	7	(2.6%)	min/avg/max/std-dev = 142.50/147.52/169.51/2.10
failed:	0	(0.0%)	

2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 3 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 3 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 3 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 1 3 1 1 1 1 1 1
1 1 1 1 1 1 3 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 3 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 3 1 1 1
1 1 1 1 1 1 1 1 1 1

表 4.9: playground.sun.com

playground.sun.com			
wide:	258	(95.6%)	min/avg/max/std-dev = 138.97/141.60/183.01/5.79
jgn:	0	(0.0%)	min/avg/max/std-dev = 251.05/253.60/300.61/6.19
qgpop:	12	(4.4%)	min/avg/max/std-dev = 152.82/154.37/225.60/5.05
failed:	0	(0.0%)	

1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 3 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 3 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 3 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 3 3 1 1 1 1 1 3 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 3 1 1 1 1 1 1 1 1 3 1 1 3 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 3 3 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 3 1 1

パラメータ n、m を探るための実験を行う上で、最小遅延の揺らぎは大きな方がよい。そこで日本とアメリカ以外の端末に対して、同じ条件で、数を増やして計測を行った。

対象とする端末は以下の通りである。

韓国 3ffe:3900:3::2(korea-6tapsl.6tap.net)

中国 3ffe:3200::1(cernet.ipv6.net.edu.cn)

タイ 3ffe:b80:2:1a2::2(cnr-coe-psu-th.tsps1.freenet6.net)

マレーシア 3ffe:80d0:fffc:10::43

イギリス 2001:630:0:1::1e(soton.tun.ipv6.ja.net)

ドイツ 3ffe:80a0:0:f00a::21(Tu17-BERKOM–NTT-ECL.r01.ipv6.berkom.de)

フランス 2001:660:80:421a::2

以下はその結果である。

フランスは測定することができなかった。これは、各上流回線において ping を行う際に「ソースルーティング機能」を利用しており、対象の端末までの間に、この機能を制限しているルータが存在する場合、計測ができないためである。

これらの結果は、$n = 0$、$m = 0$ の結果である。

次に、これらの蓄積されたデータを用いて、n と m を変化させたときの、切り替えが必要と判断された回数を図に示す。

これらの図より、n、m のいずれも大きくするほど切り替え回数は減ることがわかる。特に n を 2 以上とすると切り替え回数は大幅に減り、3 以上では n、m の値に因らず同程度の切り替え回数となることが分かる。

表 4.10: 3ffe:3900:3::2(korea-6tapsl.6tap.net)

3ffe:3900:3::2(korea-6tapsl.6tap.net)			
wide:	82	(4.7%)	min/avg/max/std-dev = 429.04/431.87/2407.76/48.04
jgn:	906	(52.3%)	min/avg/max/std-dev = 389.63/435.52/568.98/44.80
qgpop:	743	(42.9%)	min/avg/max/std-dev = 428.48/429.57/472.90/3.43
failed:	0	(0.0%)	

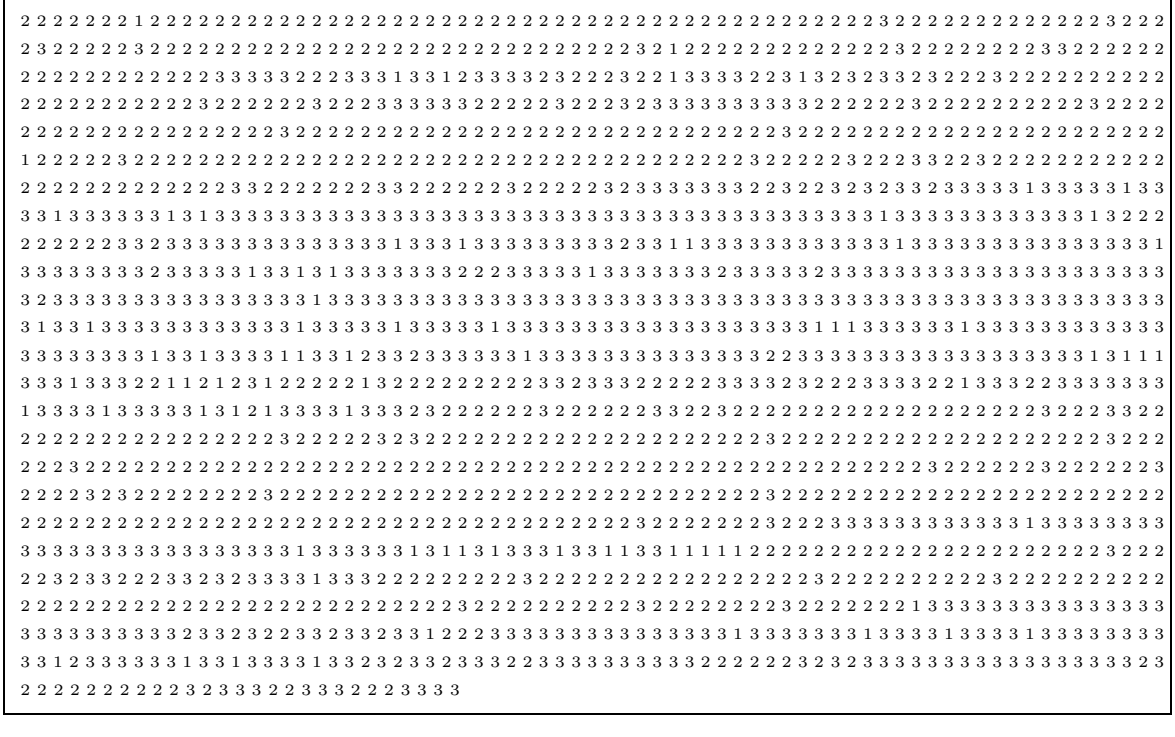

表 4.11: 3ffe:3200::1(cernet.ipv6.net.edu.cn)

3ffe:3200::1(cernet.ipv6.net.edu.cn)			
wide:	358	(20.7%)	min/avg/max/std-dev = 326.41/393.30/2781.04/127.00
jgn:	17	(1.0%)	min/avg/max/std-dev = 319.41/689.41/3325.41/159.59
qgpop:	1353	(78.2%)	min/avg/max/std-dev = 319.12/375.32/3302.67/129.04
failed:	3	(0.2%)	

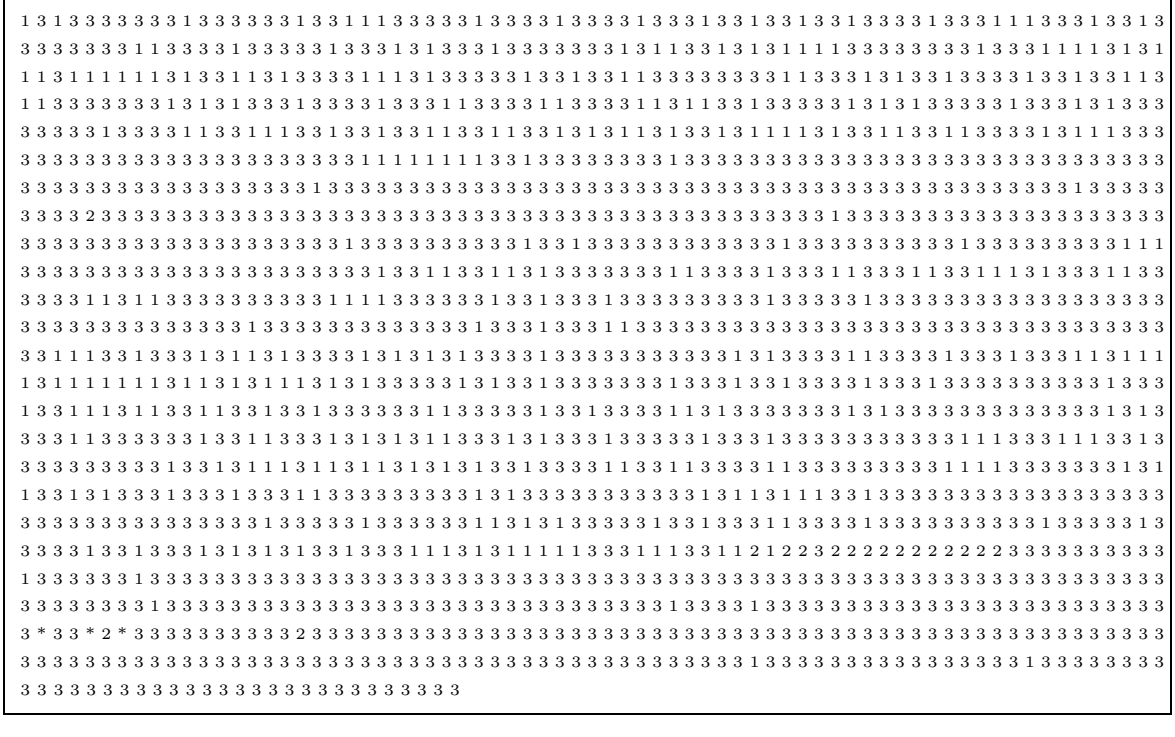

表 4.12: 3ffe:b80:2:1a2::2(cnr-coe-psu-th.tsps1.freenet6.net)

3ffe:b80:2:1a2::2(cnr-coe-psu-th.tsps1.freenet6.net)			
wide:	1125	(65.0%)	min/avg/max/std-dev = 575.59/642.84/1058.03/49.75
jgn:	230	(13.3%)	min/avg/max/std-dev = 584.60/703.52/3265.89/92.56
qgpop:	351	(20.3%)	min/avg/max/std-dev = 606.11/675.26/1499.44/59.13
failed:	25	(1.4%)	

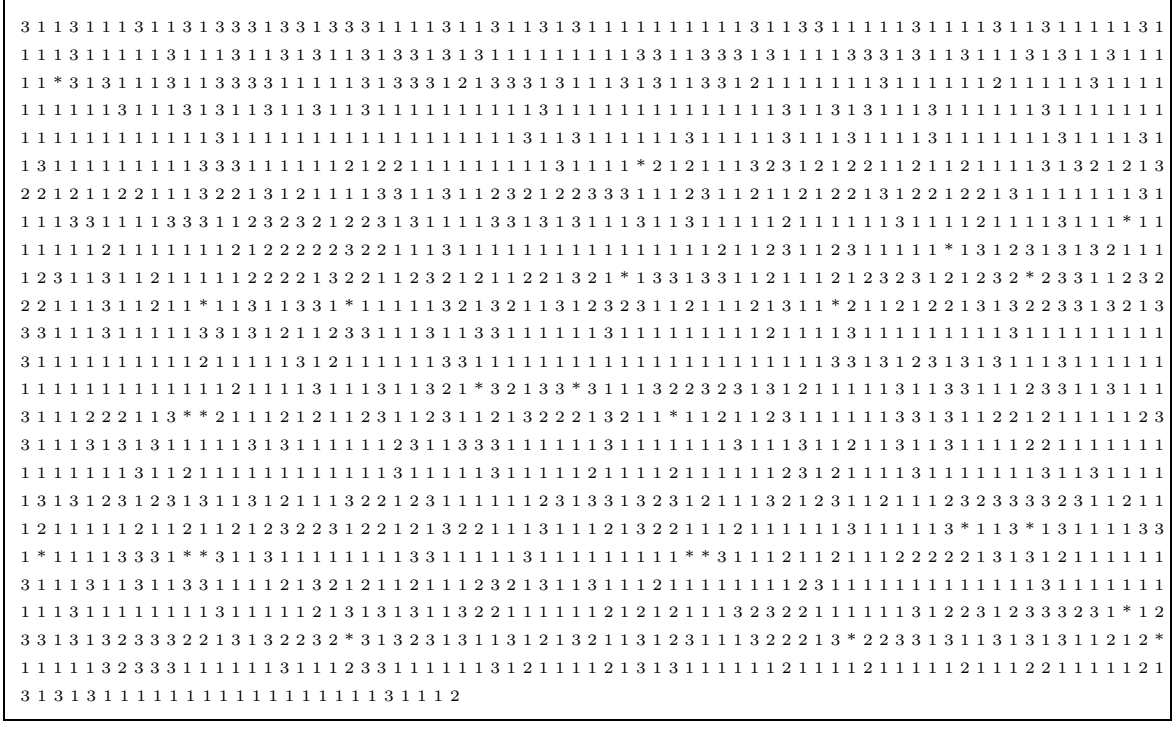

表 4.13: 3ffe:80d0:fffc:10::43

3ffe:80d0:fffc:10::43			
wide:	857	(49.5%)	min/avg/max/std-dev = 534.62/795.47/5098.66/185.08
jgn:	0	(0.0%)	min/avg/max/std-dev = 667.84/681.82/701.79/17.75
qgpop:	857	(49.5%)	min/avg/max/std-dev = 517.68/786.25/4632.45/199.73
failed:	17	(1.0%)	

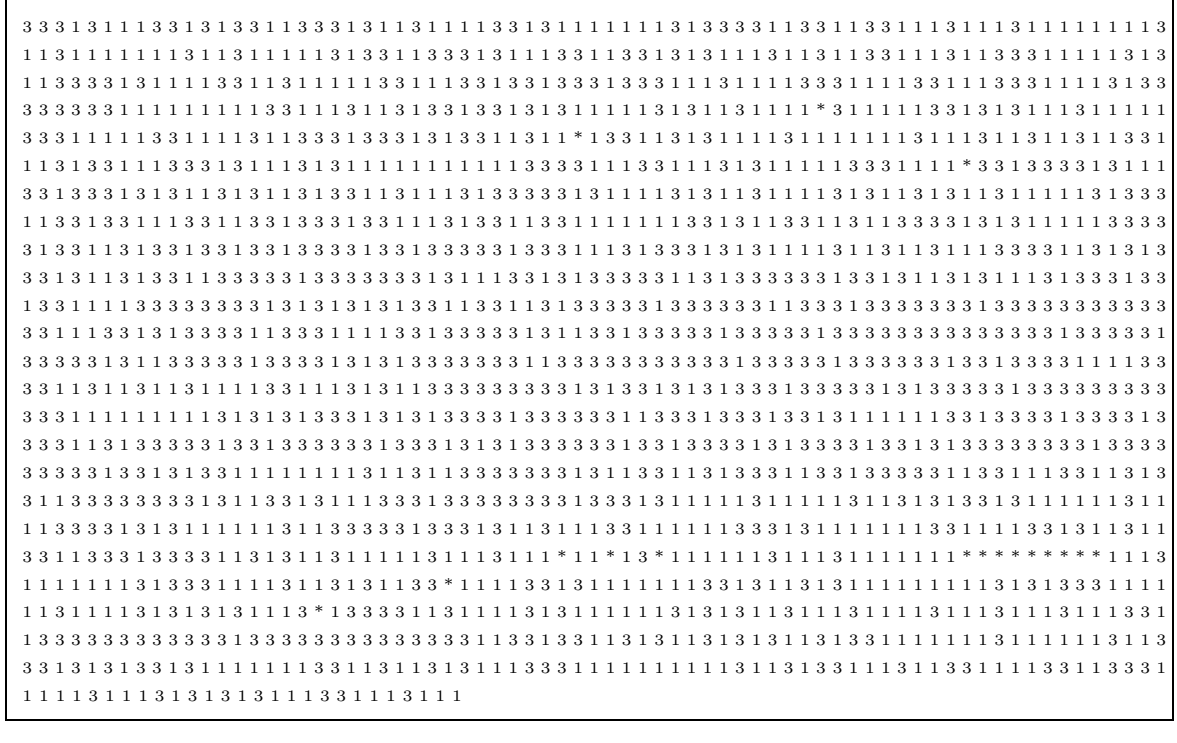

表 4.14: 2001:630:0:1::1e(soton.tun.ipv6.ja.net)

2001:630:0:1::1e(soton.tun.ipv6.ja.net)			
wide:	55	(3.2%)	min/avg/max/std-dev = 310.07/343.19/1422.68/39.59
jgn:	724	(41.8%)	min/avg/max/std-dev = 281.55/310.96/3041.42/73.19
qgpop:	948	(54.8%)	min/avg/max/std-dev = 289.71/311.82/464.82/20.99
failed:	4	(0.2%)	

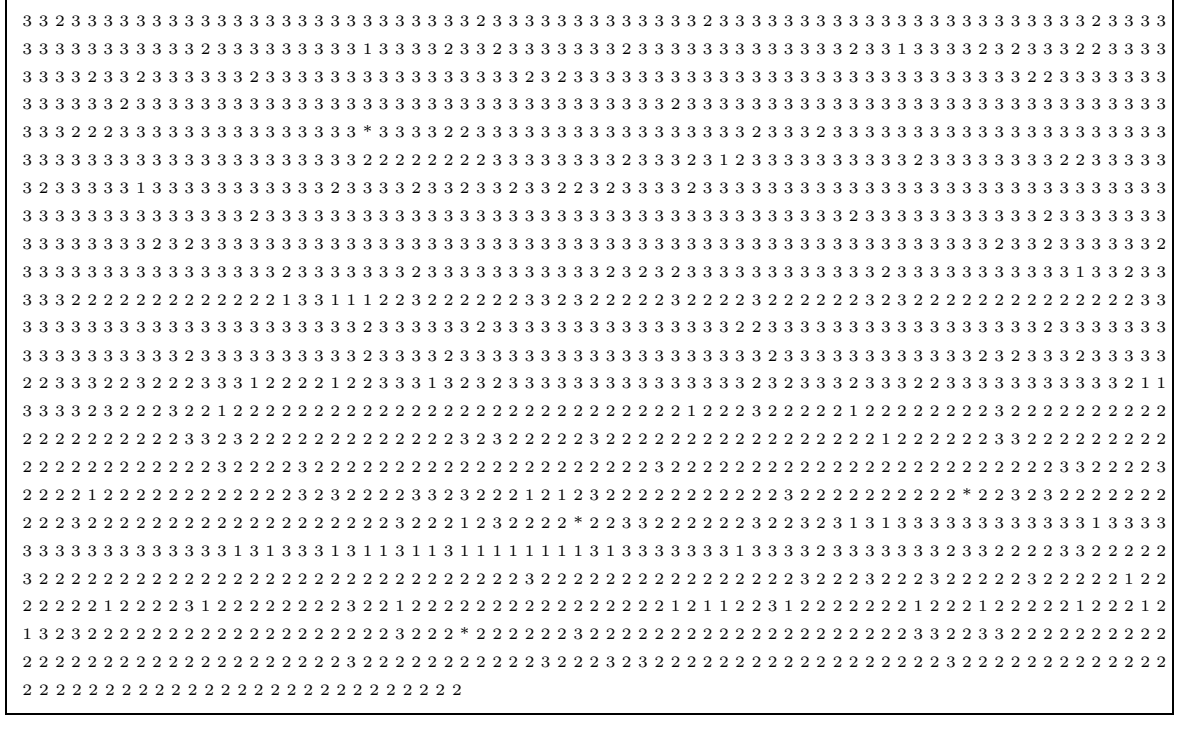

表 4.15: 3ffe:80a0:0:f00a::21(Tu17-BERKOM–NTT-ECL.r01.ipv6.berkom.de)

3ffe:80a0:0:f00a::21(Tu17-BERKOM–NTT-ECL.r01.ipv6.berkom.de)			
wide:	206	(11.9%)	min/avg/max/std-dev = 301.38/324.60/1536.70/41.37
jgn:	1364	(78.8%)	min/avg/max/std-dev = 291.40/303.94/791.22/16.86
qgpop:	160	(9.2%)	min/avg/max/std-dev = 297.28/326.03/446.68/29.63
failed:	1	(0.1%)	

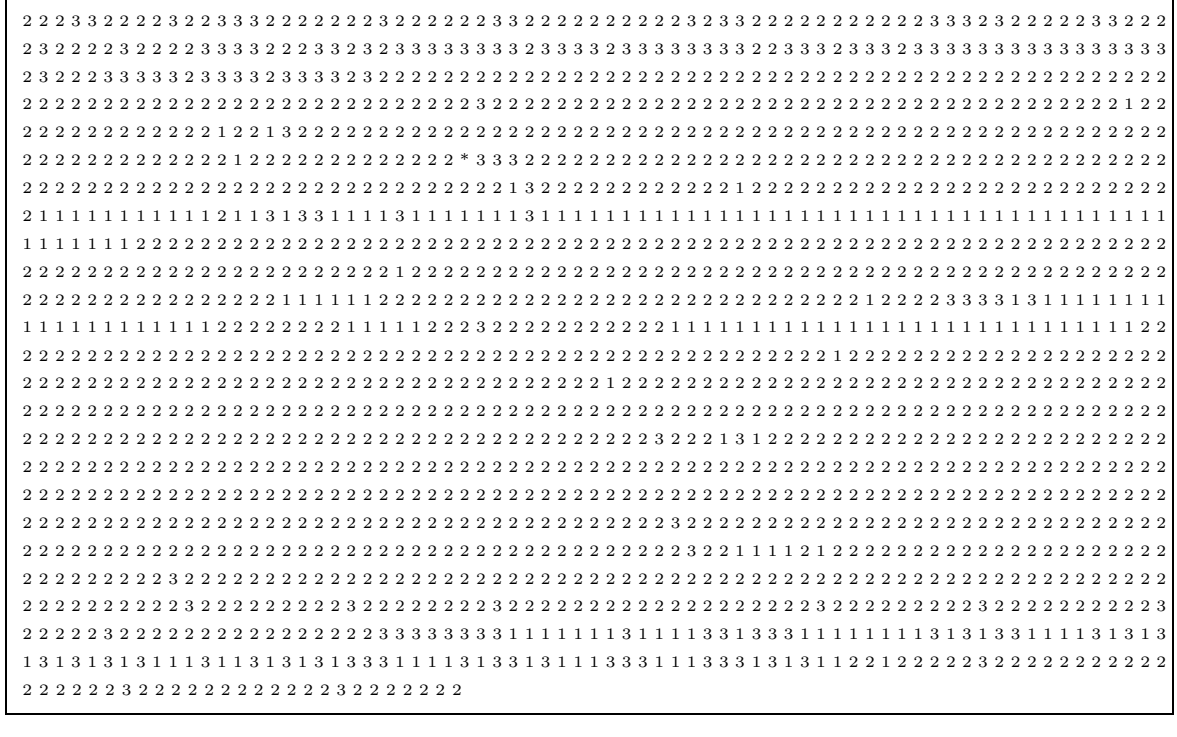

表 4.16: 2001:660:80:421a::2

2001:660:80:421a::2			
wide:	0	(0.0%)	min/avg/max/std-dev = 0.00/0.00/0.00/0.00
jgn:	0	(0.0%)	min/avg/max/std-dev = 0.00/0.00/0.00/0.00
qgpop:	0	(0.0%)	min/avg/max/std-dev = 0.00/0.00/0.00/0.00
failed:	1731	(100.0%)	

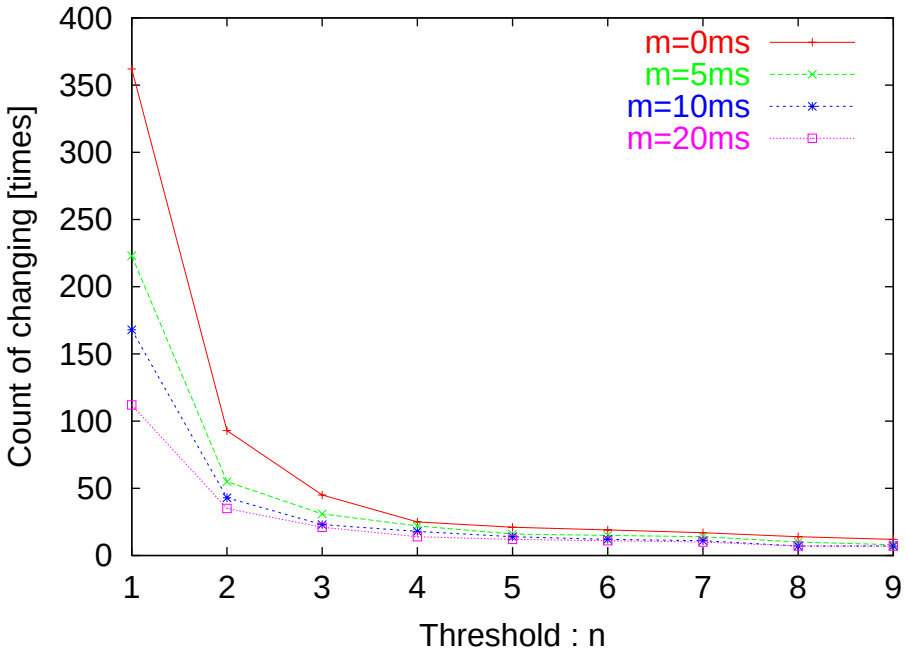

図 4.1: 3ffe:3900:3::2(korea-6tapsl.6tap.net) における切り替え回数

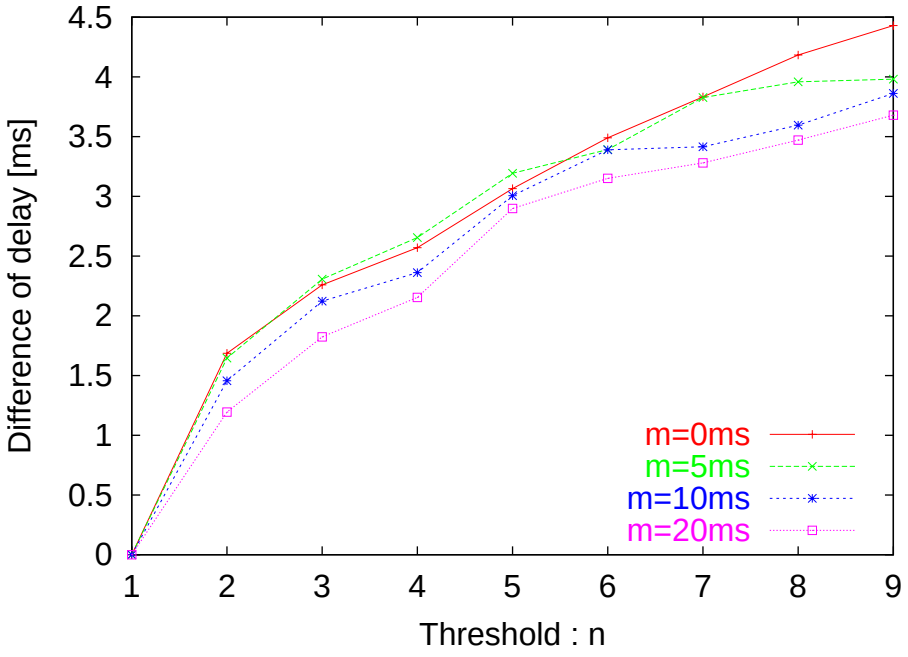

図 4.2: 3ffe:3900:3::2(korea-6tapsl.6tap.net) における最小遅延との差

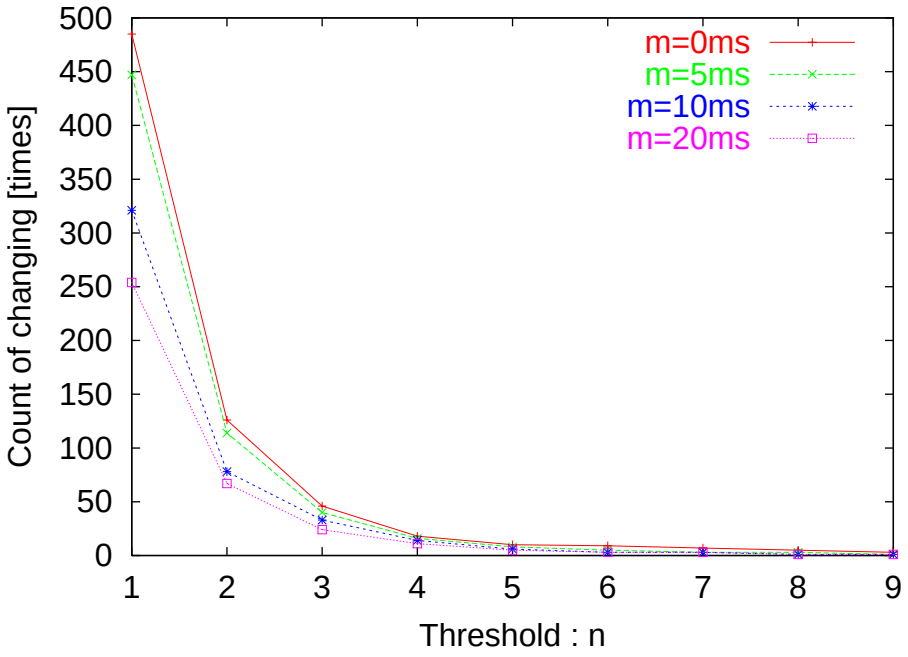

図 4.3: 3ffe:3200::1(cernet.ipv6.net.edu.cn) における切り替え回数

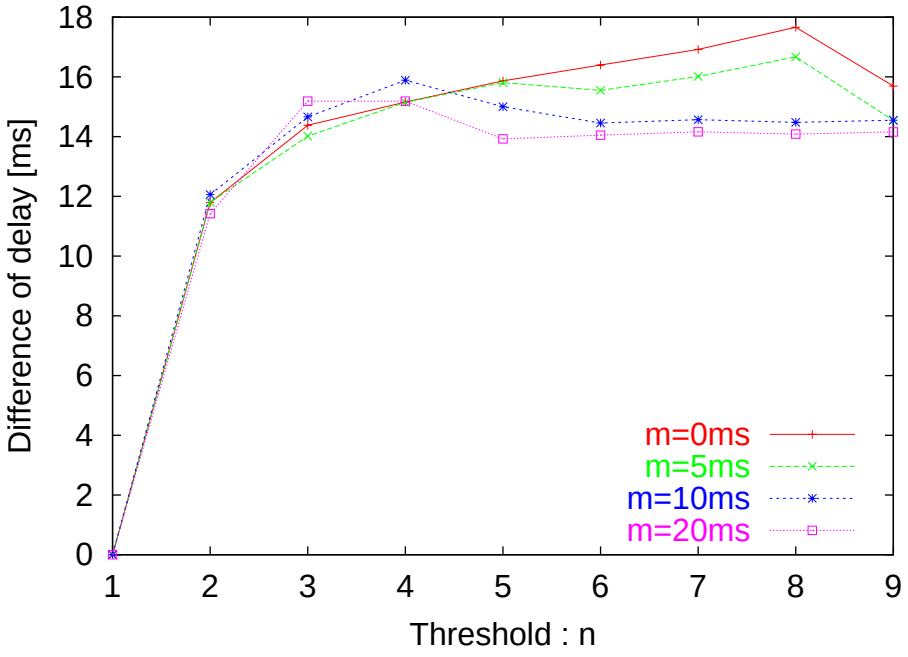

図 4.4: 3ffe:3200::1(cernet.ipv6.net.edu.cn) における最小遅延との差

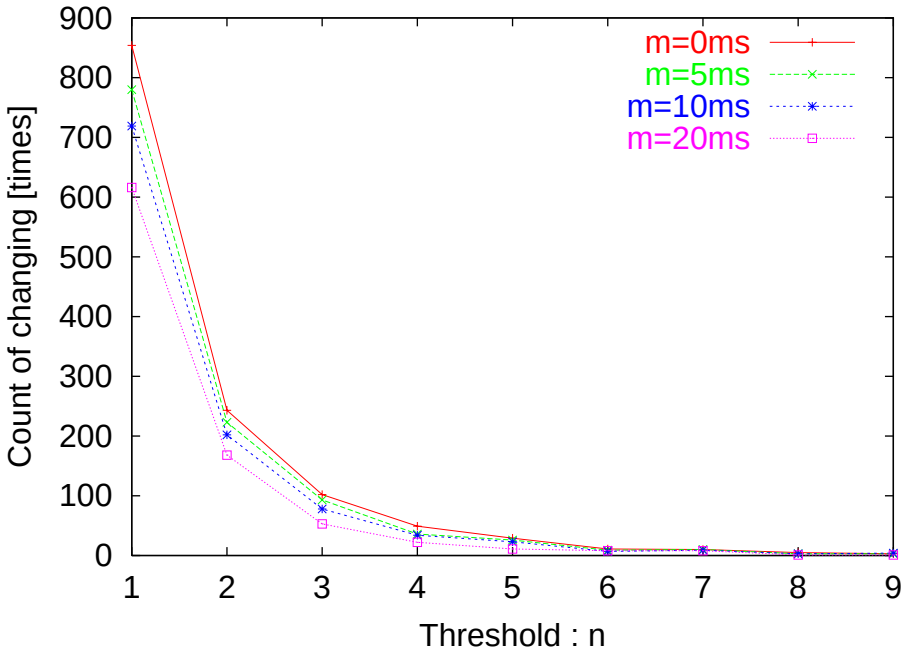

図 4.5: 3ffe:b80:2:1a2::2(cnr-coe-psu-th.tsps1.freenet6.net) における切り替え回数

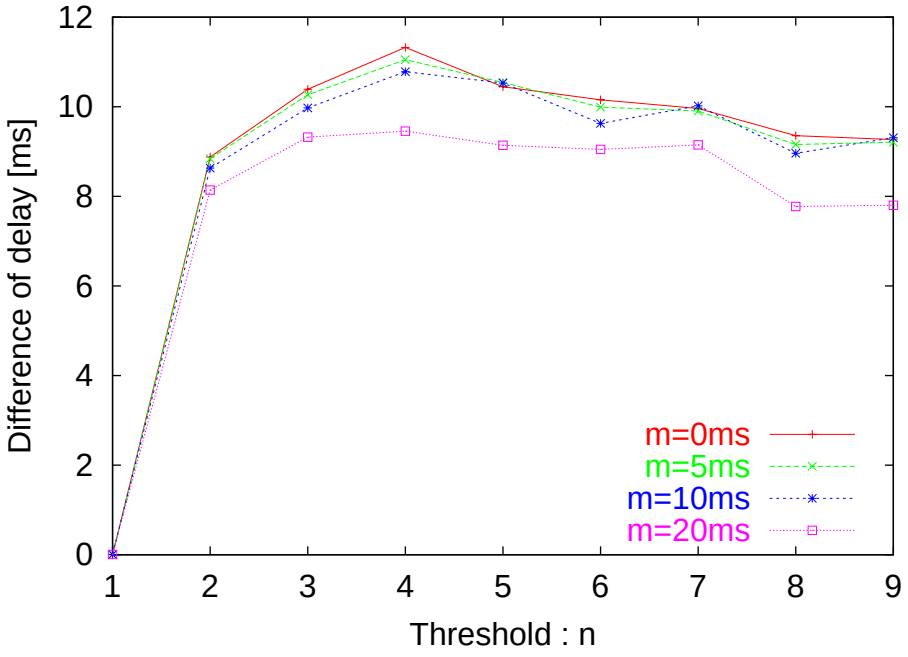

図 4.6: 3ffe:b80:2:1a2::2(cnr-coe-psu-th.tsps1.freenet6.net) における最小遅延との差

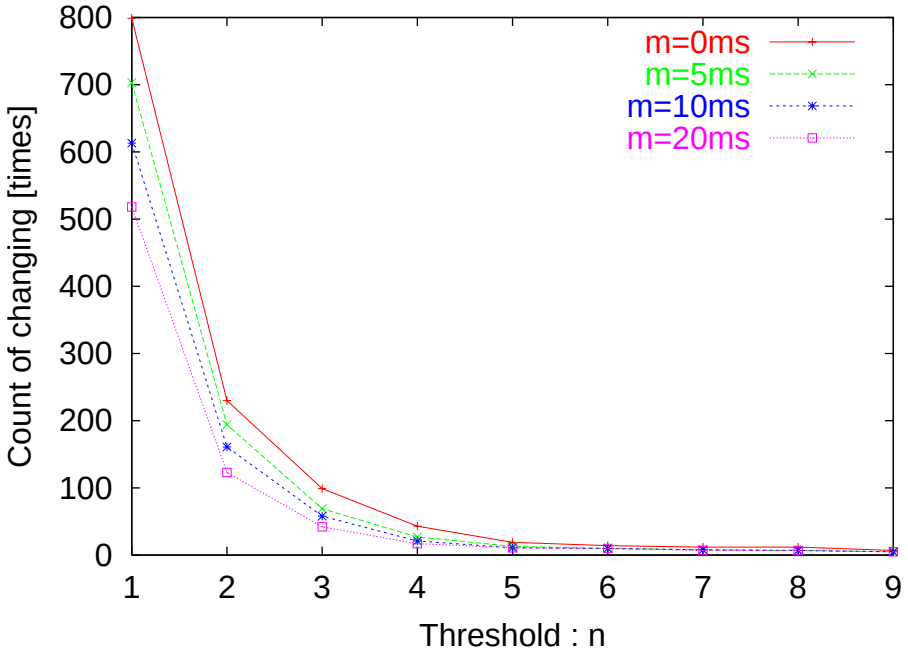

図 4.7: 3ffe:80d0:fffc:10::43 における切り替え回数

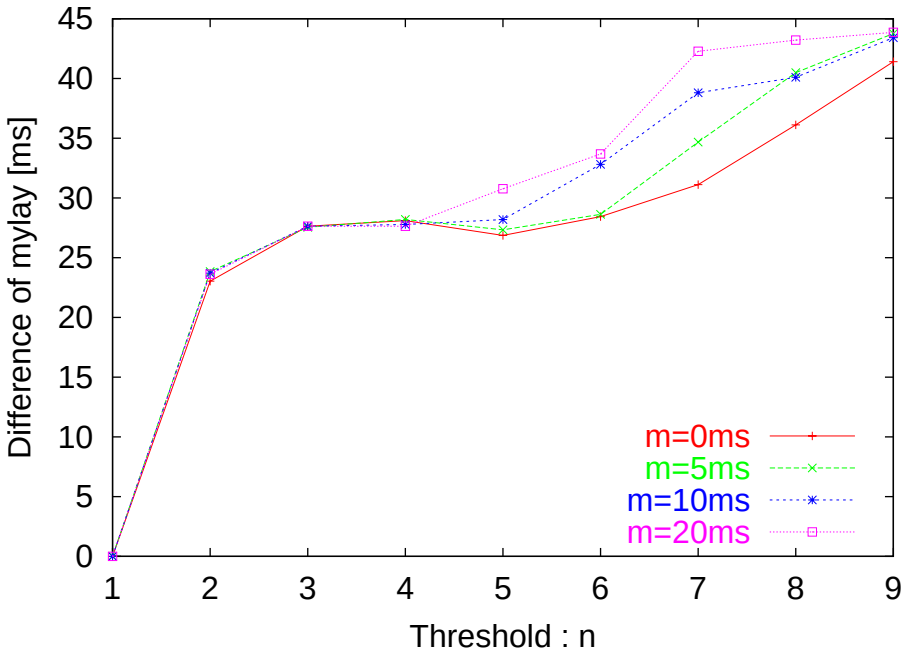

図 4.8: 3ffe:80d0:fffc:10::43 における最小遅延との差

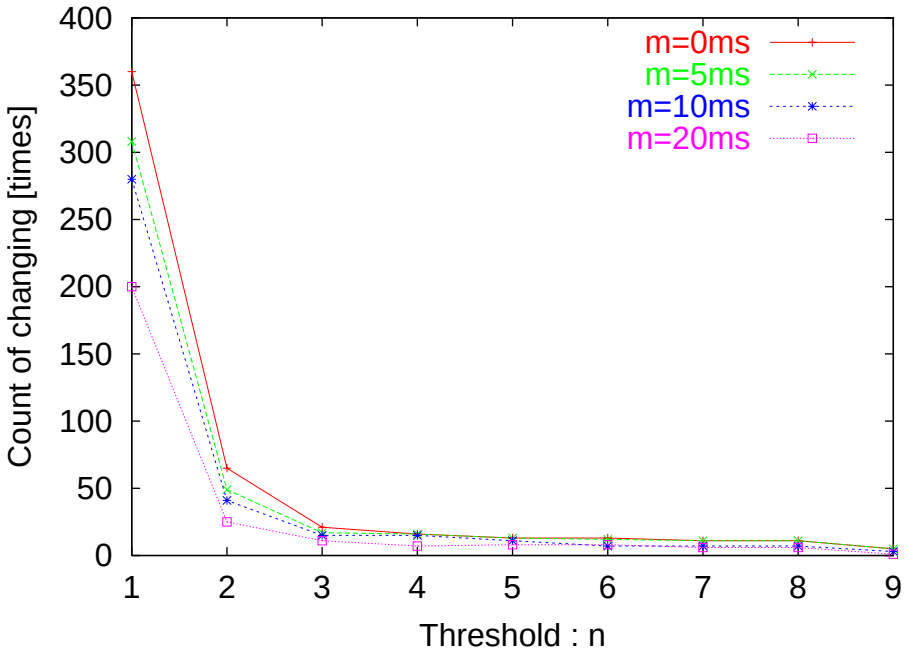

図 4.9: 2001:630:0:1::1e(soton.tun.ipv6.ja.net) における切り替え回数

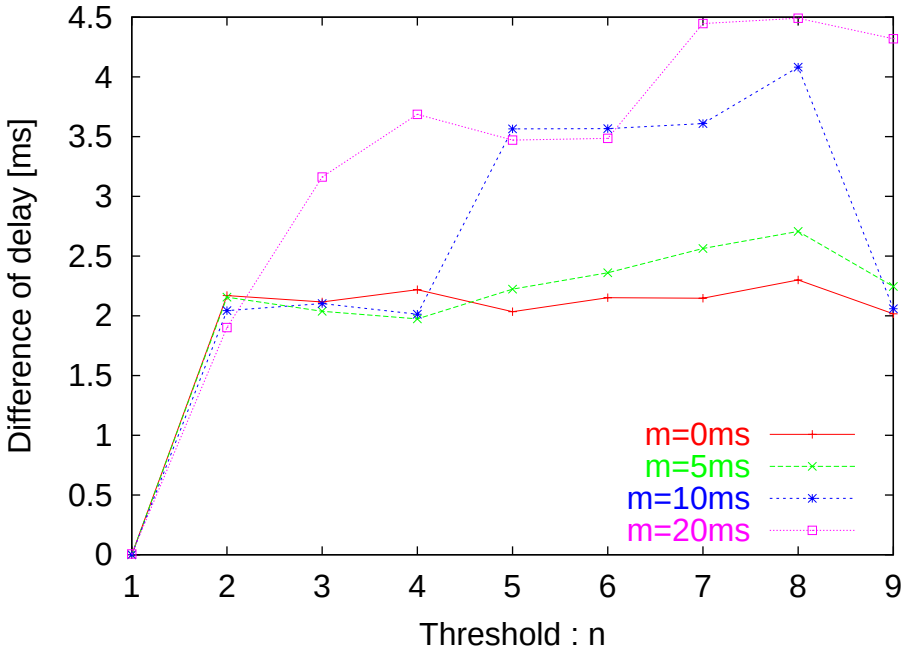

図 4.10: 2001:630:0:1::1e(soton.tun.ipv6.ja.net) における最小遅延との差

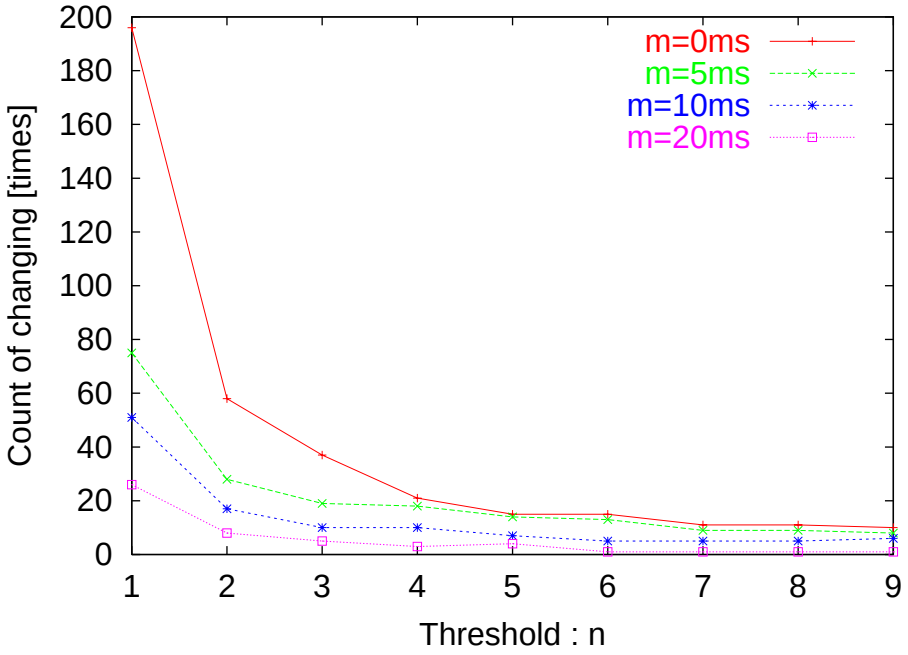

図 4.11: 3ffe:80a0:0:f00a::21(Tu17-BERKOM–NTT-ECL.r01.ipv6.berkom.de) における切り替え回数

4.3 実環境での評価

提案機構の有効性を検証するために，実際のインターネット環境において実験を行った．

図 2，図 3 の結果より，切り替えの回数が少なく，かつ遅延が最小となる回線との差の劣化が少ない n，m として，n = 3，m = 10 を採用し，本機構を使用した場合とそうでない場合で，以下の Web サーバからファイルを取得したときの平均スループットを求めた．

A www6.6b0ne.hu

B www.ipv6.unam.mx

C www.join.uni-muenster.de

ファイルの取得には wget を用い、１０分ごとに１度、１度あたり３回ファイルを取得した。

その結果すべての場合で，本機構を用いることによって高いスループットを得られることがわかった（表 1 参照）．

この結果より，最適な回線を選択していることがわかった．

5　まとめと今後の課題

電気や電話と比較すると不安定であるインターネットの利用する通信システムの、可用性を向上させるための技術としてマルチホーミング技術が注目されている。

本研究では効率のよいマルチホーム環境実現のために次世代の基幹プロトコル IPv6 の特徴を利用して，始点アドレスのネットワーク ID（プレフィクス）を回線の状態に応じて書き換える機構を提案した。また提案機構を導入することによる効果を評価した。

その結果、以下のようなことが明らかとなった。

- 本機構を導入して上流回線の選択とプレフィクス変換を行うことで，往復の経路を適切に制御できる。
- 通信の際に、適切な上流回線を往復の経路とすることで、良好なスループットを得られる．
- マルチホーム環境を実現することにより，通信システムの可用性を高められる．
- 往復遅延は、経路選択のために用いる手法として適切である。しかし今後は手法の分散化など、さらなる展開が期待できる。
- プレフィクス切り替えを判断する際に用いたパラメータ n，m は通信相手によって異なる事が考えられる．よって今後，ネットワークの状態に応じてパラメータを変更する仕組みなどを導入した場合の検証が必要である．

謝辞

本研究を進める上で、様々な方々に様々な形で支えていただきました。この場を借りてお礼申し上げます。ご指導いただきました尾家 祐二教授に心より感謝いたします。川原 憲治助教授に深く感謝いたします。九州芸術工科大学の堀 良彰 助手池永 全志 助手竹村 真奈美事務補佐員尾家研究室、川原研究室の皆様