DNS と IPv6
IPv6 への移行とその課題

加藤 朗

WIDE

慶應義塾大学/WIDE Project
kato@wide.ad.jp

今日のお話

☆ DNS について（復習）
☆ DNS における IPv6
☆ メッセージ長の話
☆ Anycast
☆ Root Zone の IPv6 対応
☆ JP サーバの状況
☆ Root サーバでの IPv6 対応

IPv6 は本当に必要か

☆ 詳細はこのあとの荒野さんの講演で
☆ 加藤の今の考え
- IPv4 で未来永劫大丈夫か？
  - ダメじゃないかしら
  - アドレス枯渇の多少の遅延は可能だろうけど
- IPv6 への移行はコストがかかるのは事実
  - でも IPv4 に留まるコストも評価するべき
  - 感情論の問題ではない
- IPv6 に移ることを前提にしたい
  - 当分は IPv4 と IPv6 を併用

DNS

☆ 名前の管理
- 非常に重要

☆ Domain Name System
- 木構造の名前空間
- その解決システム

☆ Name Space : "." を root にする木
- 部分木の管理を委任
  - 分散的な管理・運用が可能
- 同じ名前でも domain が異なれば OK
  - eg. www.csi.ad.jp, www.keio.ac.jp
- FQDN : Fully Qualified Domain Name
- FQDN は一意に識別可能
  - Root が共通なら...

DNS

☆ **Domain Name System**
- 名前から各種情報への対応を提供
  - Q: www.keio.ac.jp のアドレスは？
  - R: 131.113.1.1 です
- サービスとしてはアプリケーションの一つ
  - TCP/UDP の port 53
- 例外を除きアクセス対象は「名前」で指定
  - アドレスは覚えられない
  - アドレスは ISP を変えると変わる

→ 基盤サービスの一つとして位置づけられる
- DNS が止ると
  - パケットは相手に届く
  - でも相手のアドレスが分からない
  - 使い物にならない

DNS による名前解決

☆ **DNS サーバに問い合わせる**
- 基本的には Root DNS サーバから始める
- 答えが帰ってくる
  - 「その子のアドレスは xxxx だぜ」
- 他の DNS サーバを紹介される
  - 「俺は知らないなあ」
  - 「でもあいつに聞いてみたら？」
  - 「あいつのアドレスはちなみに xxxx ですぜ」
  - （これってたらい回し？）
- 有限回のたらい回しで答えが分かる
  - ちょっとだけ時間がかかる

DNS による名前解決

☆ 得られたデータは暫く覚えておく
- 中間結果も含めて
- 検索の効率化
- サーバの負荷の「劇的」な軽減
  - 特に上位ドメインのサーバ

☆ どれぐらい覚えておくか
- 各 record には TTL が設定
  - DNS の情報設定に基づく
- JP では 1 日 (86400秒)、Root は 41 日
- 変わる可能性があるものは短く
  - 障害時に問題が発生する可能性
  - APNIC の ISP 障害による逆引きの問題
- 負荷分散目的に極端に短い値にする場合も
  - 60 秒

DNS による名前解決

☆ 一般の計算機がたらい回しに対応するのは大変
- 最寄りの DNS サーバに問い合わせる
- そのサーバが最終的な答えを取得する
  - たらい回しを頑張る
- 最終的な答えが得られる
  - 一般の計算機はらくちん
- どれが DNS サーバなの？
  - （普通は）DHCP/IPCP が教えてくれる
- 沢山のクライアントで共有
  - Cache の利用効率が高まる期待

DNS と IPv6

☆ IPv6 のサポートの方法
- Class "IN" を IPv4 と共有する
  - インターネットの名前
  - 名前は IP version には依存しない
  - 逐次的な移行が可能
- Class "IN6" を定義する
  - まったく別な名前空間を定義
  - IPv4 と IPv6 の混在問題は発生しない
  - 使うプロトコルで名前が異なる（可能性）

☆ 共有木を選んだ
- Class "IN"

DNS と IPv6

☆ 二つの側面
- IPv6 アドレスをどう表現するか
  - コンテンツ、中身の問題
- IPv6 を使って名前の解決ができるか
  - トランスポートの問題

☆ 基本的にはこの二つは独立
- IPv4 を使って IPv4 アドレスを知る
- IPv4 を使って IPv6 アドレスを知る
- IPv6 を使って IPv4 アドレスを知る
- IPv6 を使って IPv6 アドレスを知る

IPv6 アドレスの表現

☆ **IPv4 アドレスは A レコード**

```
$ORIGIN wide.ad.jp.
www    IN   A     203.178.136.57
```

☆ **IPv6 アドレスは AAAA レコード**
- A が 32bit なら 128bit はその４倍

```
$ORIGIN wide.ad.jp.
www    IN   AAAA  2001:200:0:1::5
```

- RFC1886 で規定（1995 年 12 月）
- 全ての DNS ソフトウェアでサポート
  - 一部の組み込みのソフトウェアは....
  - 例 : AAAA を聞いても A を返す

☆ **type が違う（大きさも）以外は差はなし**
- アドレスを全ビット表現する

IPv6 アドレスの表現（道草）

☆ **A6 レコード**
- IPv6 アドレスの一部だけを記述
  - 残りの部分は別のサーバに聞いてね
- アドレス変更時に対応が簡単
  - 問合わせ際のたらい回しが増加
  - お役人も真っ青なケースも懸念

☆ **2001 年の IETF で却下**
- 非常に沢山の問合わせが必要な場面も
- DNS はリナンバ作業のごく一部だけ

☆ **結論：AAAA を使う**
- A6 は "experimental（実験的な）" に分類

IPv6 アドレス（おまけ）

☆ **AAAA を追加する場合**
- 全部のサービスが IPv6 対応であること

☆ **問題な例**
- DNS と WEB サーバを兼ねている
- しかし、DNS だけが IPv6 対応

```
server    IN  A    192.0.2.1
          IN  AAAA 2001:db8::1
```

- IPv6 対応な web client
  - IPv6 を try し、IPv4 に fallback

```
ns        IN  A    192.0.2.1
          IN  AAAA 2001:db8::1
www       IN  A    192.0.2.1
```

逆引き

☆ **IPv6 アドレス**
- 略さず全部書き、: を取り、. で繋ぐ
- ip6.arpa をつける
- 例 : 2001:200:0:1::5 の場合

```
$ORIGIN 1.0.0.0.0.0.0.0.0.2.0.1.0.0.2.ip6.arpa.

5.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0  IN PTR www.wide.ad.jp.
```

- 長いけど、機構は単純
- 手で書くのはあまり現実的ではないかも
  - 機械的に生成するのが簡単

☆ **本当に逆引きは必要なの？**
- サーバやルータ : あった方がいいかも
- End Host : さて、どうしたものか
  - 下位 64bit はそのままでも...

DNS メッセージ

☆ 基本的には **UDP** を使う
- RFC1123 : まず UDP を使え（MUST）

☆ 本質的に必要な場合だけ **TCP**
- 応答が UDP のメッセージ長の上限を越える場合
- Zone（ある DNS データの固まり）転送をする場合
  - 複数の DNS サーバ間でのデータの同期

☆ どうして？
- UDP なら、問合わせと応答で終了
  - 簡単、早い、サーバの負荷が軽い
- TCP だと 3 RTT 7 メッセージ
  - 高い負荷、遅い、サーバ側の状態管理

☆ ただし、**UDP** にはメッセージ長の上限
- 512byte
  - fragment を避けるため

DNS メッセージ長

☆ 問題
- AAAA レコードは大きい
  - A レコード 16byte
  - AAAA レコード 28byte
- 答えだけならあまり問題ない
  - 沢山 AAAA が列挙されていない限り

☆ 参照を回答する際には問題になる
- 「それならこの子に聞いてね」

DNS メッセージ長

☆ 参照を指示する場合のメッセージ
- 「それならこの子に聞いてね」

```
$ORIGIN ad.jp.
wide          IN  NS  ns-wide.wide
```

- 「その子の IP アドレスは xxx よ」

```
ns-wide.wide   IN  A    203.178.136.59
               IN  AAAA 2001:200:0:1::5
```

☆ 一般的には複数の DNS サーバを列挙
- 添付するアドレスも複数
- IPv6 を「追加」するとメッセージは太る
  - 512byte を簡単に超えちゃう

DNS メッセージ長

☆ 対策：EDNS0 (RFC2671, 1999 年 8 月)
- DNS メッセージに対する拡張機能
- 問合わせに情報を添付
  - 「僕、xxx byte までなら大丈夫」

☆ EDNS0 対応でないサーバ
- SERVFAIL を返す
- この場合、EDNS0 無しで再試行すべし

☆ 多少大きなメッセージになっても大丈夫
- あまり大きなものは依然として問題
- Fragmentation は避けたい
- IPv6 : Minimum MTU 128byte
  - あるいは Path MTU discovery してね
- 「大抵」これで大丈夫
  - DNSSEC しない限り
- bind-8.3 以降、bind-9 で対応
- 対応を義務化しようという意見もある

コンテンツとトランスポートの関係

☆ 基本的に無関係
- IPv4 で AAAA レコードの問合わせ
  - まったく正しい
- IPv6 によって A レコードの問合わせ
  - なんの問題もない

☆ NS レコードによる「紹介」の場合
- アドレス情報を添付する
  - じゃないと名前解決が進まない
- IPv4 でも IPv6 でも同じ
  - A/AAAA に関わらずあるものは教える
  - アドレス情報が増える
  - パケットが太る
  - EDNS0 は必須機能

名前空間の分割

☆ もしある名前のサーバが IPv6 only だとすると
- 一つも IPv4 アドレスがない場合

☆ IPv4 のみな DNS サーバからは名前解決ができない
→ 専門用語では Name Split という

☆ これは避けたい
- 全ての名前は解決可能であること
- 本当にアクセスできるかどうかはともかく

☆ 対策
- IPv6 only な DNS サーバはあってもいい
- 対応する DNS サーバに当面 IPv4 があること

Root DNS Server

☆ **DNS の要となるサーバ**
- **現在 13 "文字" が稼働中**
  - [A-M].ROOT-SERVERS.NET
  - **共通な名前で圧縮効率を向上**
  - 1995 **年からこの名前方式を採用**

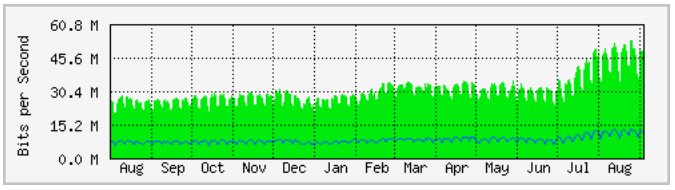
☆ **1997 年 1 月までは A-I のみが可動**
- 'I' **のみが** U.S. **外部の** Root **サーバだった**
- 'K' **が** 1997 **年** 5 **月に** London **で運用開始**
- 'M' **が** 1997 **年** 8 **月に東京で運用開始**

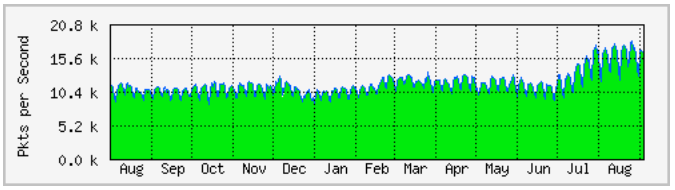
☆ **13 は実用上最大数**
- EDNS **に対応していない場合のパケット長**

☆ **Root サーバで提供されているもの**
- TLD
- ARPA と IN-ADDR.ARPA も

M-Root の問合わせ

☆ **非常に安定**
- **最近ちょっと増加傾向か**





Priming

☆ **DNS サーバはどうやって Root を知るの？**
- "root.cache" ファイルを参照
  - ftp://ftp.internic.net/domain/named.root
- 最新じゃないことも（更新をサボっている場合）

☆ **BIND はそれをヒントとして使う**
- ヒントから Root サーバを一つ選択
- その子に問合わせ
  - QNAME= . QTYPE=NS QCLASS=IN
  - （現在の Root サーバとそのアドレスを求む）
- すると最新情報が手にはいる
  - タイムアウトしたら別なサーバに再試行
- 以降はその情報を元に検索を行う

☆ **この課程は "priming" と呼ばれている**
- ヒントが多少古くても大丈夫

応答メッセージ長

☆ **Priming の応答は 436 bytes だった**
- fixed header + 13 NS + 13 A + 0 AAAA

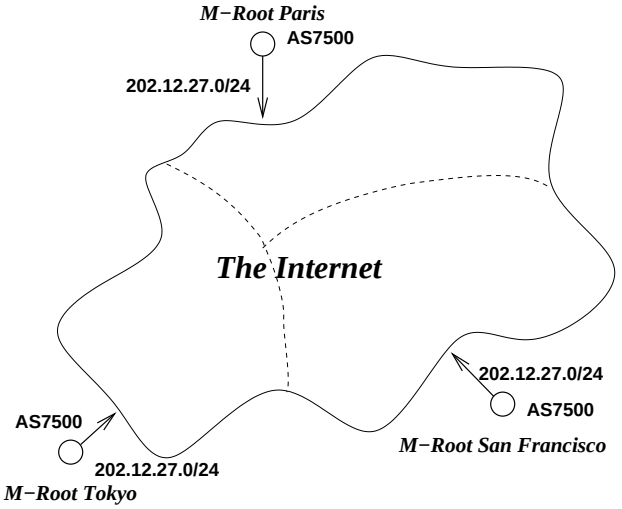
☆ **全部が IPv6 対応になると 800 bytes**
- fixed header + 13 NS + 13 A + 13 AAAA
- あるいは 811 (EDNS0 pseudo-RR つき)

☆ **Priming ではない問合わせ**
- 対象の名前は最大 255 bytes
  - 残りは 241 bytes
- 全部の情報は入りきらない場合も

☆ **NS レコードの数を減らす**
- 可用性が減る可能性がある
  - Anycasting は強力なツール

Anycasting (RFC3258)

☆ 同一の経路情報を複数の地点からアナウンス
- IGP や、大抵は BGP



Anycasting (2)

☆ これは **DNS** の技術ではない
- 経路システムに対する小細工

☆ 問合わせは近くのサーバに配送される
- 決定は経路システムが行う
- もっとも近いとは限らない
  - BGP の経路選択は AS-Path 長に基づく

☆ サーバ側での状態管理が必要な場合には不適切
- 経路決定が変わるとサービスが中断
- 大抵の TCP によるサービスは問題

☆ 冪等 **(idempotent)** な **UDP** サービス向き
- DNS はその代表例

Root サーバでの Anycasting

☆ 全部で **144 cluster**

・ A/B/D/E/G/H : anycasting していません

C : IAD, LAX, CHI, NYC (IGP anycast)
F : YOW, KPAO, SJC, NYC, SFO, MAD, HKG, LAX, ROM
F : AKL, SAO, PEK, SEL, MOW, TPE, DXB, PAR, SIN
F : BNE, YTO, MRY, LIS, JNB, TLV, JKT, MUC
F : OSA, PRA, AMS, BCN, NBO, MAA, LON, SCL, DAC
F : KHI, TRN, CHI, BUE, CCS, OSL, PTY, UIO
I : STO, HEL, MIL, LON, GVA, AMS, OSL, BKK, HKG
I : BRU, FRA, ANK, BBU, CHI, WAS, TYO, KUL, KPAO
I : JKL, WLG, JNB, PER, SFO, NYC, SIN, MIA
I : IAD, BOM, PEK, MNe, DOH
J : IAD(2), MIA, ATL, SEA, CHI, NYC, LAX, Mountain View
J : SFO, AMS, LON, ARN, TYO, SEL, PEK, SIN, DUB,
J : KUN, NBO, YUL, SYD, CAI, WAW, BSB, SOF,
J : PRG, JNB, YYZ, BUE, MAD, VIE
K : LON, AMS, FRA, ATH, DOH, MIL, RKV, HEL, GVA
K : POZ, BUD, AUH, TYO, BNE, MIA, DEL, OVB
L : LAX, MIA
M : TYO, SEL, PAR, SFO

Root Zone での IPv6

☆ **Root Zone での IPv6 のアドレス情報**

・ JP と KR は 2004 年 7 月 21 日に登録
- Root サーバは生成された情報を提供するだけ

・ サーバが複数の TLD を担当していたら
- IPv6 化は全部の TLD からの承諾が必要

・ IPv6 でアクセスできる TLDs 数（全部で 271）
- 106 サーバで 110 TLD
- 2005 年 11 月は 68/68 だった

AD, AE, AERO, AF, AG, AM, AN, AQ, AR, AS, ASIA, AT, AU, BE,
BI, BIZ, BR, BT, BW, CA, CAT, CH, CI, CL, CN, COM, CX, CY,
CZ, DE, FI, FM, FR, GA, GB, GG, GH, GI, GN, GR, GT, GW, GY,
HK, HN, HU, ID, IE, IL, IN, INFO, INT, IT, JE, JP, KG, KH
KI, KR, LC, LI, LK, LR, LS, LU, MD, ME, ML, MN, MOBI, MT,
NA, NET, NG, NL, NP, NR, NU, NZ, ORG, PH, PL, PM, PT, PY,
RE, RS, SC, SD, SE, SG, SI, TEL, TF, TN, TL, TN, TP, TRAVEL,
TW, UK, US, UY, VA, VC, VE, WF, YT, ZA, ZM

JP TLD

☆ 5 つの DNS サーバ
- A.DNS.JP : JPRS (TYO, OSA), IPv4/IPv6
- B.DNS.JP : JPNIC (TYO), IPv4 のみ
- D.DNS.JP : IIJ (TYO, OSA, SFO, JFK), IPv4/IPv6
- E.DNS.JP : WIDE (TYO, SFO, CDG), IPv4/IPv6
- F.DNS.JP : NII (TYO), IPv4/IPv6
- B.DNS.JP は IPv6 サポート計画中

☆ 更新は 15min 程度で反映
- Incremental zone transfer を採用

JP TLD

☆ 現在、NS レコードは５つ
- そのうち４つは IPv6 対応
- 5 NS + 5 A + 4 AAAA = 272 bytes
- 5 NS + 5 A + 5 AAAA = 300 bytes
- 問合わせの名前の最大は 255 bytes
  - 既に 512 byte には収まらない場合も
  - 幾つかのアドレス情報が入らない
- もし名前が 195 bytes 以下なら
  - 5 (NS/A/AAAA) は全部 OK
- 従って、EDNS0 は重要

Root サーバの IPv6 対応

☆ 長〜〜〜い議論
- Root DNS 運用担当者間
- ICANN RSSAC/SSAC

☆ いろいろと難しい点が
- Root zone 中の AAAA record よりは難しい

☆ 幾つかの危惧
- システムが動かなくなるのでは?
  - これは絶対避けたい
- パケット長に関する問題
  - EDNS0 を用いれば大丈夫
  - EDNS0 付きパケットを捨てちゃう箱は？
- IPv6 の接続性や安定性
- 応答メッセージでのアドレスの順序
  - A record を優先すべき？

Root サーバの IPv6 対応

☆ Firewall がパケットを廃棄する可能性
- 古い Cisco PIX
  - UDP による応答が 512byte より大きい場合
  - Firmware の更新と設定の変更が必要
- 幾つかの zone が解決できなくなる危惧が
  - EDNS0 を使えるようにしたとき
  - (EDNS0 なしなら OK)
- これは Root 相手の問題だけではない

☆ この場合管理者に相談してください
- 特に以下の片方しか返事がこない場合

```
% dig @a.gtld-servers.net com ns              ; 509 bytes
% dig @a.gtld-servers.net com ns +bufsiz=1024   ; 520 bytes
```

Root サーバの IPv6 対応

☆ **EDNS0 の普及度**
- 半分以上の問合わせが EDNS0 対応
  - 99% という場合も

☆ **"prefer-glue a" は必要か？**
- A record はとりあえず全部入る
- IPv6 による問合わせは EDNS0 対応を期待
- で、空きがある限り AAAA record を詰める
  - B/D/F/G/H/K/L/M はそうしている
- ばらばらの方がいいのかしら？
- そろえた方がいいのかしら？

Root サーバの IPv6 対応の経緯

☆ **SSAC が発行した文書（2007 年 1 月）**

Accommodating IP Version 6 Address Resource Records
for the Root of Domain Name System

http://www.icann.org/committees/security/sac018.pdf

☆ **ICANN Board が承認**
- 2007 年 12 月 31 日に通知
- "AAAA records to be added for root servers"
- 2008 年 2 月 4 日に実施の旨
- 6 つの Root Serves が対応を表明
  - A/F/H/J/K/M
  - Priming の応答は今は 615 bytes

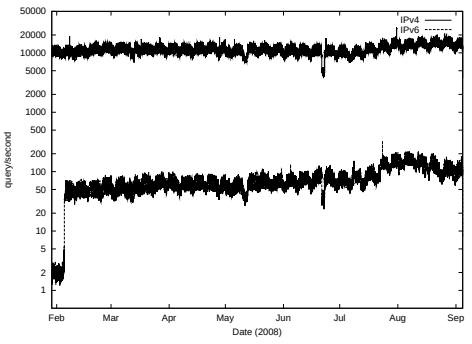
```
% dig . ns +norec +bufsize=1024
```

Root サーバの IPv6 対応の経緯

☆ **M-Root では**
- **別なサーバでテスト**
  - **４年以上前から**
- **運用中のサーバへの移行**
  - 2008 年 1 月中旬に完了
- Tokyo(3), San Francisco, Paris で Anycast
  - Seoul では IPv6 がサポートされていない
- IX での IPv6 対応を要請

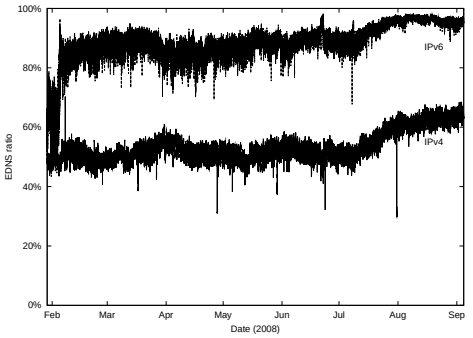
Root サーバの IPv6 対応

☆ パケット数



Root サーバの IPv6 対応

☆ EDNS 比率



Root サーバの IPv6 対応

☆ 現在の状況
- ちゃんと稼働している
- 特に問題は報告されていない
- IPv6 による問合わせは 1% 程度
  - ちょっとは増えたかな？

☆ M-Root での IPv6 による問合わせ
- 半分は Paris のサーバで処理
- 残りは殆んどが東京
- US での IPv6 が弱いということではない
  - Tier-1 ISP との peering 問題

IPv6 対応に関する別な懸念

☆ 多くの **DNS** サーバは **IPv4** のみ対応
- つまり、AAAA レコードがない状態

☆ もし **cache** に **A record** しかないとすると
- これは AAAA がないことを意味する？
- それとも、AAAA にはなにも言及していない？

☆ この解釈は実装依存
- 余分な問合わせが発生する可能性がある
- Root への問合わせに繋がることも
  - ある条件下での BIND9 など

その他の懸念

☆ **Kaminsky's Attack**
- Cache Posisoning
- Birthday attack を利用
  - DNS の ID は 16bit しかない
  - 数打てば一致する可能性がある
  - 本物の応答より早く当れば、偽情報
- Authoritative only サーバでは関係ない
  - Root/TLD/.. は大丈夫
- DNS の recursive サーバでの対策
  - ソフトウェアを最新版にする
  - Source UDP port もランダムに
  - 確率を 1/100 以下に
- 本質的な対応は DNSSEC

まとめ

☆ **DNS の運用上の問題点について概説**
- IPv6 アドレス表現
- Anycasting
- パケット長の問題

☆ **現在の TLD/Root の状況を概説**
- 特に IPv6 対応について