

Zebra ospf6d の設計と実装

小原 泰弘¹ 中村 修² 村井 純²

慶應義塾大学大学院政策・メディア研究科¹ 慶應義塾大学環境情報学部²

次世代のインターネットを良いものにするために、IPv6 ネットワークの運用経験と研究開発が必要である。これを実現するためには、多くの人が IPv6 ネットワークを構築できるツールが必要となる。本論文では、IP ネットワークを構築するために最も重要となる経路制御ツール、IPv6 をサポートした無償のオープンソース OSPF 実装を実現し、評価した。Zebra ospf6d は、世界中で利用される、実用に堪えるソフトウェアであることが確認できた。

Design and Implementation of Zebra ospf6d

Yasuhiro Ohara¹, Osamu Nakamura², Jun Murai².

¹Graduate School of Media and Governance, Keio University

²Faculty of Environmental Information, Keio University

To gain operational experience on IPv6 and make next generation Internet better, much research and development on IPv6 network is mandatory, and a tool for constructing IPv6 network is required to achieve this goal. We implemented and evaluated a free and open source OSPF software, which supports IPv6. This software, Zebra ospf6d, is proven to be a feasible software through the evaluation.

1 はじめに

インターネットルーティングは、インターネットを介した通信を実現するために必要不可欠なものである。ルーティングが正しく動作しないと、インターネット通信のデータ単位となる IP パケットが通信の終点ホストに到達できないため、どのような種類のインターネット通信も行うことができない。インターネットが社会基盤となっている今日では、インターネットルーティングは重要な研究トピックである。

インターネットは、その基盤となる通信プロトコルを IPv4[1] から IPv6[2] へ移行しつつある。インターネットユーザの大半は IPv4 を利用しているが、将来的には IPv6 へ移行する見通しである。IPv6 は、IPv4 ではサポートできない程の、より巨大なインターネットをサポートするために開発

されたものである。そのため、IPv6 におけるルーティングには IPv4 ルーティングに要求されるものに比べて高いスケーラビリティが要求される。

また、IPv4/IPv6 に関わらないインターネットルーティングそれ自体も開発が進んでいる。インターネットを構成するそれぞれのネットワークの内部経路制御 (Interior Routing) には OSPF[3] を利用するのが一般的である。インターネットルーティングの開発の主要な例には、この OSPF を利用してトラフィック制御 (Traffic Engineering, TE) を実現するという新しい機能の開発が挙げられる。

次世代のインターネットは、IPv6 を利用した、より大規模なネットワークとなる。またそこでのルーティングも、経路制御時に QoS (Quality of Service) を考慮できるなど、現在に比べいくつかの重要な新機能をサポートする必要があるだろう。

このような世界を実現するためには、早くから IPv6 ネットワークを実現し、その上で研究開発や

^{1,2}Keio University Shonan Fujisawa Campus
5322, Endo, Fujisawa, Kanagawa 252, Japan
E-Mail: yasu@sfc.wide.ad.jp

実用上の問題点を発見する必要がある。そして、そのためには多くの人々が IPv6 ネットワークを構築できるツールが必要となる。IP ネットワークを構築するための最も重要なツールは、ルーティングを実現するためのソフトウェアである。さらに、多くの人々が IPv6 ネットワークのうえで研究開発するためには、無償の、ソースの公開されたルーティングソフトウェアが必要不可欠である。

著者は、次世代インターネットの実現と、それに向けた研究開発を考慮し、IPv6 をサポートした無償のオープンソース OSPF を実装した。以下これを `ospf6d` と呼ぶ。この実装は近年数年に渡って、GNU Zebra ルーティングソフトウェアパッケージ [4] に含められ、世界中で利用されている。本論文では、この実装に関する設計思想および実装の詳細を記述する。また、現状の利用状況と SPF 計算が消費する時間を示すことにより、実装の評価とする。

2 設計

インターネットルーティングは、一般的な経路計算問題の上に状態の変化や有限な資源の効率の利用を含めた、複雑かつ困難なトピックである。インターネットルーティングを実現するソフトウェアでは、さらにこの上に、「使いやすさ」などの実用上の問題点を考慮しなくてはならない。本研究では、OSPF for IPv6 [5] の仕様を実装する上で、実用性、拡張性、性能を(この優先順位で)考慮した。

1. 実用性

本実装の主な目的は「IPv6 ネットワークを構築するためのツールとして利用できること」であったため、実用性を第一に考えて実装した。そのため、新しい機能を実現することより、既存の機能を確実に実行できるソフトウェアを目標とした。また、プログラムコードのバグは実用性の低下に直結するため、バグの修正は勿論のこと、バグを発見しやすい整理されたコードを作成することに主眼を置いた。具体的には、同じ処理を実行する冗長なコードを省き、コードをできるだけ再利用するようにし、プログラム全体の長さを短く保つよう注意した。また、性能の観点から好ましくないものでも、実用性が向上すると見込まれたものは採用した。例えば、

プロトコル処理の詳細なログは、プロトコル実行の性能を低下するが、実用性を向上させる。

2. 拡張性

OSPF や IPv6 は、これからも広く利用され、多くの拡張が取り込まれることが予想される重要な技術である。そのため、拡張性を考慮して実装の内部構造を設計した。具体的には、OSPF プロトコルの拡張性を利用して将来新しい機能を実現することを念頭において、各機能をモジュールに分けた。例えば、OSPF の仕様は、新しい LSA を作成し、それを理解する一部の OSPF ルータのみが特定の処理を実行するといった、段階的なシステムの機能拡張を許している。これを本実装上で実現する場合、プログラムコード中の分散した多くの場所を変更するより、新しいモジュールを一つ追加するだけであることが望ましい。

3. 性能

性能に関する OSPF の一般的な議論は、SPF 計算に関するものと LSDB 検索に関するものである。OSPF の計算量は一般的に、SPF 計算時の候補リスト中の最小コストのノードを発見する計算量に比例する、と言われる。本実装では候補リストにヒープソートを利用したプライオリティキューを採用し、一般的な実装と同様に $O(N \log(N))$ の計算量が見込める (N はルータとマルチアクセスネットワークをノードとした際のノード数)。

実際には、SPF 計算よりも LSDB 検索の方が実行される回数が多く、SPF 計算自体が LSDB 検索を実行するため、LSDB 検索に利用されるアルゴリズムも重要である。本実装では、LSA の LS Type、Link State ID、Advertising Router の三つのフィールドを連結したものをキーとし、経路検索と同様のパトリシアツリーを利用して LSDB を構築した。そのため、LSDB 中の LSA の数を N としたとき、LSDB の検索にかかる時間は $O(\log(N))$ であり、キーのビット数を w とすると最悪でも $O(w)$ の計算時間内に検索が終了する。

表 1: ospf6d モジュールの説明

モジュール名	説明	モジュール名	説明
ospf6_abr	エリア境界ルータ機能	ospf6_area	area 構造体とその操作関数群
ospf6_asbr	AS 境界ルータ機能	ospf6_flood	LSA flooding 機能
ospf6_interface	interface 構造体とその操作関数群	ospf6_intra	エリア内経路計算機能
ospf6_lsa	LSA 操作関数群	ospf6_lsdb	LSDB 構造体とその操作関数群
ospf6_main	main 関数とシグナルハンドラ	ospf6_message	メッセージ操作関数群
ospf6_neighbor	neighbor 構造体とその操作関数群	ospf6_network	パケット送受信関数群
ospf6_proto	プロトコルデータの共通操作関数群	ospf6_route	経路、経路表とその操作関数群
ospf6_spf	SPF 計算機能	ospf6_top	top 構造体とその操作関数群
ospf6_zebra	zebra への送受信機能	ospf6d	ospf6d 全体に関する機能

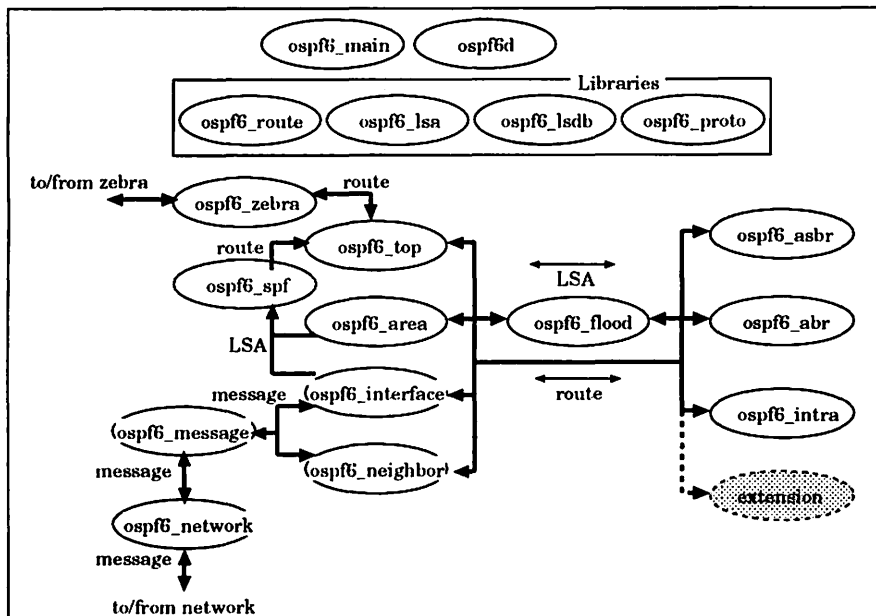


図 1: ospf6d 内部構造

表 2: LSA と処理モジュール

LSA の種類	生成モジュール	計算モジュール
Router-LSA	ospf6_intra	ospf6_spf
Network-LSA	ospf6_intra	ospf6_spf
Inter-Area-Router-LSA	ospf6_abr	ospf6_abr
Inter-Area-Prefix-LSA	ospf6_abr	ospf6_abr
AS-External-LSA	ospf6_asbr	ospf6_asbr
Intra-Area-Prefix-LSA	ospf6_intra	ospf6_intra
Link-LSA	ospf6_intra	ospf6_spf

表 1 に、各モジュールの説明を示す。図 1 に、Zebra ospf6d の内部構造を示す。モジュールは 18 個存在し、図に示すデータの交換が行われる。データは LSA、経路、プロトコルメッセージの三種類あり、データ交換を示す矢印のそばにどのデータが交換されるかが示されている。

ospf6_main モジュールは main 関数とシグナルハンドラを、ospf6d モジュールは ospf6d プロセス全体に関わる関数を含んでいる。ospf6_route、ospf6_lsa、ospf6_lsdb、ospf6_proto の各モジュールはライブラリのように用いられ、その他のモジュールから多く利用される。

OSPF プロトコルに利用される ospf6d の外部との相互作用は、zebra デーモンとの経路やインターフェイス情報の送受信 (図 1 中左) と、Unix カーネルの socket を利用したメッセージの送受信 (図 1 中左下) である。

LSA の生成と計算は、ospf6_asbr、ospf6_abr、ospf6_intra の各モジュールで実行される。サポートされる LSA の種類と、生成、計算を行うモジュールを表 2 に示す。LSA は基本的に同一のモジュールによって生成、計算されるが、Router-LSA、Network-LSA、Link-LSA の計算は例外的に ospf6_spf で実行される。

LSA の生成/計算と、LSA の伝播や生存期間の確認 (aging) を分離することによって、拡張性が向上する。新しい LSA の定義とその生成/計算の操作を追加する場合は、その LSA を取り扱う新しいモジュールを追加する。この概念の例を、図 1 中右下、斜線の extension モジュールに示した。

OSPF for IPv6 の仕様は複雑なものであり、モジュール間で交換されるデータや各モジュールが持つ状態は多数存在する。そのため、より詳細なデータフローや状態遷移は、簡潔性と可読性のため割愛する。

3 実装

ここでは、Zebra ospf6d の実装の中核を占める、経路表およびそれに付随するフックを説明する。ここで、経路表とは、ospf6d 内部に持つ OSPFv3 経路のみを保持するためのデータ構造を指す。ospf6d のなかでは、area 構造体や top 構造体が、それぞれ異なる目的の複数の経路表を持つ。

経路表は、パトリシアツリーと双方向連結リストを組み合わせたデータ構造を持つ。パトリシアツリーは、もともと Zebra パッケージに付随していた lib/table.c モジュールを利用した。このモジュールは、双方向連結リストもサポートしているが、一つのキーに対する複数エントリの保持をサポートしていない。つまり、同一キーで追加すると、新しいエントリが古いエントリを書き換えてしまうことになる。OSPF プロトコルにおいては、同一の終点に対する経路が複数発見できる場合も多いため、このモジュールを拡張する形で ospf6_route モジュールを作成した。拡張後の経路表概念を図 2 に示す。

図 2 中では、経路 (Ra) と経路 (Rb) は、同じ終点への経路となっている。経路はその計算結果により (例えばエリア内経路かエリア間経路か)、優先順位が決定される。ある終点への経路のなかで、最も優先される経路が一つだけ、システムの

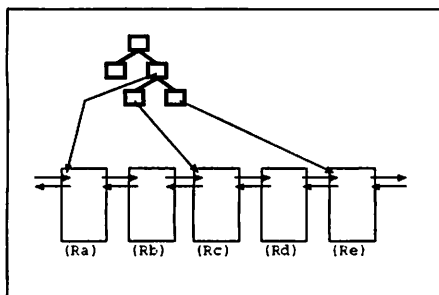


図 2: 経路表

経路表にインストールされる。この同一終点への複数経路の間の順序整列は、ospf6_route_add () 関数のなかで、挿入ソートとして実現される。

経路の追加や削除は、多くの場合他の機能呼び出さなくてはならない。例えば、SPF 計算によって計算されたエリア内経路が追加されると、それは他のエリア (もしあれば) に伝播する必要がある (つまり、Inter-Area-Prefix-LSA として広告しなければならない)。これらの機能呼び出しは、経路表のフックの形で実現した。経路表のフックの概念を、図 3 に示す。

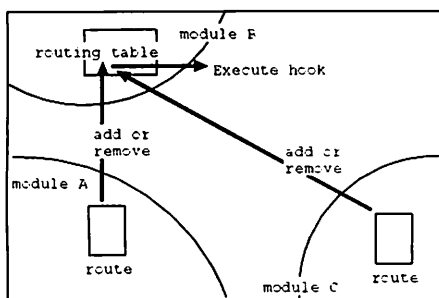


図 3: 経路表のフック

モジュール A とモジュール C は、モジュール B の経路表に経路を追加または削除する。経路追加後の処理は、モジュール B の経路表に設定されたフック関数を呼び出すことにより実行される。この方法では、経路の追加または削除を実行する側にいちいち処理を書く必要がなく (つまり、モジュール A 内部とモジュール C 内部にフック関数を呼び出す処理を追加する必要がない)、冗長なプログラムコードを減らすことができる。

4 評価

4.1 利用状況

ospf6d は、IPv6 ネットワークを構築するためのツールである。そのため、ospf6d が実際に利用されているかどうかの評価の基準となる。また、実際に運用し問題が起きていないことから、実運用されていることは動作確認を兼ねる。

ospf6d は、世界中で利用されており、他のルーティングベンダの OSPFv3 実装とも相互運用されている。ospf6d が利用されている例として、三組織を以下に示す。

- OCCAID (Open Contributor's Corporation for Advanced Internet Development)
OCCAID[6] は、北米を横断するトンネルベースの仮想 IPv6 ネットワークであり、ルーティングプロトコルの実験やネットワーク管理者の教育を目的とした、オープンネットワークである。執筆時点 (2004/08/16) で OCCAID は 25 台の IPv6 ルータを運用しており、そのうち 12 台が Zebra ospf6d を利用している。このネットワークでは 678 個の IPv6 プレフィックスがルーティングされている。
- AI3 (Asian Internet Interconnection Initiatives) Project
AI3 Project[7] は、衛星リンクを使って東南アジア各国を接続する研究プロジェクトである。現在は 9 ヶ国が接続され、衛星リンクを使った研究活動を行っている。AI3 Project は衛星リンクを利用した独自 IPv6 ネットワークを構築し、運用している。
AI3 Project の IPv6 ネットワークでは、38 台の IPv6 ルータが存在し、その全てで Zebra ospf6d を運用している。このネットワークでは、83 個の IPv6 プレフィックスがルーティングされている。
- WIDE Project
WIDE Project[8] は、コンピュータネットワーク技術の研究を行う研究コンソーシアムであり、基盤技術からアプリケーションまで広範な技術の研究を目的としている。WIDE Project は日本を網羅する独自の研究用ネットワークを運用しており、トンネルを利用しない IPv6 ネット

表 3: SPF 計算時間の測定値

25.615	24.881	24.755	26.349
25.075	25.480	25.207	24.990
25.508	25.160	26.441	24.685
25.150	25.282	25.669	26.821
24.724	24.637	24.770	24.986
25.518	25.233	24.723	25.342
24.997	25.088	25.437	25.505
27.062	24.768	25.105	24.750
26.300	24.669	32.714	25.054
25.524	25.809	25.535	25.558

ワークを構築している。

WIDE Project の IPv6 ネットワークには 68 台のルータが存在し、そのうち 39 台のルータが ospf6d を動作させている。他のベンダとしては、日立、juniper、cisco のルータがそれぞれ複数台動いている。このネットワークでは、168 個の IPv6 プレフィックスがルーティングされている。

無償のオープンソース実装として世界に配布しているため、Zebra ospf6d の正確な利用状況を知ることにはできない。しかし、以上の三つの実運用経験から、Zebra ospf6d を利用して IPv6 ネットワークを構築し運用できることがわかる。

4.2 SPF 計算時間

WIDE Project の IPv6 ネットワークにおいて、CPU: AMD K6-2 (586-class), 400.92 MHz、メモリ: 383 MB のホスト上で、SPF 計算にかかった時間を 40 回計ると、結果は表 3 のようになった (単位は ms)。ここから、平均は 25.522、標準偏差は 1.289 となる。これは、十分に実用可能な性能値であると言える。

5 結論と今後の課題

次世代のインターネットを良いものにするために、IPv6 ネットワークの運用経験と研究開発が必要である。これを実現するためには、多くの人が IPv6 ネットワークを構築できるツールが必要となる。本研究では、IP ネットワークを構築するために最も重要となるツール、IPv6 をサポートした無

償のオープンソース OSPF 実装を実現し、評価した。

評価では、ospf6d を実運用する組織があること、その実運用での経路計算時間が十分に小さいことを示した。ここから Zebra ospf6d は、世界中で利用される、実用に堪えるソフトウェアであることが確認できた。

今後の課題として、ルータ数やネットワーク数を数百、数千の規模に拡大した場合の Zebra ospf6d のスケーラビリティ評価が挙げられる。

参考文献

- [1] J. Postel. RFC 791: Internet Protocol, September 1981. Obsoletes RFC0760. See also STD0005. Status: STANDARD.
- [2] S. Deering and R. Hinden. RFC 2460: Internet Protocol, Version 6 (IPv6) specification, December 1998. Obsoletes RFC1883. Status: DRAFT STANDARD.
- [3] J. Moy. RFC 2328: OSPF version 2, April 1998. See also STD0054. Obsoletes RFC2178. Status: STANDARD.
- [4] IP Infusion Inc. Gnu zebra - routing software, 2003. <http://www.zebra.org/>.
- [5] J. Moy R. Coltun, D. Ferguson. RFC 2740: OSPF for ipv6, December 1999. Obsoletes RFC2740. Status: PROPOSED STANDARD.
- [6] OCCAID. Welcome to occaid, 2001. <http://www.occaid.org/>.
- [7] AI3 Project. Asian internet interconnection initiatives project, 2004. <http://www.ai3.net/>.
- [8] WIDE Project. Wide project home page, 2004. <http://www.wide.ad.jp/>.