Making do with what we've got: Using PMTUD for a higher DNS responsiveness

Victor Boteanu, Hanieh Bagheri

February 8, 2013

Introduction

- Classic DNS:
  - Normally uses UDP
  - A few truncated responses ->TCP
- Emergence of EDNS0 and DNSSEC
  - Bigger responses: RRset + signature
  - Capability of using UDP for responses >512 bytes
  - fragmentation instead of truncation

Fragmentation in IPv6

- only done on end-to-end hosts
- Path MTU Discovery (PMTUD)
  - finding the smallest MTU in the path
  - ICMPv6 Packet Too Big (PTB): MTU + the trimmed part of the original message

Research question

Would it be feasible to utilize ICMPv6 PTB messages to increase a name server response deliverability?

What strategies can be applied and what effects and risks would they have?

Previous Research

- Maikel de Boer and Jeffrey Bosma:
  - IPv6 path MTU black hole discovery
- Gijs van den Broek et. al.:
  - Monitoring real-world resolvers dealing with fragmented DNS responses
  - Two server-side solutions to prevent fragmentation

Motivation

- Fact 1: About 10% of firewalls filter IPv6 fragments
  - Send responses with the min size guaranteed by all the routers: 1232 bytes
- Fact 2: ICMPv6 PTB message: original message contains (besides headers) the trimmed response
- Fact 3: Name servers are not aware of the PTB messages
  - DNS responses may become lost without informing the name server

Idea

- Our idea:
  - Ability to handle the failed responses due to their big size
  - Send larger responses than 1232, but still less than 1452
- Expected Result:
  - Decrease number of fragments
  - Increase the responsiveness of name servers

Proposed solution 1

- Simply send the response again to the client and set the TC flag
- The client should send the query again using TCP
- Implications:
  - Prevents DNS ID hacking

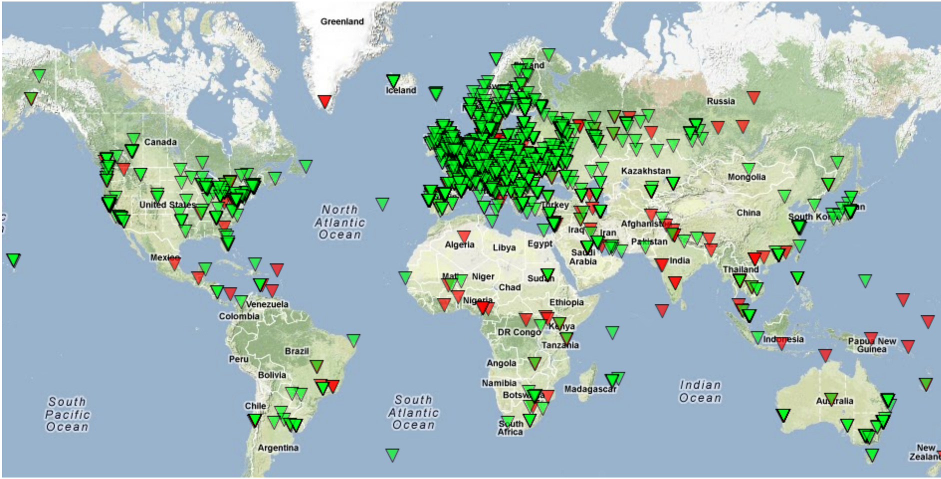
Proposed solutions 2 and 3

- Solution 2 - Use the PTB message payload to resubmit query to the name server
- Solution 3 - Use the PTB message payload to create shorter answers
  - for example omitting the ADDITIONAL section
  - making correction to decrease the value of the EDNS0 option
- Implications:
  - With both solutions, we circumvent ICMPv6 PTB spoofing

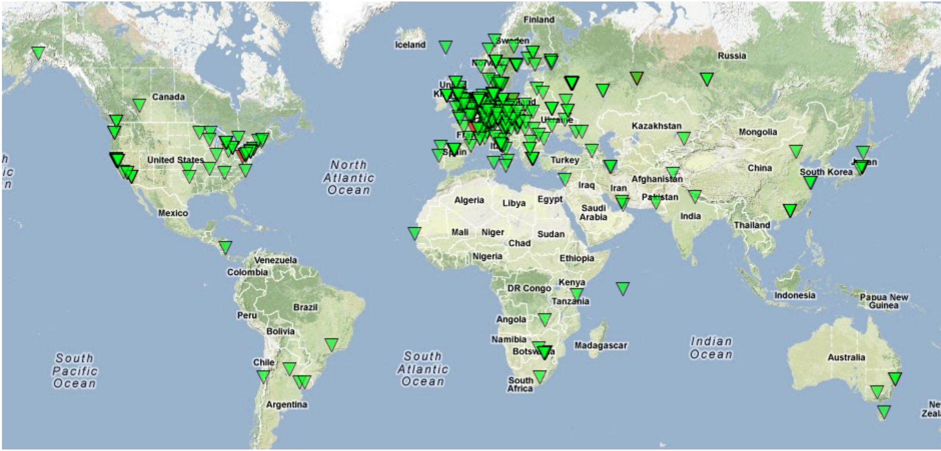
Setup and resources

- NSD 3.2.14 running on NLNOG RING node NLNOG RING
- IPv6 only, no filtering
- RIPE Atlas probes (only IPv6)
- Packet captures provided by SURFnet
- DNS traffic provided by SIDN

Setup and resources



Setup and resources

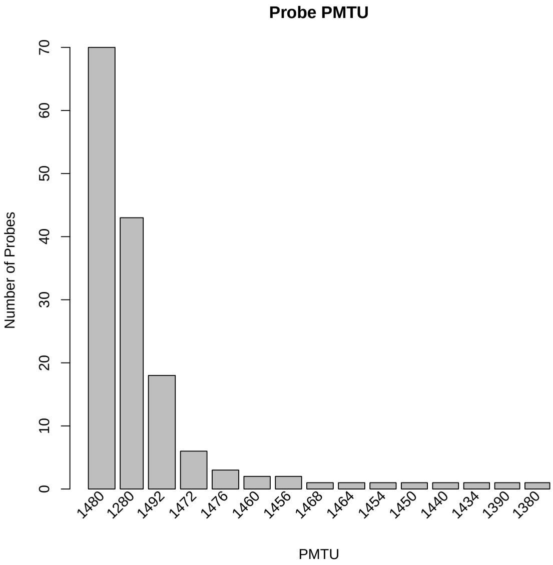


NLnet Labs

Setup and resources

- Number of IPv6 ready probes - around 850
- Available for our experiment - 442
- TXT record >1500 bytes (1560 response size)
- MTU on Ring-server interface set to 1280
- Query TXT record from all Atlas probes

Results and observations



Results and observations

SIDN response sizes (DNSSEC IPv6 only)
Capture time of 2h

Sizes	Number of responses
$\leq 1232 bytes$	99.66%
(1232, 1452] bytes	0.002%
$>$1452 bytes	0.32%

Results and observations

SURFnet response sizes (IPv6)
Capture time of 1h on 7 name servers

Sizes	Number of responses
<1232 bytes	97.77%
(1232,1452] bytes	2.14%
>1452 bytes	0.07%

Results and observations

Atlas probe experiment - PTB and Fragment Reassembly messages

- 34 probes sent back Fragment reassembly messages
- 1 probe sent back PTB message despite MTU of server set to 1280
- This probe only accepted messages of at most 1232 bytes

Results and observations

SIDN and SURFnet - ICMPv6 messages

Type of message	SIDN	SURFnet
Time Exceeded Fragment Reassembly	333(8.1%)*	26(0.06%)
Packet Too Big	43(1%)	16(0.03%)
Administratively prohibited	7991	3624(8.1%)

*out of response sizes >1232 bytes

Implementation of Solution 2

- 427 unique sources
- query TXT record
- Raw socket intercepting packets
- handled 56 problematic sources
- only 5 probes still sent back Fragment Reassembly

Conclusions

- DNSSEC gaining in popularity
- Responses will grow in size
- Firewalls are still configured to filter fragments
- ICMPv6 messages are not used their full potential

Questions