

# DNSミーティング: IPv6 DNSの現状

神明 達哉

(株)東芝 研究開発センター/KAMEプロジェクト  
jinmei@{isl.rdc.toshiba.co.jp, kame.net}

Copyright (C) 2001 Toshiba Corporation.

# 内容

- IPv6のRR
  - AAAA, ip6.int.
- A6 RR
  - 定義、問題点、最近の動向
- ip6.arpa.
  - 現状、移行手段
- DNSのIPv6トランスポート
  - 現状、運用方法
- DNS server discovery
  - 背景、動作例、実装
  
- (not covered) EDNS0とIPv6

# IPv6アドレスのためのRR

□RFC 1886

□正引き: AAAA RR

www.kame.net. IN AAAA =>

3ffe:501:4819:2000:280:adff:fe71:81fc

www.kame.net. IN A 203.178.141.220

□逆引き: PTR RR

○上位ドメイン: ip6.int.

○4ビットずつを1ラベルとし、下位バイトから逆順に32階層で表現

;;2001:0200:0000:4819:0280:adff:fe71:81fcの逆引き

\$ORIGIN 9.1.8.4.0.0.0.0.0.2.0.1.0.0.2.ip6.int.

c.f.1.8.1.7.e.f.f.f.d.a.0.8.2.0 IN PTR www.kame.net.

□実装

○サーバ: BIND 4/8/9, djbdnsで利用可能

□リゾルバ: 多くのOSの標準リゾルバで対応

○\*BSD/Solaris/Linux/Windows XP

# A6 RR (1/2)

## □RFC 2874

- RFC 1886の「おきかえ」

## □IPv6のアドレス割り当て構造に応じてRRを分割して登録

- 1つのアドレスを複数のRRで表現する
- AAAA RRの例

www.kame.net. AAAA 3ffe:501:4819:2000:280:adff:fe71:81fc

- これを3つのA6 RRで表現した例

www.kame.net. A6 64 ::2000:280:adff:fe71:81fc karigome.wide.ad.jp.

karigome.wide.ad.jp. A6 48 0:0:4819:: wide.ad.jp.

wide.ad.jp. A6 0 3ffe:501::

## □A6 RRの目的

- マルチホーム: 下位ビットに対応するRRを共有できる
- リナンバリング: 上位ビットのRRだけ書き換えればよい

# A6 RR (2/2)

## □A6 RRの問題点

### ○複雑

- ▶設定間違いを生みやすい
- ▶実装が大変(実際、使える実装はごくわずか)

### ○効果が疑問

- ▶複数アドレスには設定用スクリプトでも対処できる
- ▶頻繁なリナンバリングは現実味が薄い

## □IETFでの議論: A6を採用すべきか

### ○2001/3 dnsexp wg

- ▶利点・欠点を整理してMLで議論する

### ○2001/8 ngtrans wg

- ▶A6は"experimental"に。実運用はAAAAでやる。

### ○現在

- ▶最終合意に向けてMLで議論中

# IPv6用RRに関するその他の話題

## □DNAME, ビットラベル

- 「A6の逆引き版」
- A6同様 "experimental"に

## □ip6.arpa.ドメイン

- RFC 3152
- ARPA: Address and Routing Parameters Area
- IPv6逆引き用の新しい上位ドメイン
  - ▶政治的な理由により導入: int.は他の用途に使いたい
  - ▶移行は避けられない雰囲気
  - ▶慌てる必要はないが、心構えは必要
  - ▶2001:0200::/24がAPNICまで委譲

# ip6.arpa.への移行(1/2)

- リゾルバ側: ip6.arpa. と ip6.int. の両方を試す
  - 実装待ち
- サーバ側: 運用上の問題(実装は変えなくてよい)
  - ip6.arpa. と ip6.int. の両方を管理
- プライマリサーバの設定例 (BIND 8, 9)
  - 単一のゾーンファイルを二つのゾーンで共有
  - \$ORIGINによる上位ドメイン変更に注意

(in named.conf)

```
zone "9.1.8.4.0.0.0.0.0.2.0.1.0.0.2.ip6.int." {
    type master;
    file "2001:200:0:4819::zone";
};
zone "9.1.8.4.0.0.0.0.0.2.0.1.0.0.2.ip6.arpa." {
    type master;
    file "2001:200:0:4819::zone"; //ip6.int.と共有
};
```

(in 2001:200:0:4819::zone)

```
;;$ORIGIN 9.1.8.4.0.0.0.0.0.2.0.1.0.0.2.ip6.int. うまくいかない
c.f.1.8.1.7.e.f.f.d.a.0.8.2.0 IN PTR www.kame.net.
```

## ip6.arpa.への移行(2/2)

### □ セカンダリサーバの設定例 (BIND 8, 9)

```
zone "9.1.8.4.0.0.0.0.0.2.0.1.0.0.2.ip6.int." {  
    type slave;  
    file "bak/2001:200:0:4819::int.zone";  
    masters ...  
};  
zone "9.1.8.4.0.0.0.0.0.2.0.1.0.0.2.ip6.arpa." {  
    type slave;  
    file "bak/2001:200:0:4819::arpa.zone"; //共有しない  
    masters ...  
};
```

# DNSのIPv6トランスポート

## □DNSのトランスポート

- 問い合わせ、応答、ゾーン転送に使うプロトコル
- 例: TCP/IPv4, UDP/IPv4, TCP/IPv6, UDP/IPv6...

## □3つのケース

- 1. IPv6用のRR, IPv4トランスポート
- 2. IPv6用のRR, IPv6トランスポート
- 3. IPv4用のRR, IPv6トランスポート
- (IPv4用のRR, IPv4トランスポート => IPv6とは無関係)
  - ▶現在稼働している実装の多くは1のみ
  - ▶1.のみで実運用上は問題ないが、IPv4に依存しない環境構築には2.が重要
  - ▶2.をできる実装はふつう3.もできる

# IPv6 トランスポートの現状

## □ TLDの現状: IPv6アドレスを持つネームサーバの数

- ルート, gTLD: 13ネームサーバ中ゼロ

- ▶ 改善の試みが進展中

- ccTLD: 243ゾーン、584ネームサーバ中

- ▶ 2ゾーン、3サーバ

- ▶ ルートゾーンからのグルーRRなし

## □ 実装状況

- サーバ

- ▶ BIND 9

- リゾルバ

- ▶ FreeBSD/NetBSD/OpenBSD/Linux

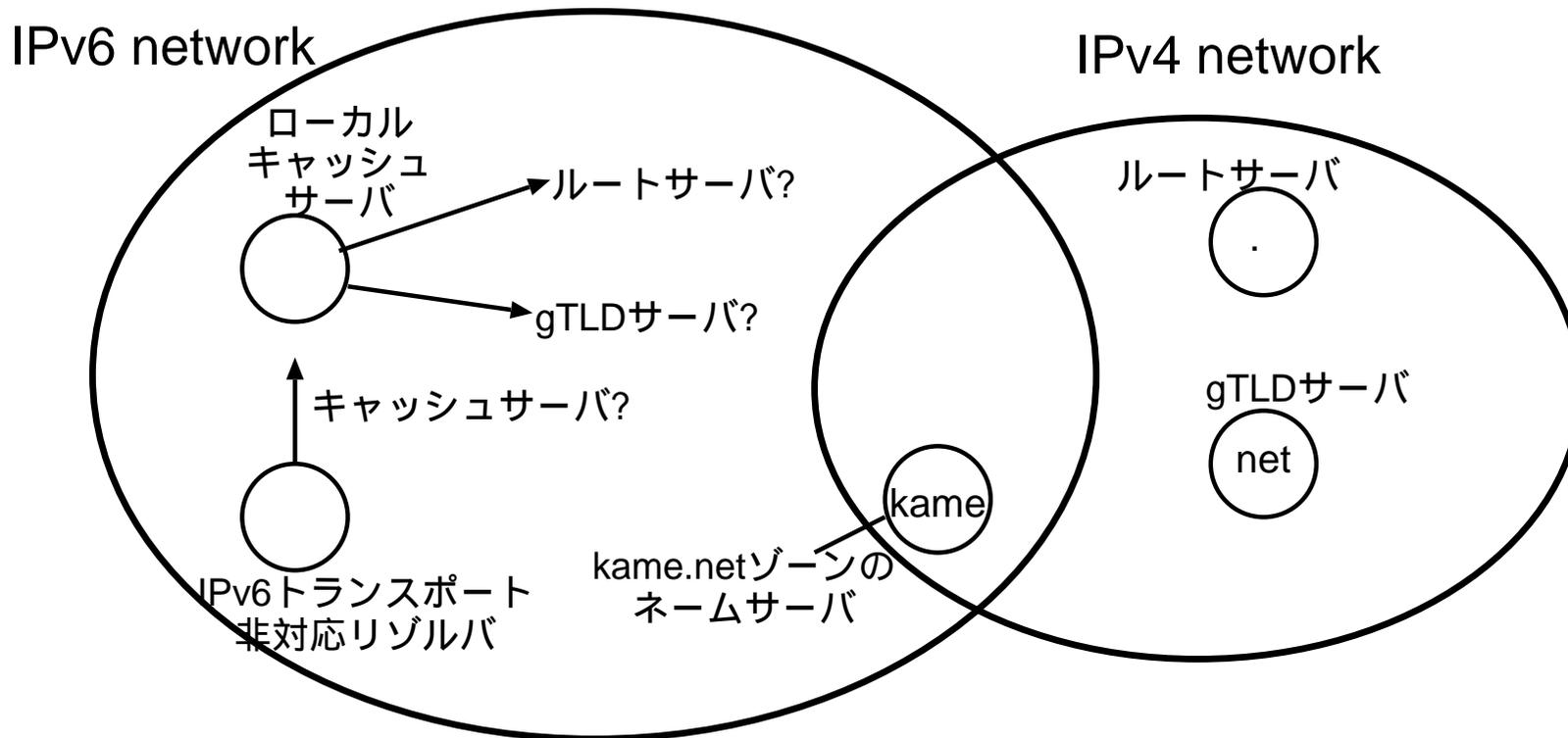
- ▶ BIND9 lwres: アプリケーションの再コンパイルが必要

- ▶ BIND8, 9 libbind: OS標準ライブラリへの統合はこれから

- ▶ BSD/OS, Solaris, Windows...は未対応

# 現状の問題点

- IPv6トランスポートに対応しないスタブリゾルバはキャッシュサーバへの問い合わせができない
- IPv6のみのネットワークにいるキャッシュサーバは名前解決できない



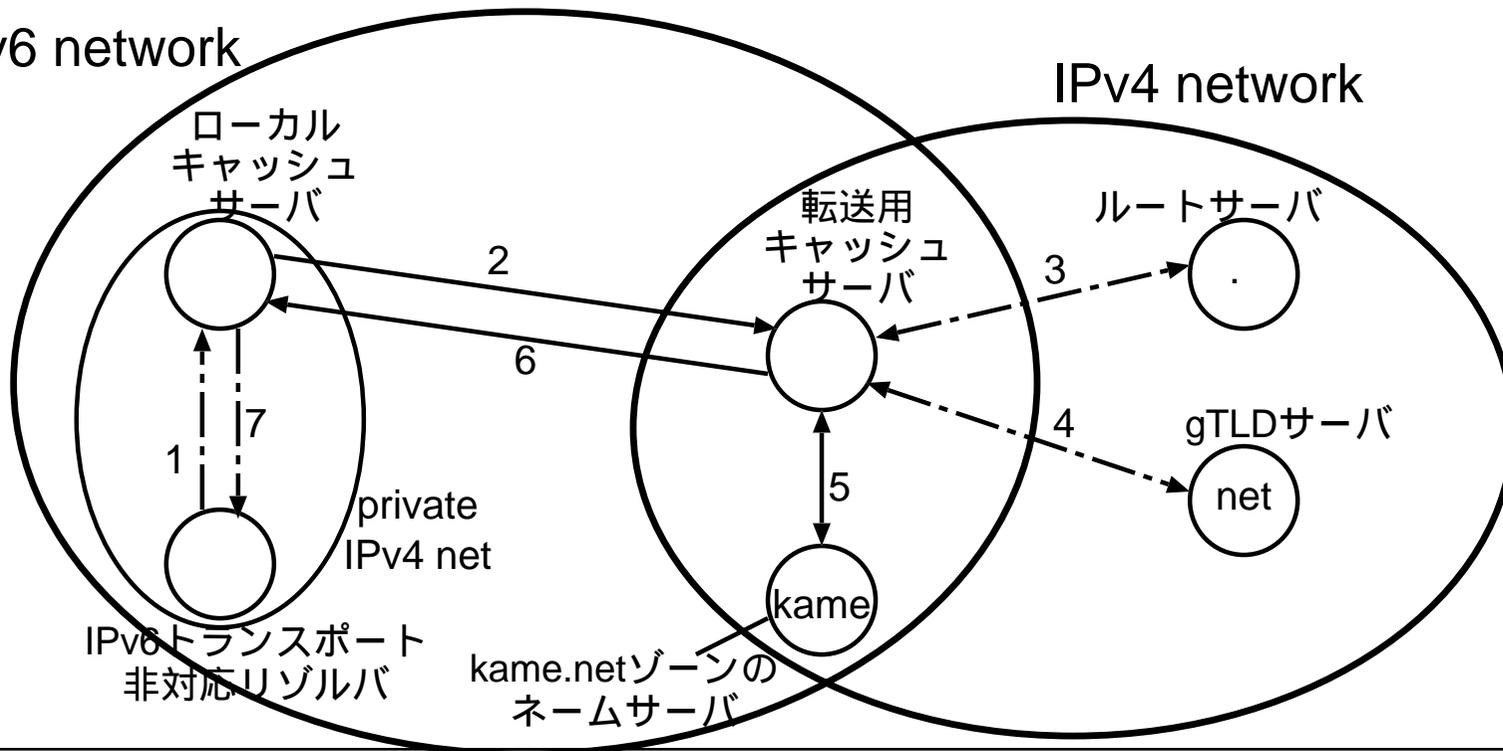
# 運用での対応

- 名前解決用のIPv4ローカルネットワーク
- デュアルスタックの転送用キャッシュサーバ

-----> IPv4による問い合わせ、応答  
—————> IPv6による問い合わせ、応答

IPv6 network

IPv4 network



# DNS server discovery: 背景

## □DNS(キャッシュ)サーバのアドレス設定

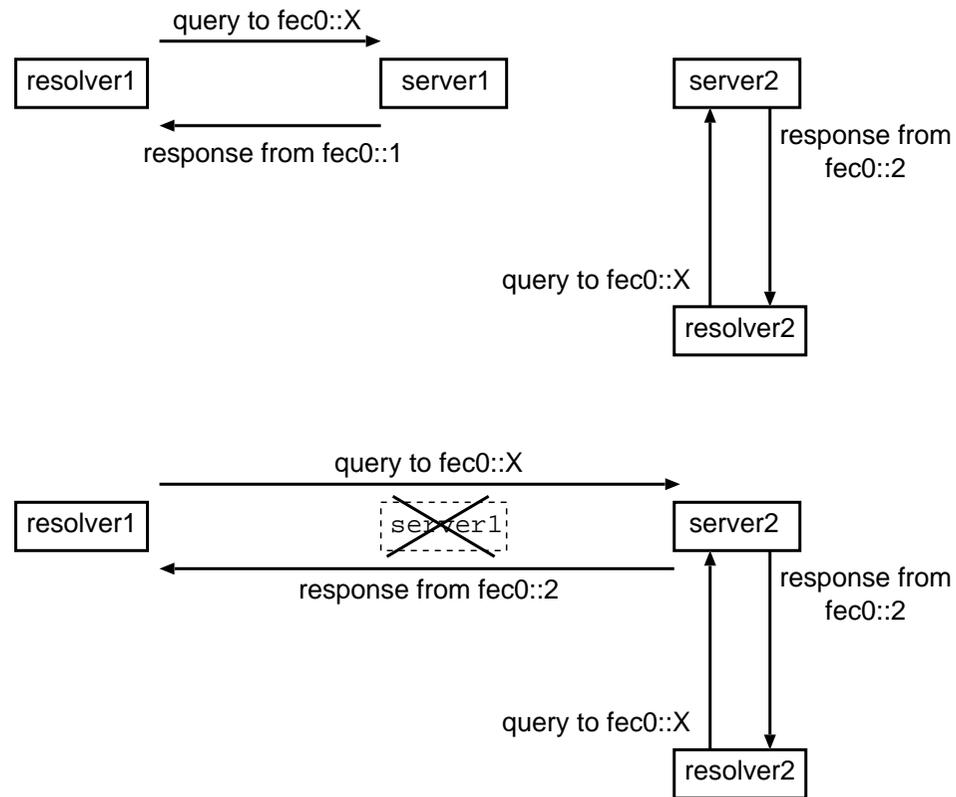
- resolv.confを手で書きたくない
- DHCPv6はあまり使われてない

## □well-known anycastによる解

- draft-ietf-ipngwg-dns-discovery-03.txt
- resolv.confではエニキャストアドレスだけ指定
- 「一番近い」サーバがユニキャストで応答
- リゾルバはアドレスのチェックを省略して応答を利用
  - ▶「問い合わせの宛先 != 応答の送信元」を許す

# DNS server discovery: 動作例

- server1: fec0::X (anycast), fec0::1 (unicast)
- server2: fec0::X (anycast), fec0::2 (unicast)



# DNS server discovery: 実装と今後

## □実装

### ○サーバ側

▶todd, BIND 9のKAMEパッチ

### ○リゾルバ側

▶\*BSD KAME snapのlibinet6 ("options insecure1")

▶KAME mdnsd

### ○入手

▶<http://www.kame.net/>

## □今後

### ○well-known addressの決定

▶進行中

▶I-D 03ではfec0:0:0:ffff::1, fec0:0:0:ffff::2, fec0:0:0:ffff::3

### ○セキュリティを考えた運用方法の確立