Everything on DNS

- DNS
  - 「地球上で、唯一、成功している分散データベース」
- ヒトには、困ったときにはDNSに逃げ込む習性がある。
  - janog2@KDD新宿ビルで、BGPの経路情報をDNSで認証するというI-Dの紹介があった。
  - そもそもclass HSからしてそーぢゃん!
  - UBE blacklist然り
  - X25、ISDN(RFC 1183)、GPOS(RFC 1712)、…然り

512 octetの壁

- RFC 1035
  - 2.3.4 size limits
    - UDP messagesは512 octets or lessと書いてある。
- RFC 760
  - 2.1 Internet Header Format
    - 512 octetはreasonable sized data blockと書いてある。

512 octetの壁に挑むplayerたち

- AAAA
- PTR for v6 address
- SPF
- Punycode
  - この辺は、ゾーンデータの見た目は派手だが、実は大したことない。

512 octetの壁に挑むplayerたち(続き)

- キpee-!イのようなround robin
  - wwwという名前にAとAAAAを11個ずつ設定してANYを索いてみたら、641 > 512 octetになった。
  - 実は10個のときに興味深い挙動が観測された(後述)。
- Active Directoryがらみの SRV RRを索いたら512 octetを越えた、という事例があるらしい。
  - 何で外部にそんな物を索きに行ったのかは不明。
- ある種のウィルスが索く名前は512 octetを越えるらしい。
  - こいつらは、まぁ特殊なケースでしょう…が、
  - ロバストなシステムとしては、受けて立たなければ。

512 octetの壁に挑むplayerたち(続き)

- DNSSECを使うと、1000 octetを越えるらしい。
  - deployするの? は、今日は禁句☺
- DomainKeysは512 octetに納まるように配慮してはいるらしいが、鍵長、他のRRとの組み合わせによっては…?

additional sectionを切り捨てる、という荒業

- round robinをいっぱい設定して、512 octetを越えさせる実験。

```
$ORIGIN example1.jp.
@  IN SOA …
          :
www   A      192.168.1.1
      AAAA   2001:db8::192:168:1:1
          :
      A      192.168.1.11
      AAAA   2001:db8::192:168:1:11
```

11組設定したら越えた

additional sectionを切り捨てる、という荒業(続き)

- 9組のとき、509 octet
- 10組のとき、509 octet??
- 11組のとき、641 octet

```
; <<>> DiG 9.3.2 <<>> @localhost example1.jp ANY
; (2 servers found)
;; global options:  printcmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id:
   46294
;; flags: qr aa rd; QUERY: 1, ANSWER: 20,
   AUTHORITY: 0, ADDITIONAL: 2
```

!

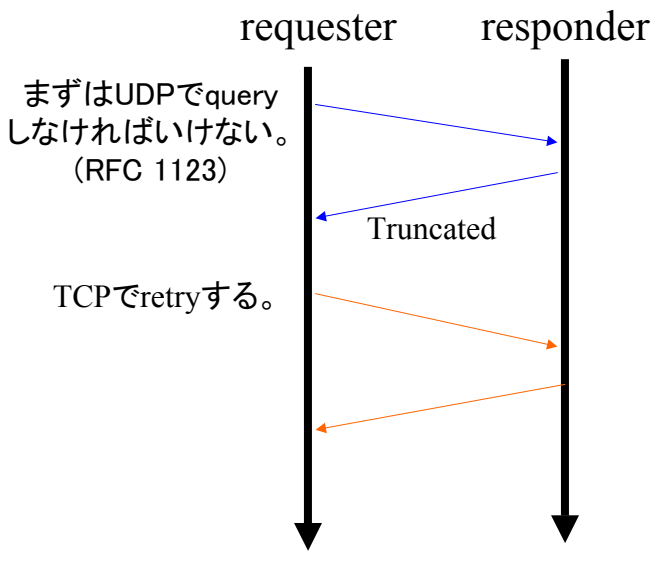
additional sectionを切り捨てる、という荒業(続き)

- BIND 9.3.2とNSD 2.3.6でAUTHORITY: 0を返す挙動が観測された。
- 切り捨てて512 octetに収まるなら、切り捨てるらしい。
- RFC 1035
  - 4.1 Format
    - Answer section,Authority section,Additional sectionは possibilly emptyと書いてある。
- ということは、Additional sectionにAAAAが入ることは、今は気にしなくていいみたい。

512 octetの壁を越えてみよう

- TCPにfall backする
  - UDPでqueryして、Truncatedな応答が返ってきたら、TCPでretryする。
  - DNS+TCP=ゾーン転送 ではない。
  - RFC 1035からある古典的手法。
- EDNS0でいってみる
  - 上限が512 octetより増えるだけで、通知された上限を超えれば、やっぱりTCPにfall backする。
  - 自分だけでなく相手も対応していないとダメ。
    - 相手がEDNS0非対応ならTCPにfall backする(か、あきらめる)。
  - BINDだと8.3から対応。

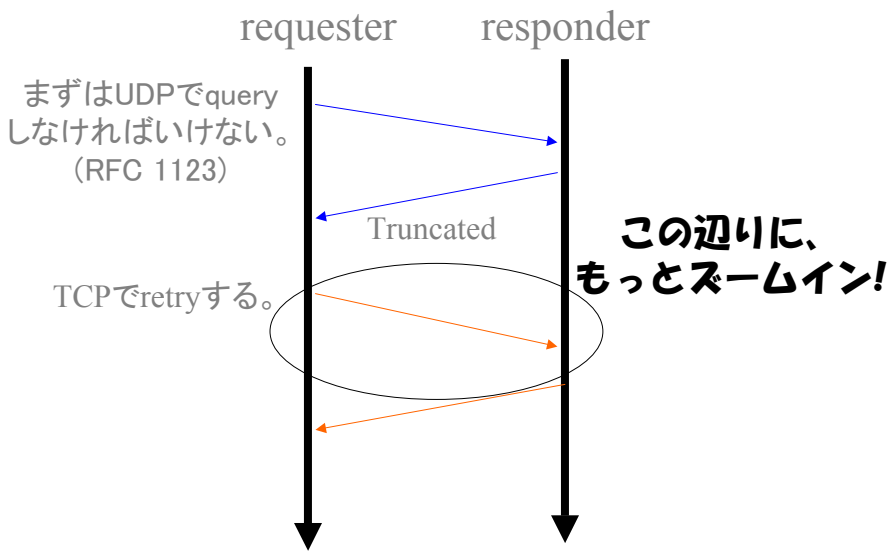
TCPへのfall backにズームイン!



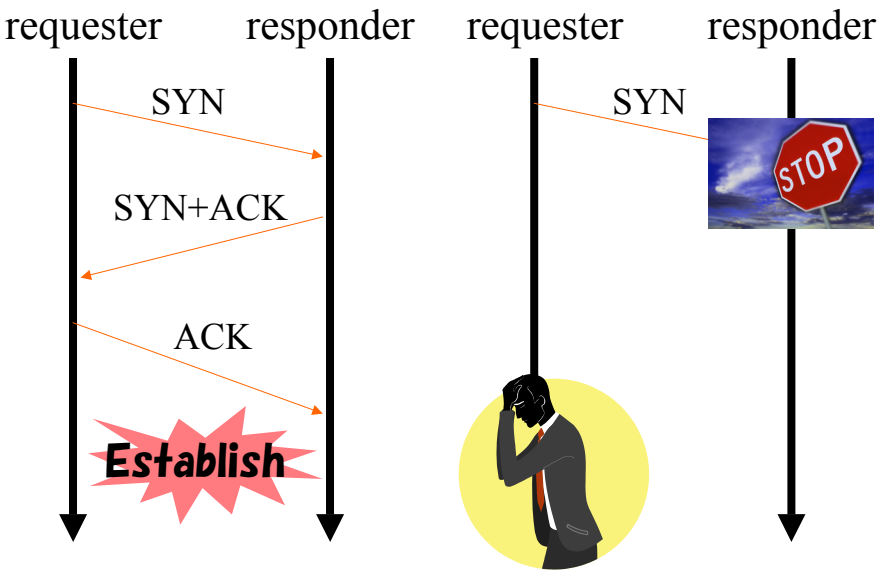
EDNS0にズームイン!

- requesterはadditional sectionに、自分が受けられるUDPのサイズなどを入れたpseudo-RRを突っ込んでqueryを投げる。
- 対応していないresponderはNOTIMPL、FORMERR、SERVFAILを返すだろう。
  - そうしたら、EDNS0なしでretry。
  - でもBINDのstubレゾルバはあきらめる。
- responderがEDNS0対応なら平然と応答する。
  - パケットのサイズの上限はrequesterの要求に沿う。

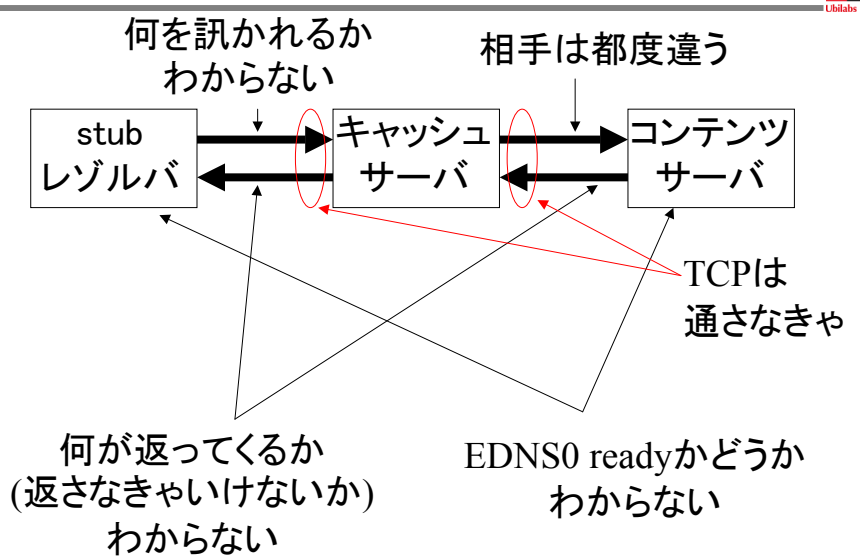
もう一度、TCPへのfall backにズームイン!



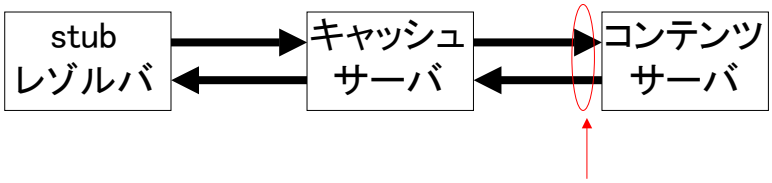
もう1度、TCPへのfall backにズームイン!(続き)



TCPが行く

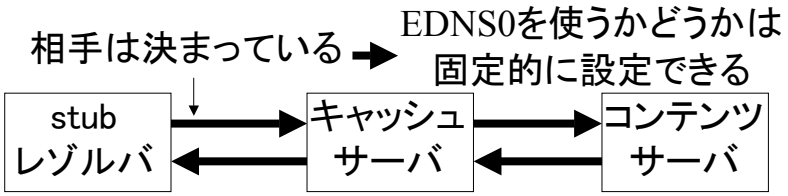


TCPが行く、か?



- オレは512 octet以上は絶対返さない、と断言できれは、TCPは通らないだろう。
- でも、それって管理できますか?
- RFC 1123はSHOULD NOT refuseと言っている。

TCPが行く、か?(続き)



汝、むやみにTCPを落とすなかれ

- DNS+TCP=ゾーン転送 ではない。
- 世の中がみんなEDNS0 readyではない。
- RFC 1123はSHOULD NOT refuseと言っている。
  - RFC 1035(1987)
  - RFC 1123(1989)
  - 今年(2006)