

IPv6 multicast における転送トポロジーの探索手法

菊池 豊^{†1} 藤岡 里佳^{†2}

高知工科大学 情報システム工学科^{†1} データ通信システム^{†2}

概要

ネットワークの管理を行う際に、traceroute に代表される始点から終点までの経路を探索するツールが有用である。本稿では IPv6 multicast での配送木を探索する手法を提案し、その実装について紹介する。

A Method of Exploring Distribution Trees of IPv6 Multicast

KIKUCHI Yutaka^{†1} and FUJIOKA Rika^{†2}

Kochi University of Technology^{†1}, Data Communication System Co., Ltd. ^{†2}

Abstract

Tracing tools are useful on network management, which is well known as traceroute command that traces route from source address to destination. We propose a method to explore distribution trees of IPv6 multicast and report an implementation of it.

1 はじめに

ネットワーク管理において経路情報の監視は不可欠である。このような場合に、特定の始点から特定の終点までの経路を探索するツールは大変有用である。このようなツールで代表的なツールに traceroute がある。これは IPv4 の unicast での経路を探索するツールである。この他に IPv6 の unicast 用に traceoute6 があり、また multicast 用には IPv4 用の mtrace や IPv6 用の mtrace6 がある。

multicast の伝送トポロジーは一般に木構造をとるため、特定の始点から特定の終点までの経路を示すだけでなく、特定の始点から配送している全ての経路が探索できると便利である。このようなツールとして IPv4 用には map-mpbone コマンドが

ある。ところがこの機能を持つ IPv6 用のツールはこれまでなかった。

我々は IPv6 の multicast において、配送木全体を表示するツールを開発した。以下では、まず、第 2 節で既存の経路探索ツールについて述べる。次に、第 3 節で目的とするツールの要求仕様と、我々の手法の提案を行う。さらに、第 4 節で提案手法の実装と機能確認の実験について述べる。第 5 節で本手法についての議論を行い、第 6 節でまとめる。

2 既存の経路管理ツール

本節では既存の Multicast 経路管理用のツールを示す。

2.1 IPv4 用のツール

- mtrace
受信ホストから送信ホストへの到達性が把握できる。受信ホストから送信ホストへの経路上のホップのアドレス、パケット数、ルーティングのエラー状況の情報が得られる。
- map-mbone
指定した multicast ルータから到達可能なすべての multicast ルータを把握できる。
- mrinfo
指定したルータの IP アドレスに続き、隣接するルータの IP アドレスと名前、メトリック、スレッシユホールドが表示される。
- pimstat
IPv4 対応の PIM[1][2] におけるデーモンの状態や統計情報を把握できる。

2.2 IPv6 用のツール

- mtrace6
受信ホストから送信ホストへの到達性が把握できる。受信ホストから送信ホストへの経路上のホップのアドレス、ルーティングのエラー状況の情報が得られる。
- pim6stat
IPv6 対応の PIM[1][2] の状態や統計情報を把握できる。multicast インタフェースの状態、隣接 PIM ルータの情報、MLD Querier のリスト、multicast 経路表の情報が表示される。

2.3 IPv6 multicast 配送木探索ツール

以上のツールのうち、経路を把握するツールを表 1 に示す。単独経路を表示するツールは IPv4 にも IPv6 にもある。しかし、配送木を把握するツールは IPv4 用が存在するものの IPv6 には存在しない。

IPv4 multicast において複数経路を表示しているツールがあって、IPv6 multicast にはない理由には、用いられるプロトコルの違いがある。

IPv4 は、グループ管理プロトコル IGMP の ASK NEIGHBORS メッセージを利用できる。このメッセージで隣接する multicast ルータを発見し、それを再帰的に繰り返すことで、multicast 配送の全体像を得ることが可能である。

一方、IPv6 multicast においては、グループ管理プロトコルである MLD にも、multicast 経路プロトコルである PIM にも、上述のメッセージと同様な機能は存在しない。

このため IPv4 の map-mbone の手法を応用して IPv6 用のツールを作ることはできない。

3 IPv6 multicast 経路の探索

転送トポロジを把握することができれば、IPv6 multicast のトラブルシューティングを容易にすると考えられる。

我々は IPv6 multicast における転送トポロジの探索手法を提案し、この手法を “mtopology6” として実装した。本節では、multicast における転送トポロジを探索・図示する手法を述べる。

3.1 要求仕様

提案する手法における要求仕様は以下である。

- 1 つのホスト上に実装するのみで実行可能
全ての関連する multicast ルータに新しい実装を持ち込むことは現実的でない。
- 既存のプロトコルを用いて探索可能
新たなプロトコルを考案して、その実装をルータに行うことはやはり現実的ではない。
- 転送トポロジを Tree 構造に図示
全体の配送木は配送元からの木構造を持つので、それを分かりやすく視覚化する。

IP ver.	表示内容	
	単独経路	配送木
IPv4	mtrace	map-bone
IPv6	mtrace6	既存なし

表 1: Multicast 経路探索ツール

3.2 実現方法

本研究で採用した手法を直観的に説明すると、ICMP [3] echo メッセージ、traceroute のアルゴリズム、そして mtrace6 のアルゴリズムの合成である。経路探索のおおまかな流れは次のようになる。

1. 受信ホストの探索
2. 各受信ホストへの最終ホップルータの探索
3. 最終ホップルータから送信ホストへの multicast 経路の探索
4. 各経路情報の集約・結果表示

なお、コマンドを実行するホストは送信ホストに限定した。また、その送信ホストは、検索する転送トポロジーの multicast アドレスに対しての経路が設定されているとする。

以下では各ステップについて詳細を述べる。

3.2.1 受信ホストの探索

ICMP echo request メッセージを multicast グループ宛てに送信する。request メッセージは PIM によってルータからルータへと転送され、受信ホストへと届けられる (図 1)。これを受信したホストは、echo request を送信したホストへ向けて echo reply を unicast で送出する。これによって、受信ホストの IP アドレスを取得することができる。

3.2.2 最終ホップルータの探索

次に、上で発見した各受信ホストから送信ホストへの multicast 経路を取得したい。これには次節に示すように MLD mtrace メッセージを利用できる [4]。ところが、MLD mtrace メッセージは受信ホストではなく最終ホップルータで処理されるため、送信ホストから受信ホストへの multicast 経路上での最終ホップルータを探す必要がある。

送信ホストから受信ホストへの multicast 経路上での最終ホップルータを見つけるために、以下の traceroute のアルゴリズムを援用する (図 2)。

使用していないと思われるポート番号を選んで UDP メッセージを各受信ホスト宛てに送信する。パ

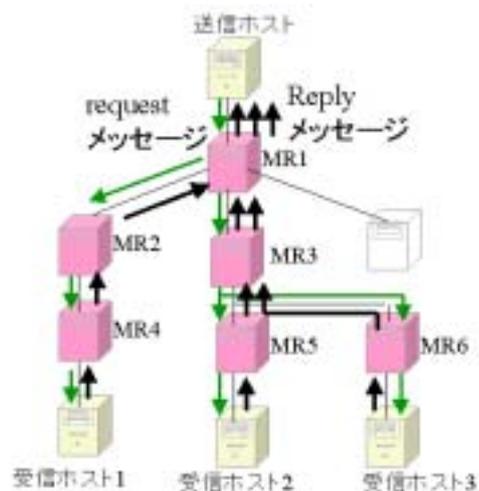


図 1: 受信ホストの探索

ケット伝送の経路途中で hop limit¹が 0 になったら、ルータは ICMP の Time Exceeded for a Datagram メッセージを返送する。受信ホストが受け取った場合には、ホストは ICMP の Destination-Unreachable タイプの「ポート到達不可」メッセージを返送する。これにより、hop limit を 1 から順に大きくしながら ICMP のエラーメッセージを受け取ることで、経路全てのルータを探索できる。

3.2.3 最終ホップルータから送信ホストまでの multicast 経路の取得

各最終ホップルータから送信ホストへの multicast 経路は MLD の mtrace メッセージによって取得可能である。

上で得た各最終ホップルータ宛てに mtrace query メッセージを送信する。それに対する response メッセージを受信することで、multicast グループのすべての受信者から送信ホストへの multicast 経路情報を取得する (図 3)。

3.3 経路情報の集約と結果表示

前節までの手順で得た各 multicast 経路は、受信ホストから送信ホストへの順になっているので、これを逆の順に並べ替える。次に最初の経路につ

¹IPv4 における TTL に相当する。

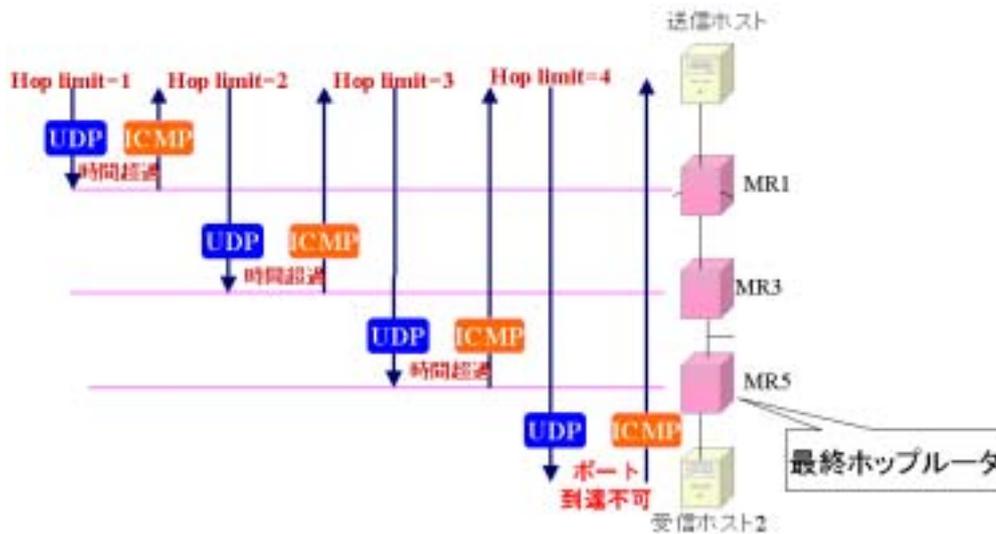


図 2: traceroute アルゴリズム

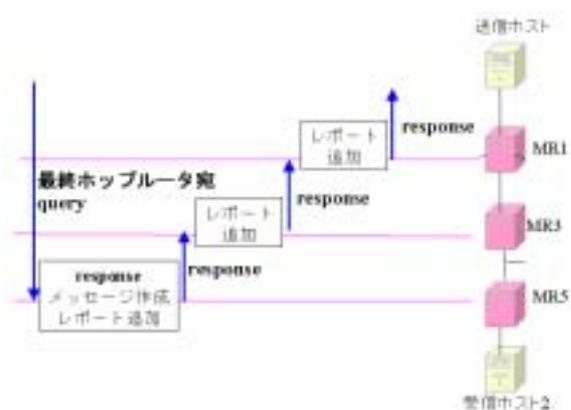


図 3: mtrace メッセージの動作

いて幹を生成し、他の経路情報について生成した木と比較し、新しい子や兄弟を追加していく。これによって得られた木を結果として出力する。

3.4 メッセージ送出手間調整

作成中における数十回の実地試験により、その他のパケットの送受信の後に一定時間間隔を空けて行わなければ、MLD mtrace メッセージ受信に失敗するという結果を得た。そして、実際に 20 回ほど作成した本手法のツールを用いて試したところ、4 分間程度の間隔を空けると MLD mtrace メッ

セージの受信に失敗する可能性は低いという結果を得ている。

この原因は究明できていないものの、実装はこれを考慮してある。

4 実装と実験

前節の手法を FreeBSD4.4-RELEASE で実装した。C 言語でおよそ 1900 行ほどのプログラムである。このプログラムを用いて実験を行い、機能の確認を行った。

実験は図 4 に示すネットワークで行った。全ホス

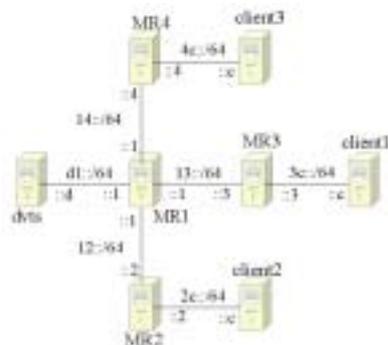


図 4: 実験ネットワーク

トに FreeBSD4.4-release と pim6sd をインストール

し、pim6sd.confの設定により Designated Router (DR) や Rendezvous Point (RP) を設定した。実験は DR, RP, client を表 2 に示すように設定し、コマンドの出力がどのようになるかを調べた。結果を図 5～図 8 に示す。

実験	DR	RP	受信ホスト	結果
実験 1	dvts	mr1	c1, c2, c3	図 5
実験 2	dvts	mr1	c1, c2	図 6
実験 3	dvts	mr3	c1, c2, c3	図 7
実験 4	dvts	mr3	c1, c3	図 8

表 2: 実験条件

```

Multicast Delivering Topology
-----
01---02
   |-03
   |-04

Address measurement table [No.]Address
-----
[01] d1::1
[02] 3c::3
[03] 4c::4
[04] 2c::2

```

図 5: 実験 1 結果

```

Multicast Delivering Topology
-----
01---02
   |-03

Address measurement table [No.]Address
-----
[01] d1::1
[02] 4c::4
[03] 2c::2

```

図 6: 実験 2 結果

```

Multicast Delivering Topology
-----
01---02---03
   |-04

Address measurement table [No.]Address
-----
[01] 3c::3
[02] d1::1
[03] 4c::4
[04] 2c::2

```

図 7: 実験 3 結果

```

Multicast Delivering Topology
-----
01---02---03

Address measurement table [No.]Address
-----
[01] 3c::3
[02] d1::1
[03] 4c::4

```

図 8: 実験 4 結果

- MLD において用いることのできるメッセージタイプが mtrace query と mtrace response メッセージに限定されている。
- PIM において用いることのできるメッセージタイプは Hello メッセージであるものの、Hello メッセージに対しての応答は発生しない。
- マルチキャストアドレスに対する通信で ICMP エラーメッセージは発生しない。

以降の本節では提案した以外に考案した探索手法と、その実現可能性について述べる。

5.1 ICMP ルータ要請・広告 + IPv6 始点経路制御

これは以下の 2 つの行程を繰り返すことで、受信ホスト方向へ multicast パケットが到達可能な経路を探索する案である。

5 議論

IPv6 multicast での配送木の探索は以下の理由で難しい。

- multicast ルータは直接接続したリンク上に「受信ホストが存在するか否か」の情報しか持っていない。
- 送信側から受信側への multicast パケットの流れをたどることはできない。

- 直接接続しているリンク上に ICMP のルータ要請を multicast アドレス宛に送信し、これに対して送られてくるルータ広告メッセージによって、multicast ルータを発見する。
- 次に、IPv6 拡張ヘッダの始点経路制御の途中経路ホストとして、上で発見したルータを

追加した形のルータ近隣要請メッセージを、multicast アドレス宛に送信する。

この案はうまく機能しない。まず、multicast に対して ICMP パケットを送信しても、multicast でのルーティングによって受信ホストまでパケットが転送され、途中のルータでは受け取られない。また、IPv6 拡張ヘッダの始点経路制御において、指定する途中経路のアドレスにも最終的な宛先アドレスにも multicast アドレスは使用できない。

5.2 ICMP 近隣要請・広告 + IPv6 始点経路制御

これは、上記で述べた案と類似しており、ICMP ルータ要請・ルータ広告メッセージの代わりに、ICMP 近隣要請・近隣広告を用いる案である。

- multicast アドレスに向けて ICMP 近隣要請メッセージを直接接続しているリンク上に送信する。これに対して送られてくる近隣広告メッセージを受信することによって、multicast パケットを受信または、転送しているホストを発見する。
- 次に、IPv6 拡張ヘッダの始点経路制御の途中経路ホストとして、上で発見したルータを追加した形の近隣要請メッセージを、multicast アドレス宛に送信する。

この案も前節同様の理由でうまく動作しない。

5.3 hop limit 付き multicast パケット

これは Traceroute 同様に hop limit を 1 ホップずつ増加させながら、ICMP 時間超過およびポート到達不可メッセージを受信する案である。

これもうまく働かない。multicast パケットに対して ICMP によるエラーメッセージは発生しないからである。従って、ICMP の時間超過メッセージや、ICMP ポート到達不可メッセージは送られてこないため使用できない。

我々の手法では unicast を用いることで解決した。このため、unicast と multicast とで最終ホップルータが同一である場合にのみ正確に表示するという制限がついた。

6 まとめ

既存の multicast 管理ツールを分類し、IPv6 multicast において配送木を表示するツールが存在しないことを示した。また配送木を探索・表示する手法を提案し、その実装を行い実験により効果を確認した。

この手法では、unicast における最終ホップルータと multicast における最終ホップルータとが同一である場合にのみ正確に機能すること、ツールが送信ホストのみで実行が可能であるという制限が生じた。

今後の課題として、PIM-DM における実験や PIM-SM の共有ツリーと送信元ツリーが混在する状況における実験を行う必要がある。また、MLD mtrace メッセージの動作について、FreeBSD の実装がプロトコル仕様を満たしているのか検証する必要があると考える

参考文献

- [1] Andrew Adams, Jonathan Nicholas, and William Siadak. Protocol independent multicast - dense mode (PIM-DM): Protocol specification (revised), November 2001. draft-ietf-pim-dm-new-v2-00.txt.
- [2] D. Estrin, et al. Protocol independent multicast - sparse mode (PIM-SM): Protocol specification, June 1999. RFC2362.
- [3] A. Conta and S. Deering. Internet control message protocol (ICMPv6) for the internet protocol version 6 (IPv6) specification, December 1998. RFC2463.
- [4] S. Deering, W. Fenner, B. Haberman, et al. Multicast listener discovery (MLD) for IPv6, October 1999. RFC2710.