今年もやります！
ランチのおともにDNS

Internet Week 2008

民田雅人・森下泰宏
株式会社日本レジストリサービス

本日のランチメニュー

- DNSのIPv4/IPv6合わせ盛　1034円
  - DNSのIPv6対応
- DNSの512バイト包み焼き　1035円
  - 512バイトの壁の由来とEDNS0
- DNSよろず相談　時価

DNSのIPv6対応

DNSのIPv6対応は2つある

- DNS通信のIPv6対応
  - DNSの問い合わせ、応答のやりとりにIPv6の通信を使う
- DNSコンテンツ(ゾーンデータ)のIPv6対応
  - IPv6アドレス(AAAAリソースレコード(RR)) を登録する
  - IPv6の逆引きを登録する
- 2つは独立の事象
  - IPv4の通信を使ってIPv4アドレス(A RR)を検索
  - IPv4の通信を使ってIPv6アドレス(AAAA RR)を検索
  - IPv6の通信を使ってIPv4アドレス(A RR)を検索
  - IPv6の通信を使ってIPv6アドレス(AAAA RR)を検索

DNS通信のIPv6対応

- DNSサーバの実装が、IPv6での通信に対応しているかどうか
  - 比較的新しい実装は、ほとんどがIPv6での通信に対応
- 権威DNSサーバの実装
  - NSD、PowerDNS、BIND 9、BIND 8(8.4系)、etc…
- キャッシュDNSサーバの実装
  - Unbound、PowerDNS recursor、BIND 9、BIND 8(8.4系)、etc…

BIND 9のIPv6関連の設定項目(抜粋)

- IPv6アドレスを直接記述できるもの
  ⇒ IPv4アドレスと併記できるもの
  - allow-query, allow-recursion, allow-query-cache
  - allow-notify, allow-transfer
  - match-destinations, match-clients
  - zone文内のmasters
- IPv6専用のオプションがあるもの
  ⇒ IPv4とは別に設定するもの
  - listen-on-v6　　listen-onのIPv6版
  - notify-source-v6　　notify-sourceのIPv6版

DNSコンテンツのIPv6対応

- 正引き　ドメイン名 ⇒ IPv6アドレス
  - AAAA RRを使う

```
www.example.jp.    IN AAAA   2001:DB8::1
```

- 逆引き　IPv6アドレス ⇒ ドメイン名
  - PTR RRを使う点はIPv4と同じ
  - 4bit単位で区切り、逆順に並べ最後に“ip6.arpa”
  - 2001:DB8::1
    ⇒ 2001:0DB8:0000:0000:0000:0000:0000:0001

```
1.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.8.B.D.0.1.0.0.2.ip6.arpa.    IN PTR www.example.jp.  (1行で記述)
```

  - 桁数が多いので記述ミスに注意

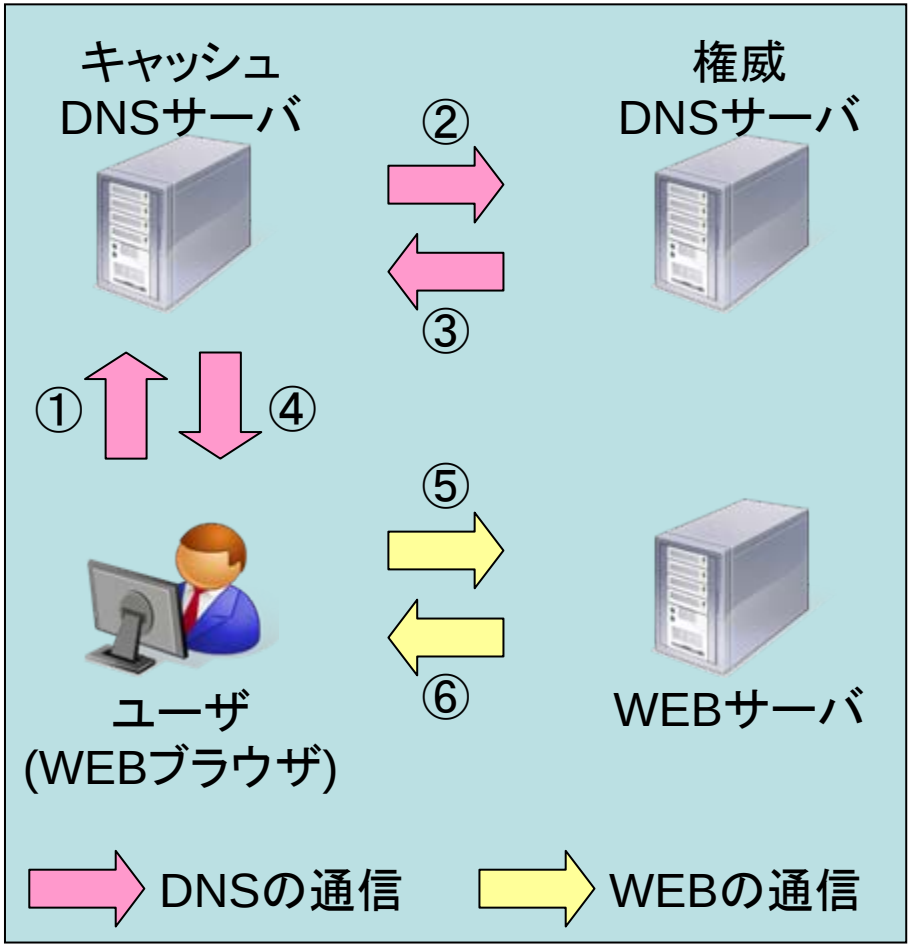
IPv6の逆引き記述のTIPS

- 人手で記述する場合、ミスを減らすため
  digコマンドの-xオプションを活用するのがベター
  - digコマンドは、DNS登録の有る無しに関わらずQUESTION SECTION をコメントで表示

```
$ dig -x 2001:db8::1
.....
;; QUESTION SECTION:
;1.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.8.b.d.0.1.0.0.2.ip6.arpa.  IN PTR
.....
```

  - この行をコピー&ペーストして活用

IPv4/IPv6の混沌とした世界



DNSはIPv4のみでも、WEBのAAAA RRがあれば、ユーザはWEBにIPv6でアクセス可(Google, 2ch等のIPv6)

- 関係する全通信がIPv4/IPv6両対応とは限らない

①~⑥のIPv6/IPv4環境に、なんらかの問題がある

- ユーザがWEBアクセスに時間がかかる、あるいはアクセス不能になることもある

トラブルシューティングを難しくする可能性がある

DNSのIPv6対応 まとめ

- DNSでは、通信とゾーンデータのIPv6対応がある
  - 最近の実装であれば、いずれも対応済
- 正引きの登録はAAAA RRを使う
- 逆引きは4bitづつで区切り最後に「ip6.arpa.」
  - 桁数が多いので記述ミスに注意
- WEBサーバーとPCはIPv6で通信していても、DNSはIPv4で通信していることもある
  - PC(WEBブラウザ) ⇔ キャッシュDNSサーバ
  - キャッシュDNSサーバ ⇔ 権威DNSサーバ

512バイトの壁の由来とEDNS0

DNSに存在する「512バイトの壁」

- UDPによる問合せ・応答の上限値
- ~~1035円~~ RFC 1035で定義
  - Page 32より抜粋

> 4.2.1. UDP usage
>
> Messages sent using UDP user server port 53 (decimal).
>
> Messages carried by UDP are **restricted to 512 bytes** (not counting the IP or UDP headers). Longer messages are truncated and the TC bit is set in the header.

なぜ「512バイトの壁」が存在するのか?

- Paul Mockapetrisさんが来日された際に聞いてみました
- インターネット建設当時(1980年代) の状況に従ったもの
- その当時のインターネットの信頼性を考慮
  - 「576バイト = 512バイトのデータ + 64バイトの上位層ヘッダ」までは、最低限、一度に受け取れるように(RFC 791)
- 実はこのRFC 791における定義は今でも有効
- 「512バイト」「64バイト」…ともに「2のべき乗」
  - コンピュータが扱いやすい
  - 処理の負荷が軽くて済む
- DNSでは問合せ・応答とも、できる限り1パケットでのやりとりで済ませるようにしたかった

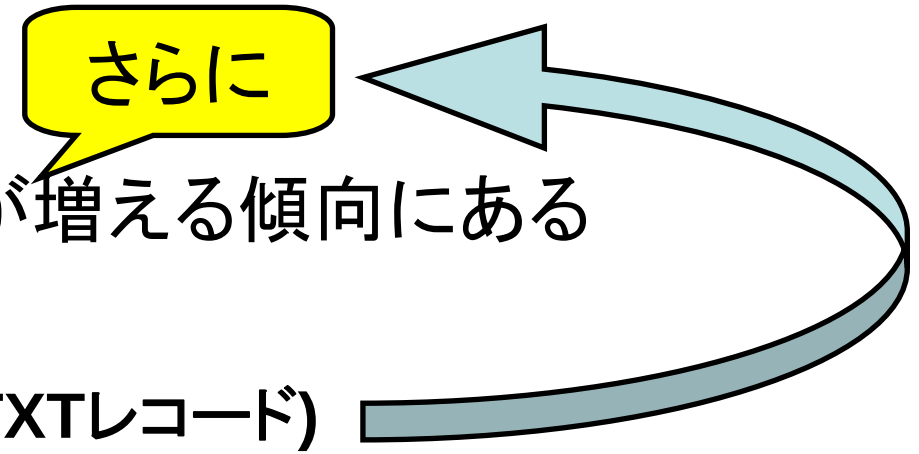
参考: RFC 791(Page 13より抜粋)

Total Length: 16 bits

Total Length is the length of the datagram, measured in octets, including internet header and data. This field allows the length of a datagram to be up to 65,535 octets. Such long datagrams are impractical for most hosts and networks. All hosts must be prepared to accept datagrams of up to 576 octets (whether they arrive whole or in fragments). It is recommended that hosts only send datagrams larger than 576 octets if they have assurance that the destination is prepared to accept the larger datagrams.

The number 576 is selected to allow a reasonable sized data block to be transmitted in addition to the required header information. For example, this size allows a data block of 512 octets plus 64 header octets to fit in a datagram. The maximal internet header is 60 octets, and a typical internet header is 20 octets, allowing a margin for headers of higher level protocols.

最近のDNS



- DNS応答のサイズが増える傾向にある
- 例:
  - **spam対策 = SPF(TXTレコード)**
  - IPv6への対応
  - DNSSECの導入、など
- 「512バイトの壁」の存在が顕在化
  - 512バイトより大きいTXTレコード(SPF)は、既にかなりある
- どうする?
  - 可能であれば、512バイトよりもデータを小さくする(おすすめ)
  - 「512バイトの壁」を越えるための方策を考える

「512バイトの壁」を越える方法

①TCPでデータをやりとりする

②EDNS0(イーディエヌエスゼロ)を使う

①TCPでデータをやりとりする

- 65,536バイトまでのデータを取り扱える
- しかし、現在のDNSでは「最初にUDPで試してから」でないと、TCPで問い合わせてはいけないことになっている (RFC 1123: Page 75より抜粋)
- サーバ負荷の問題など

> 6.1.3.2 Transport Protocols
>
> DNS resolvers and recursive servers MUST support UDP, and SHOULD support TCP, for sending (non-zone-transfer) queries. Specifically, a DNS resolver or server that is sending a non-zone-transfer query MUST send a UDP query first. If the Answer section of the response is truncated and if the requester supports TCP, it SHOULD try the query again using TCP.

②EDNS0を使う

EDNS0が生まれた背景

- ネットワークの信頼性向上・コンピュータの処理能力向上
- データの分割・再構成処理をしても、十分なパフォーマンス・信頼性が得られるようになってきた(はず)

⇒ DNSプロトコルを拡張し、UDPでも大きなデータを取り扱えるようにしよう

⇒ EDNS0の誕生

EDNS0の概要

- RFC 2671で定義
- 「OPT」という「擬似RR」を使用
  - 通信の際にのみ現れる(データファイルには記述しない)
- 実際の使われ方
  - 最初にOPT RRつきパケットを権威DNSサーバに送信
  - 正しい応答 ⇒ 以降の通信にはすべてEDNS0を使用
  - エラー応答 ⇒ 従来のDNSプロトコルで通信で通信
- 最近の主なサーバ実装はEDNS0をサポートしている
  - BIND 9
  - Microsoft DNS Service (Windows 2003 Server以降に付属のもの)
  - Nominum ANS / CNS
  - NSD / Unbound
  - PowerDNS / PowerDNS recursor
- IPv6 / DNSSECサポートの際にはEDNS0が必須(RFC 3226)

TCPフォールバックとEDNS0の比較

TCPフォールバック

```
% dig ***.com txt
;; Truncated, retrying in TCP mode.
    (TCPフォールバック)
(途中略)
;; Query time: 179 msec
;; SERVER: ***.***.***.***#53(***.***.***.***)
;; WHEN: Mon Jun  2 20:31:20 2008
;; MSG SIZE  rcvd: 668 ← 応答の大きさ
```

EDNS0

```
% dig ***.com txt +bufsize=4096
    受信可能な大きさを指定(EDNS0)
(途中略)
;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
    EDNS0有効↑
(途中略)
;; Query time: 192 msec
;; SERVER: ***.***.***.***#53(***.***.***.***)
;; WHEN: Tue Jun 10 17:49:47 2008
;; MSG SIZE  rcvd: 679 ← 応答の大きさ
```

- TCPフォールバックではUDPの応答確認後にTCPで問合せる
- EDNS0では一度目のUDP問合せのみで正しい応答が得られる
- EDNS0ではOPT RRの分、応答がさらに大きくなることに注意

EDNS0(TCPフォールバック)運用上のTIPS

- 特に、spam対策として**SPFを利用(設定)している**場合
- 使用しているネットワーク機器やソフトウェアがEDNS0に対応しているかどうか確認しておくこと
  - EDNS0をうまく処理できないファイアウォール
  - EDNS0をうまく処理できないブロードバンドルータ、等
- DNSサーバに対しファイアウォールを設定する際、TCPでの問合せについても忘れずに考慮すること
  - UDP/53に加え、TCP/53も忘れずに許可しておくこと
  - 「TCP/53に対し何も返さない」設定はしないこと
- 大きなDNSパケットは、DNS Amp攻撃の際の元ネタとして使われる可能性があることに注意すること

質問 & DNSよろず相談