

WIDE Technical-Report in 2008

IPv6 Multicast の設計と開発活動
wide-tr-usagi-mcast6-01.pdf



WIDE Project : <http://www.wide.ad.jp/>

If you have any comments on this document, please contact to ad@wide.ad.jp

Title: IPv6 Multicast の設計と開発活動

Author(s): USAGI プロジェクトコアメンバ (usagi-core@linux-ipv6.org)

Date: 2008-01-07

Linux におけるマルチキャストの実装

本章では、Linux におけるマルチキャスト実装の現状について述べる。IPv4 と IPv6、ならびにクライアントとしての機能とルータとしての機能に分類して述べる。なお、本文章で述べる現状に関しては、Linux kernel 2.6.24-rc5 に準拠している。

IPv6 マルチキャストの実装

IPv6 マルチキャストクライアントの機能としては、MLDv1 と MLDv2 をサポートしている。IPv4 と同じく、マルチキャストルータがサポートしている機能によって判断して自動的に切り替える仕組みとなっている。

マルチキャストルーティング機能は、現在の Linux kernel には実装されていない。過去に、いくつかのプロジェクトによる実装があったが、本流 kernel には取り入れられることはなかった。現在利用できるマルチキャストルーティング実装としては、次の2つである。

- Linux IPv6 Multicast Forwarding (http://clarinet.u-strasbg.fr/~hoerdt/dev/linux_ipv6_mforwarding/)
- MRD6 (<http://unix.freshmeat.net/projects/mrd6/>)

Linux IPv6 Multicast Forwarding 実装は、Linux kernel 2.6.7 に対するパッチ形式となっている。この実装は既に開発が停止しているが、USAGI プロジェクトメンバーの手によって、USAGI プロジェクトの開発 git ツリーに取り込まれており、現在 Linux kernel 2.6.24-rc5 にて動作するよう修正が加えられている。この実装は IPv4 のマルチキャストルーティング実装、ならびに BSD 系 IPv6 マルチキャストルーティング実装と類似のものであり、マルチキャスト経路をフォワーディングキャッシュとして保持する実装となっている。

一方、MRD6 は一風変わった実装であり、パケットフォワーディングならびにパケットコピーをユーザランドのデーモンで行ってしまう実装である。PACKET socket を利用してユーザランドからパケットの読み取り、書き込みを行ってマルチキャストルーティングを実現している。debian にはパッケージも用意されており、カーネルを入れ替えることなく手軽にインストールならびに利用することのできる実装となっている。また、更新の頻度は高くないものの開発は継続されている。しかし、USAGI プロジェクトでは MRD 実装の検証は行っていないため、動作に関しては不明である。

また、IPv6 PIM ルーティングデーモンの実装は、次の3つである。

- pim6sd for Linux (http://clarinet.u-strasbg.fr/~hoerdt/dev/pim6sd_linux/)
- MRD6 (<http://artemis.av.it.pt/mrd6/>)
- XORP (<http://www.xorp.org/>)
- mcast-tools(<http://sourceforge.net/projects/mcast-tools/>)

pim6sd for Linux の実装は、前述の Linux IPv6 Multicast Forwarding カーネル実装に対応した pim6sd の実装である。BSD 用の pim6sd をもとに、Linux にて動作するよう改造を加えたものである。MLDv2/PIM-SSM にも対応している。しかし、ベースとなっている pim6sd は 2003 年当時の実装のままであり、本家の pim6sd の更新には追従しておらず、開発も止まっている。

MRD6 は前述の通りユーザランドで機能するマルチキャストルーティング実装であるため、PIM ルーティングデーモンもそのパッケージに含まれる。しかし動作は不明である。

XORP はルーティングプロトコルデーモンの集合パッケージであり、その中に PIM ルーティング機能も含まれる。しかし、XORP はあくまでもルーティングプロトコルの集合パッケージであるため、IPv6 マルチキャストルーティング機能の無い Linux kernel 上において動作させた場合には、PIM-SM ルーティングプロトコルをサポートするのみで、マルチキャストルーティングを行うことはできない。したがって、実質上 IPv6 PIM-SM ルータとして機能させることはできない。また、PIM-SSM もサポートされていない。

mcast-tools は主に BSD 用の IPv6 PIM-DM, PIM-SM デーモンを提供しているパッケージである。過去に Linux サポートも盛り込まれていたため、Linux 用コードも存在している。しかし、Linux サポートは 2006 年 7 月を最後に中断されており、現状そのままではコンパイルすることができない。

そこで USAGI Project では、メインラインカーネルに IPv6 マルチキャストルーティング機能を統合すべく、前述の Linux IPv6 Multicast Forwarding 実装ならびに pim6sd for Linux、mcast-tools 実装をベースに、最新 Linux kernel にて IPv6 マルチキャストルーティングを提供するための作業を 2006 年度から行っている。

IPv6 マルチキャストルーティング設定

本文章では、Linux IPv6 Multicast Forwarding ならびに pim6sd for Linux および mcast-tools を用いた IPv6 PIM-SSM の設定例ならびに検証結果について示す。なお、最新 IPv6 Multicast Forwarding を含んだカーネルコードは、USAGI プロジェクトの git ツリーにて公開されている。また、このカーネルに対応した pim6sd は、その修正が完全に検証されていないため、未リリースである。修正が検証され次第、リリースする。

まず、カーネルコンパイル時のオプションとして、以下を有効にする。

```
IPV6_MROUTE
IPV6_PIMSM_V2
```

次に、カーネル起動後、以下の設定を sysctl もしくは proc filesystem を用いて行う。

```
net.ipv6.conf.all.mc_forwarding = 1
```

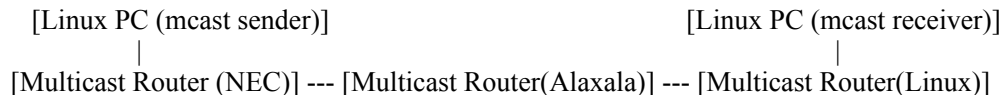
さらに、pim6sd.conf を以下のように設定し、pim6sd を起動する。

```
phyint eth1 mld_version any;
phyint eth2 mld_version any;
log all;
```

phyint にてマルチキャストルーティングを有効にする物理インタフェースを指定し、MLD バージョンを指定する。

pim6sd 起動後は、pim6stat コマンドを用いて pim6sd の状態を見ることができる。pim6stat コマンドを実行すると、/var/run/pim6sd.dump というファイルが生成され、このファイルに状態が記録されている。

動作検証は以下の構成にて行った。



ルーティングプロトコルは IPv6 PIM-SSM を使い、ssmpingd/ssmping ツールを用いて動作検証を行った。この際の Multicast Router (Linux)における pim6stat の結果を以下に示す。

Multicast Interface Table

Mif	PhylF	Local-Address/Prefixlen	Scope	Flags
0	eth0	fe80::213:72ff:fe3b:c1fb/64	2	DR PIM QRY
		2001:200:1b0:1000:213:72ff:fe3b:c1fb/64	0	
		Timers: PIM hello = 0:15, MLD query = 1:50		
		possible MLD version = 1 2		
1	eth1	fe80::213:72ff:fe3b:c1fc/64	3	DR QRY NO-NBR
		2001:200:1b0:ffe::2/64	0	
		Timers: PIM hello = 0:15, MLD query = 1:50		
		possible MLD version = 1 2		
2	lo	::1/128	0	DISABLED
		Timers: PIM hello = 0:00, MLD query = 0:00		
		possible MLD version = 1		
3	regist	fe80::213:72ff:fe3b:c1fb/64	2	REGISTER
		Timers: PIM hello = 0:00, MLD query = 0:00		
		possible MLD version = 1		

PIM Neighbor List

Mif	PhylF	Address	Timer
0	eth0	fe80::2000:1	90
		2001:200:1b0:1000::2000:1	

MLD Querier List

Mif	PhylF	Address	Timer	Last
0	eth0	fe80::213:72ff:fe3b:c1fb	255	46s

```
1 eth1 fe80::213:72ff:fe3b:c1fc          255          46s
```

Reported MLD Group

```
Mif PhylF Group(Group-Timer,MLD-ver(Filter-Mode,Compat-Timer))/Source(TimerID)
1 eth1 ff3e::4321:1234 (#0 (v2 IN #1024))
    2001:200:0:1c04:213:72ff:fe52:b05f (#19)
```

Multicast Routing Table

```
Source      Group      RP-addr      Flags
-----
(S,G)-----
2001:200:0:1c04:213:72ff:fe52:b05f ff3e::4321:1234 NULL      CACHE SG
Joined oifs: ....
Pruned oifs: ....
Leaves oifs: .l.
Asserted oifs: ....
Outgoing oifs: .o..
Incoming : l...
Upstream nbr: fe80::2000:1
```

TIMERS: Entry=0 JP=60 RS=0 Assert=0

```
MIF 0 1 2 3 4 5 6 7 8 9
    0 0 0 0 0
```

-----(*,RP)-----

```
Number of Groups: 1
Number of Cache MIRRORS: 1
```

-----RP-Set-----

```
Current BSR address: 2001:200:1b0:ffe::2 Prio: 0 Timeout: 25
RP-address(Upstream)/Group prefix      Prio Hold Age
2001:200:1b0:ffe::2(myself)
    ff00::/8          0 150 145
```

-----CallOut Timer Queue-----

```
TimerID Expiry-Time[s]
#20 5
#19 254
```

上記の通り、mcast receiver は ff3e::4321:1234 というマルチキャストグループに join し、そのマルチキャストグループの mcast sender は 2001:200:0:1c04:213:72ff:fe52:b05f というホストであるということが認識されている。この状態で mcast receiver はマルチキャストパケットを受信できており、1 台の Linux ルータを含む、3 台のマルチキャストルータを経由した IPv6 マルチキャストルーティングを検証することができた。

今後の方針

Linux IPv6 Multicast Forwarding ならびに mcast-tools の動作検証ならびに改造を進め、安定した動作が行えるレベルまで達した段階で、一度メインラインカーネルへの統合を進める方針である。

メインラインカーネルに統合した後は、Linux のカーネル構造に適したマルチキャストルーティング機構への改造を計画している。現在の Linux における IPv6 マルチキャスト経路表の実装は、BSD 系と同じく、フォワーディングキャッシュを利用した実装となっている。通常の IPv6 ユニキャスト経路表において、ff00::/8 宛の経路の next-hop を :: とし、そのパケット処理関数の ip6_mc_input 内にてマルチキャストルーティングの処理を追加することで実現している。

USAGI プロジェクトでは、このマルチキャスト経路情報を、フォワーディングキャッシュとして持たせるのではなく、従来の IPv6 ユニキャスト経路表の枠組みを使って管理できるよう再設計しようと試みている。マルチキャスト経路の設定、削除も netlink socket を利用したものに変更する。しかし、この場合にも従来の setsockopt API も受け付けるよう実装することにより、従来の pim6sd といった経路制御デーモンを変更することなく利用できるよう設計、改造することを目標とする。

Copyright Notice :

Copyright(C) USAGI/WIDE Project 2006, 2007, 2008. All Rights Reserved.