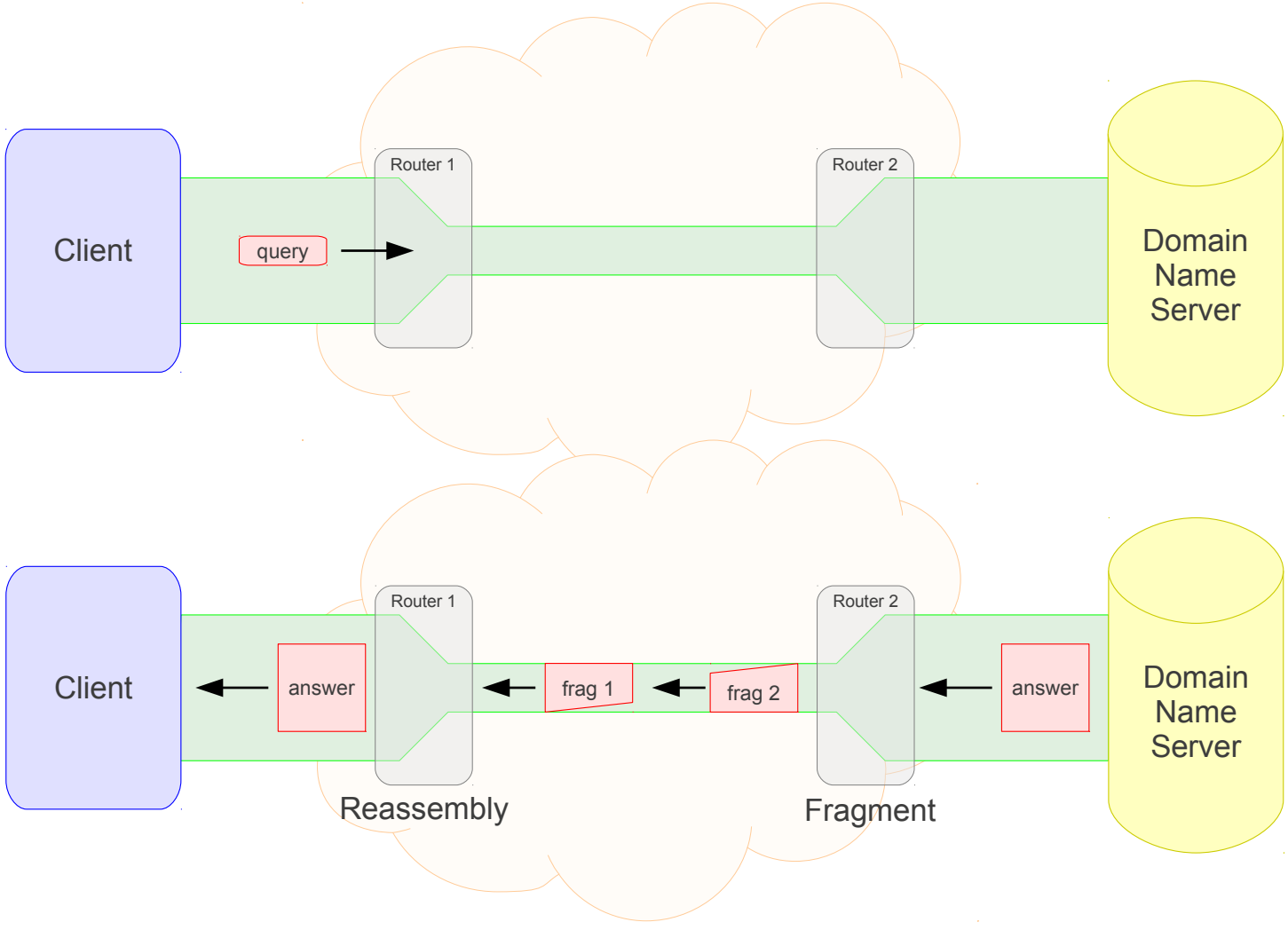
Using PMTUD for DNS
What is this about?

- MTU : Maximum Transmission Unit (on a link)
- PMTU : Maximum Transmission Unit on a Path
  = The smallest MTU on that path.
- PMTUD: Path MTU Discovery

- Follow up of UvA student projects at NLnet Labs:
  - M. de Boer, J. Bosma,
    "Discovering Path MTU black holes on the Internet using RIPE Atlas"
    (July 2012)
- Research performed early this year by UvA Students
  - H. Bagheri, V. Boteanu,
    "Making do with what we've got:
    Using PMTUD for a higher DNS responsiveness" (February 2013)

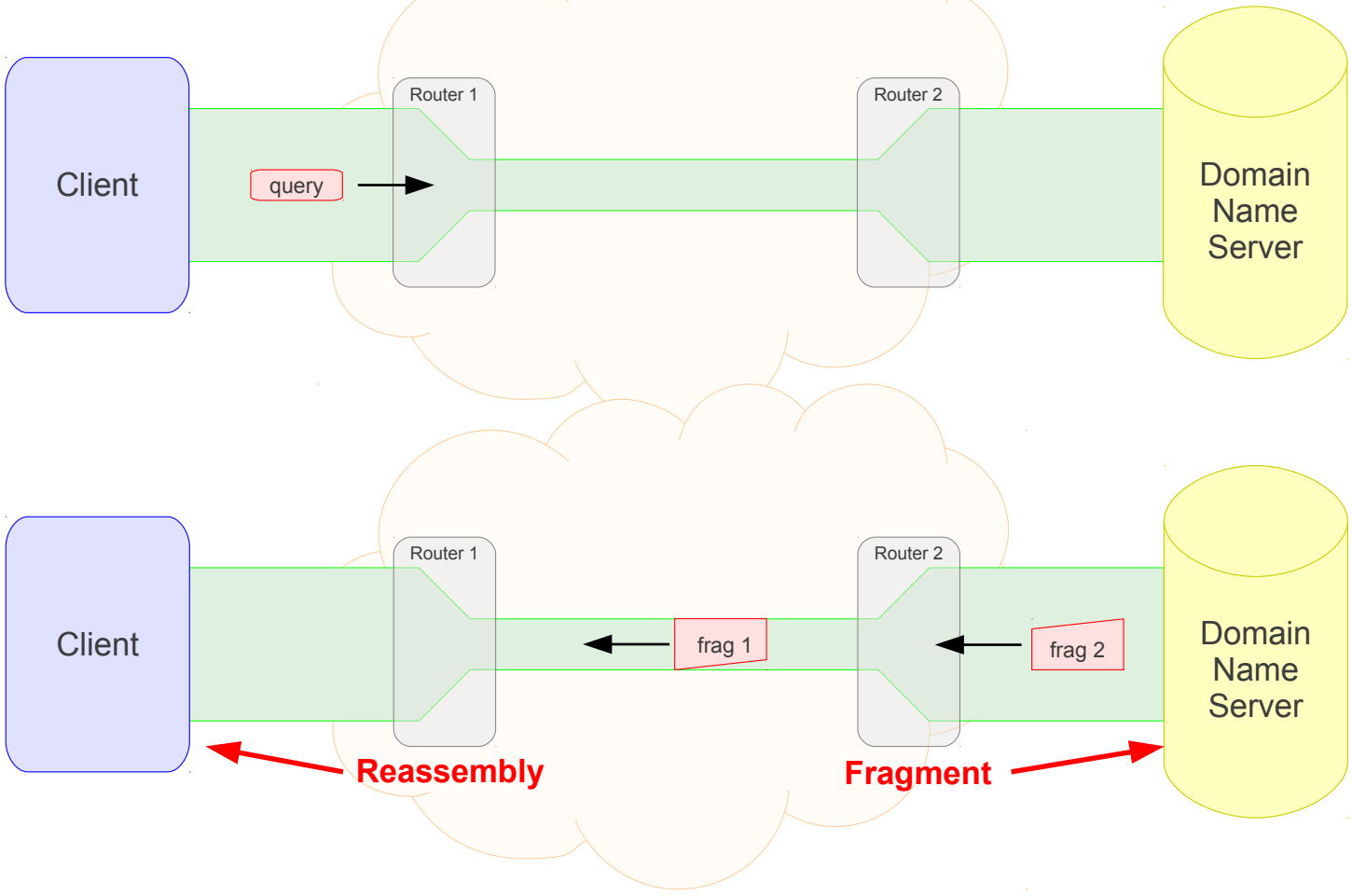
Using PMTUD for DNS
What is this about?

- With IPv4 fragmentation was handled by the network



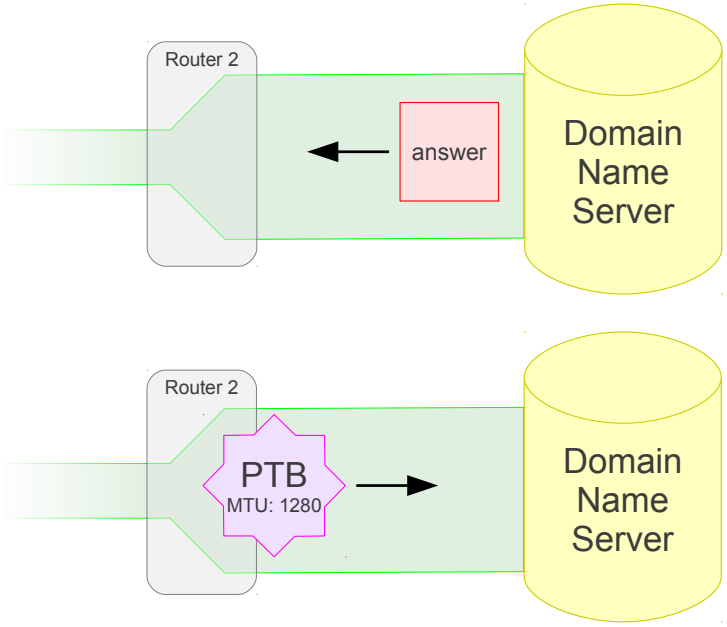
Using PMTUD for DNS
What is this about?

- With IPv4 fragmentation was handled by the network
- With IPv6 only end-points may fragment and reassemble



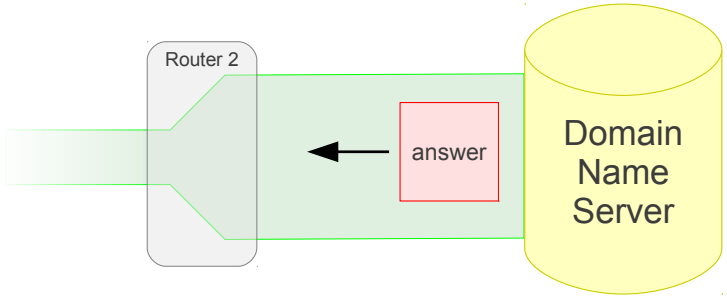
Using PMTUD for DNS
What is this about?

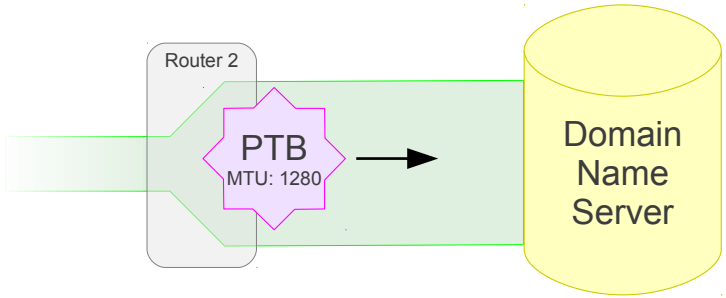
- With IPv6 only end-points may fragment and reassemble
- But currently DNS servers do not handle Packet-Too-Big

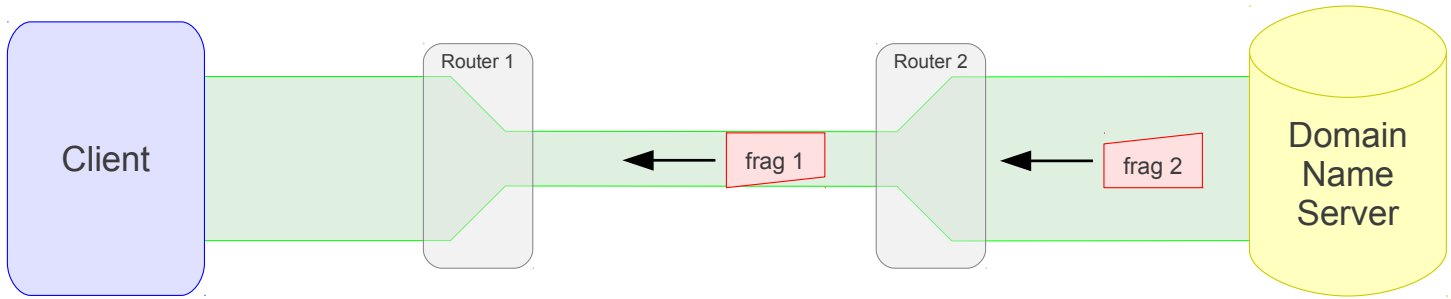


Using PMTUD for DNS
What is this about?

- With IPv6 only end-points may fragment and reassemble
- But currently DNS servers do not handle Packet-Too-Big
- The OS caches PMTU for 10 minutes, or so...
- and requery happens after 5 seconds, or so...

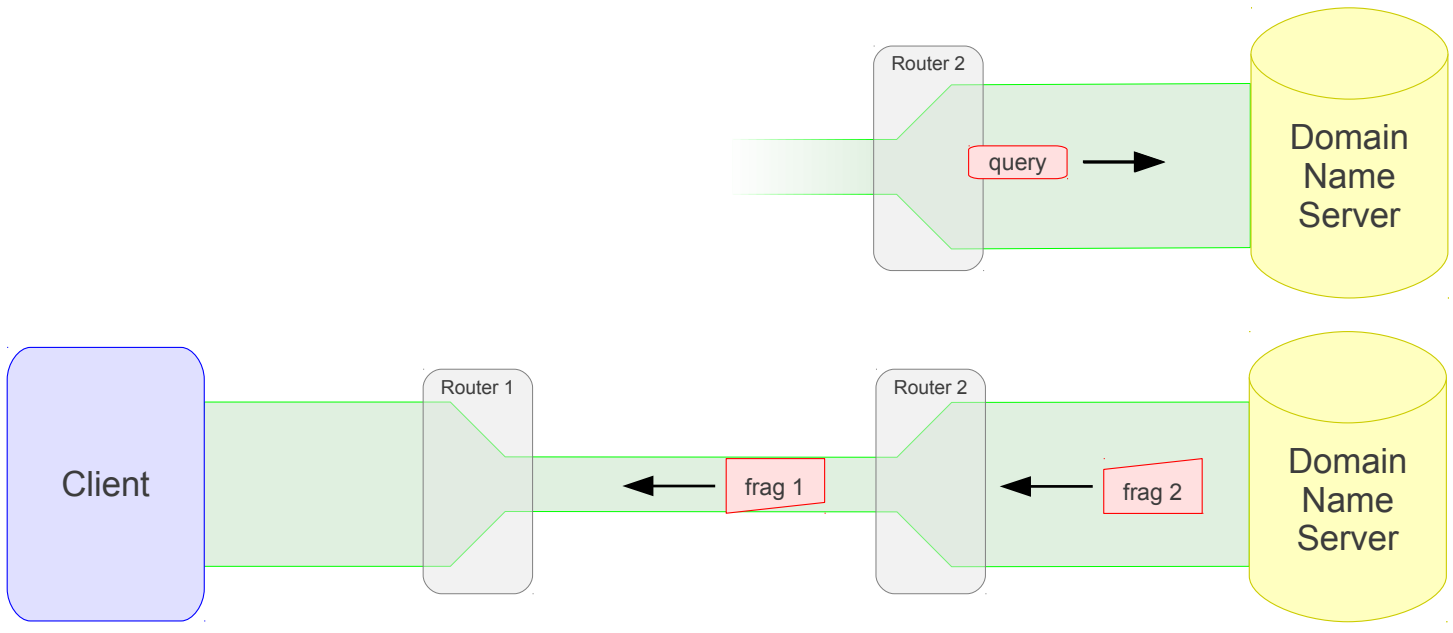






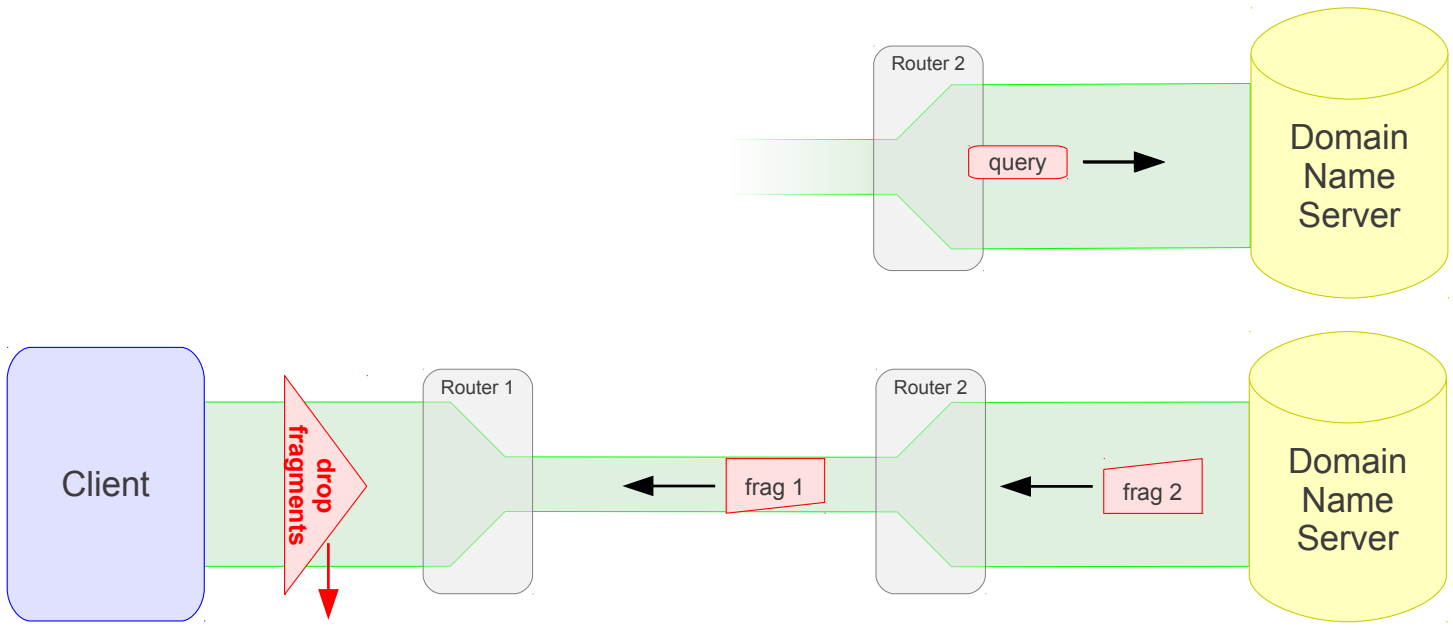
Using PMTUD for DNS
What is this about?

- With IPv6 only end-points may fragment and reassemble
- But currently DNS servers do not handle Packet-Too-Big
- draft-andrews-dnsext-udp-fragmentation-01



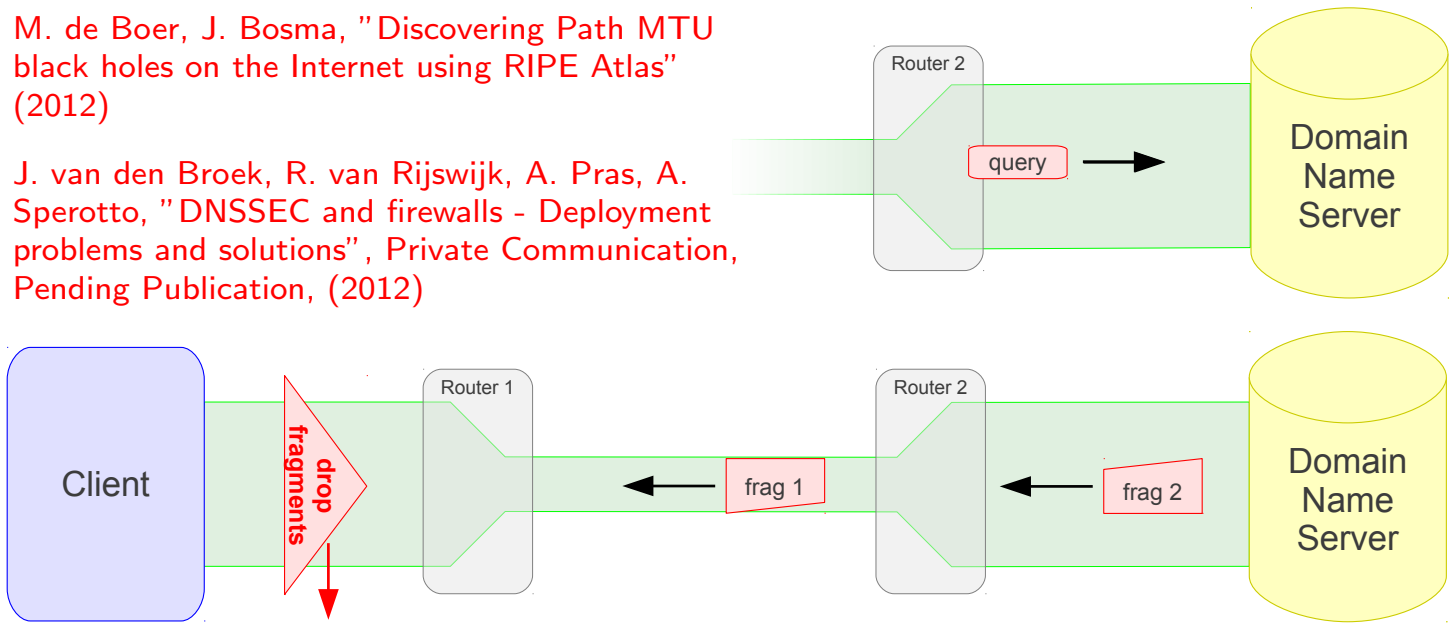
Using PMTUD for DNS
What is this about?

- With IPv6 only end-points may fragment and reassemble
- But currently DNS servers do not handle Packet-Too-Big
- draft-andrews-dnsext-udp-fragmentation-01
- But then messages in size range 1232-1452 *packet size* 1280−1500 will be fragmented too!



Using PMTUD for DNS
What is this about?

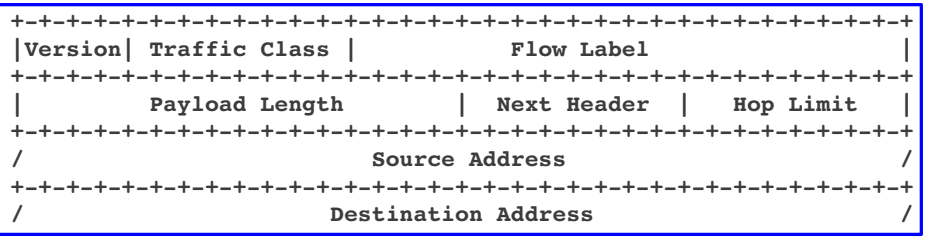
- With IPv6 only end-points may fragment and reassemble
- But currently DNS servers do not handle Packet-Too-Big
- draft-andrews-dnsext-udp-fragmentation-01
- But then messages in size range 1232-1452 $packet\ size\ 1280-1500$ will be fragmented too!
- And ±10% of all end-points/resolvers discard IPv6 fragments!
  - M. de Boer, J. Bosma, "Discovering Path MTU black holes on the Internet using RIPE Atlas" (2012)
  - J. van den Broek, R. van Rijswijk, A. Pras, A. Sperotto, "DNSSEC and firewalls - Deployment problems and solutions", Private Communication, Pending Publication, (2012)



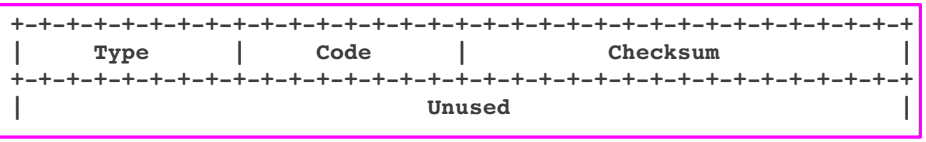
Using PMTUD for DNS
What is this about?

- ICMPv6 Error Messages contain as much of invoking packet as possible without the ICMPv6 packet size exceeding 1280

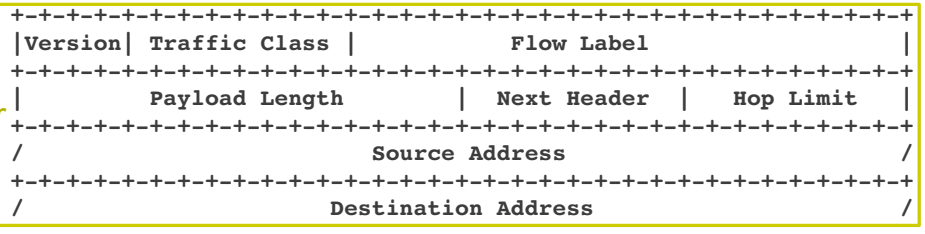
Router IPv6



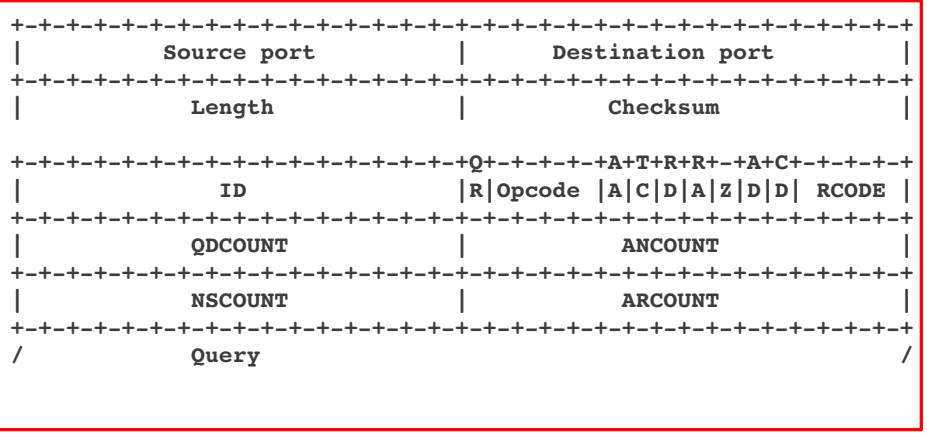
ICMPv6 Error msg



Domain Name Server IPv6



Beginning of Answer

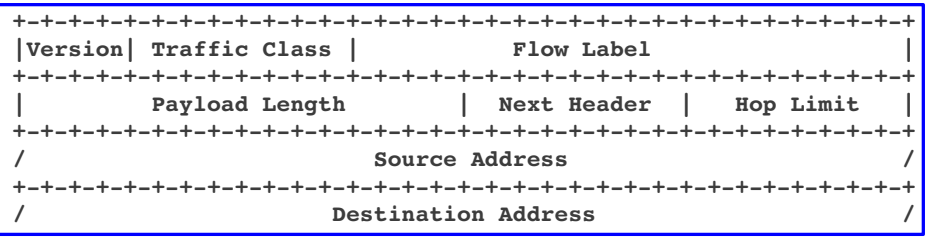


Using PMTUD for DNS
What is this about?

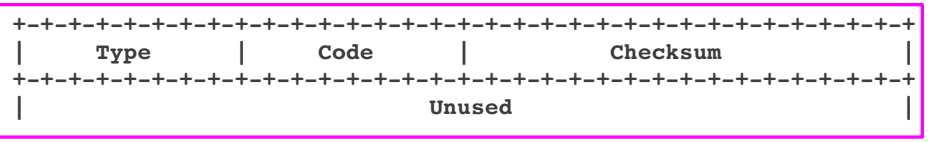
- ICMPv6 Error Messages contain as much of invoking packet as possible without the ICMPv6 packet size exceeding 1280
- Utilizing ICMPv6 PTB messages to send bigger unfragmented answers

  (in the 1232-1452 range)
- Increase DNS responsiveness

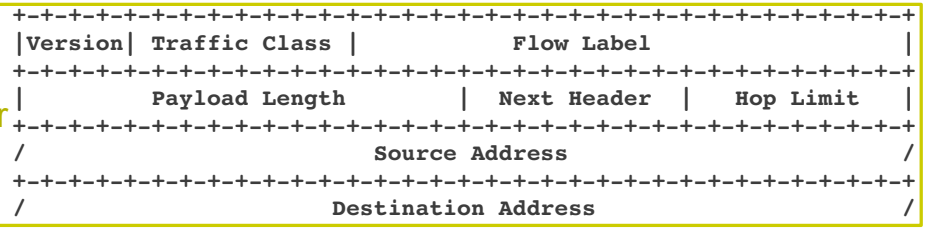
Router IPv6



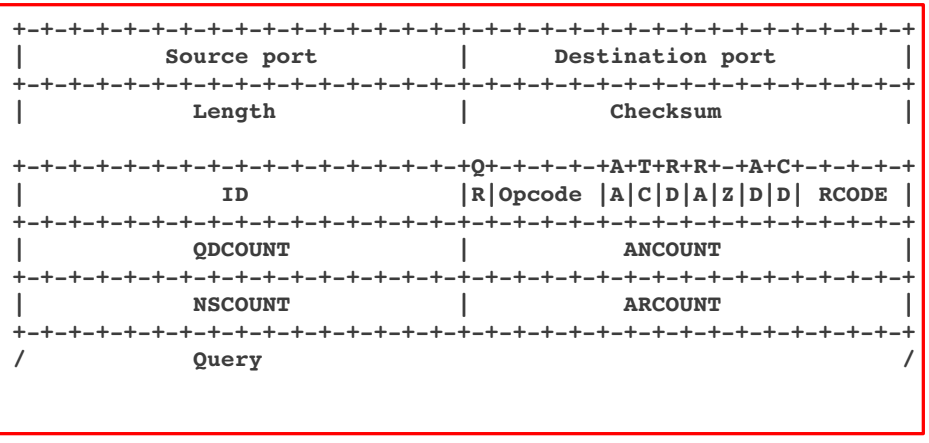
ICMPv6 Error msg



Domain Name Server IPv6



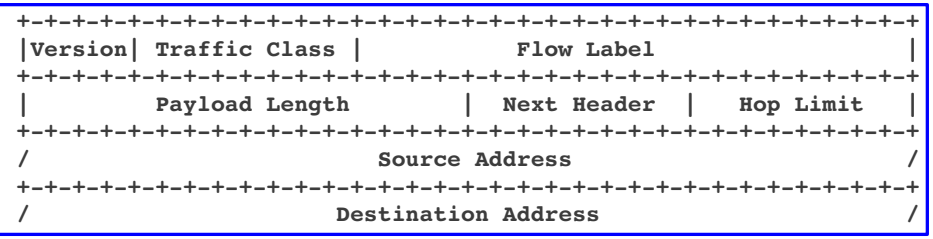
Beginning of Answer



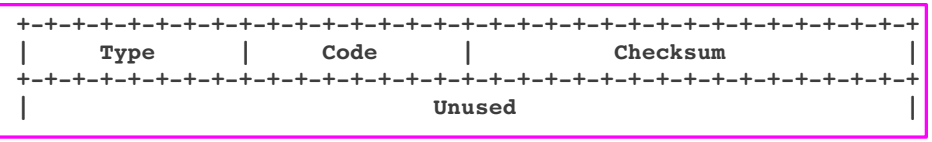
Using PMTUD for DNS Observations

- Bypass BCP38: Anyone can spoof a source address.

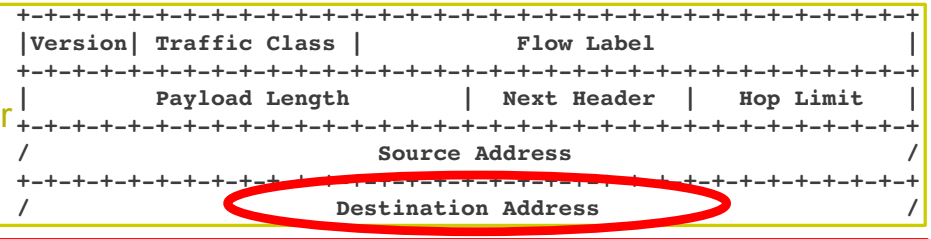
Router IPv6



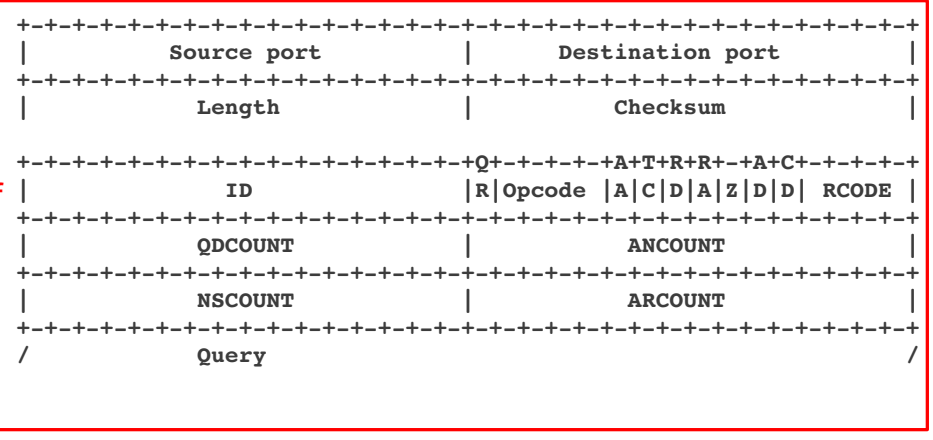
ICMPv6 Error msg



Domain Name Server IPv6



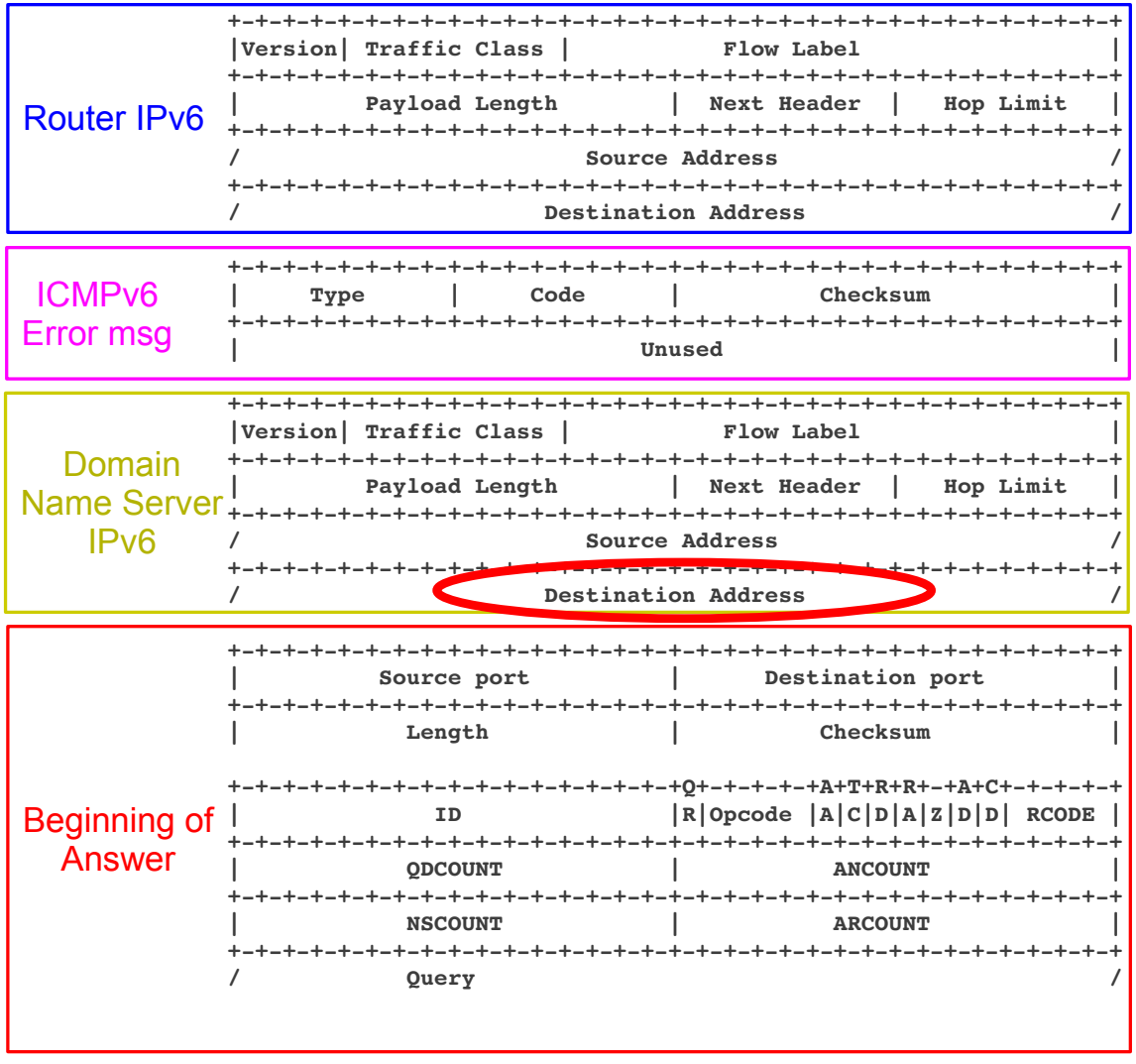
Beginning of Answer



Using PMTUD for DNS Observations

- Bypass BCP38: Anyone can spoof a source address.
- Simply re-inject with TC bit: NO GO! (cache poisoning)
- So re-evaluate query at Domain Name Server

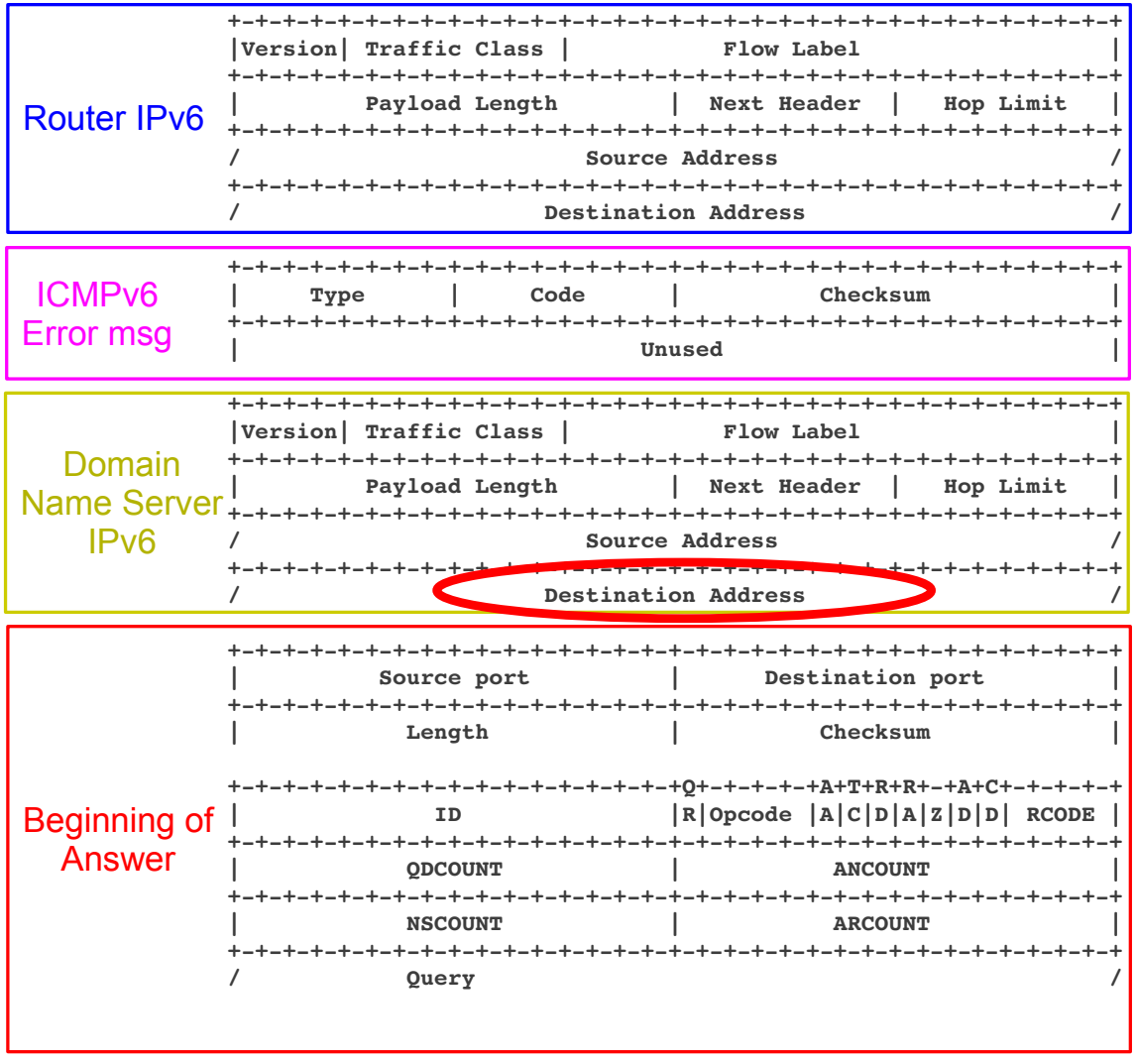
  (or resubmit spoofing the source)



Using PMTUD for DNS Observations

- Bypass BCP38: Anyone can spoof a source address.
- Simply re-inject with TC bit: NO GO! (cache poisoning)
- So re-evaluate query at Domain Name Server

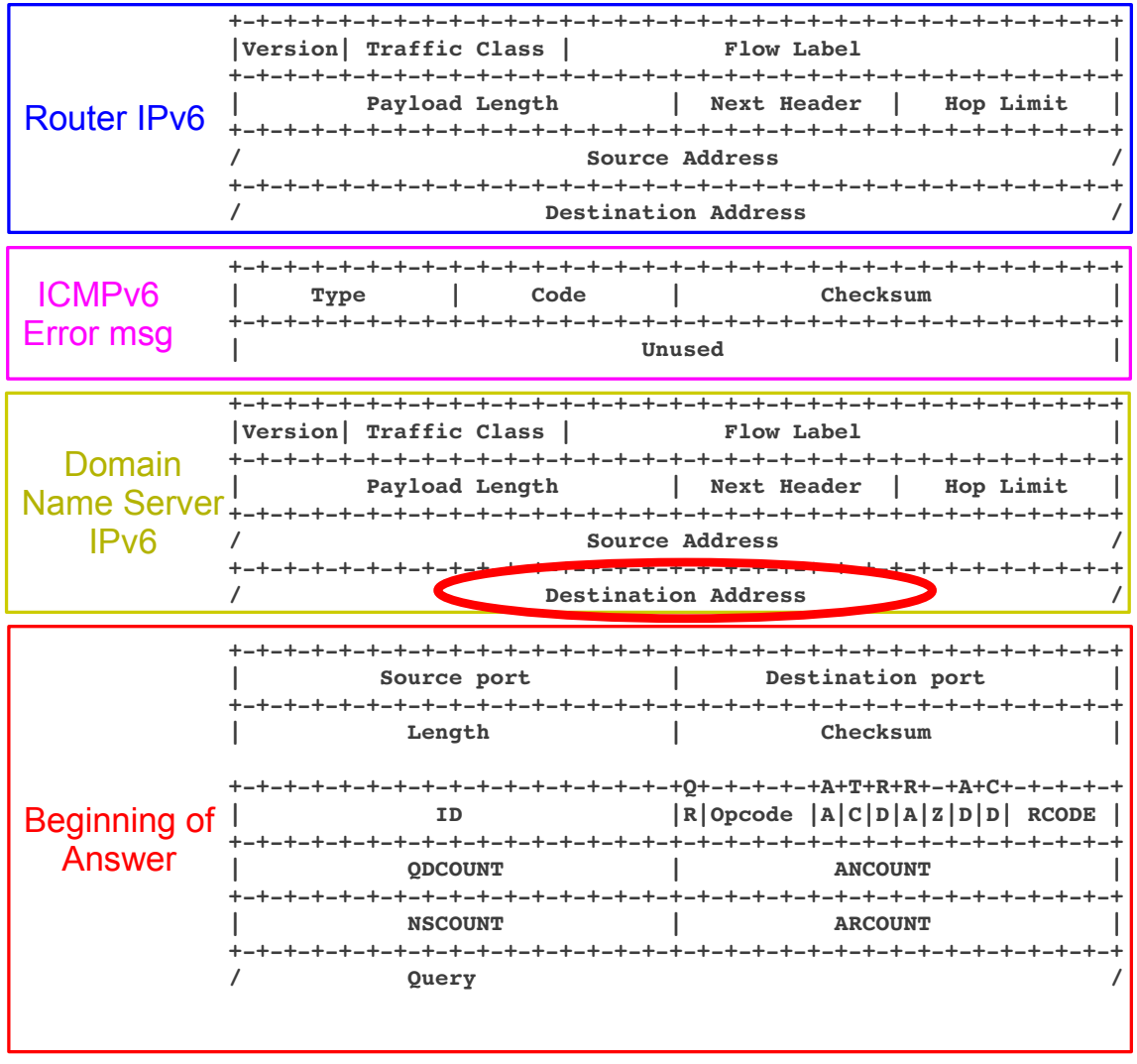
  (or resubmit spoofing the source)
- What message size is client willing to receive?
- Original EDNS0 is lost



Using PMTUD for DNS Observations

- Bypass BCP38: Anyone can spoof a source address.
- Simply re-inject with TC bit: NO GO! (cache poisoning)
- So re-evaluate query at Domain Name Server

  (or resubmit spoofing the source)
- What message size is client willing to receive?
- Original EDNS0 is lost
- Assume 4096: NO GO! (amplification attack)
- So, set EDNS0 udp size to ICMPv6 packet size - 48



Using PMTUD for DNS
Tests and Measurements

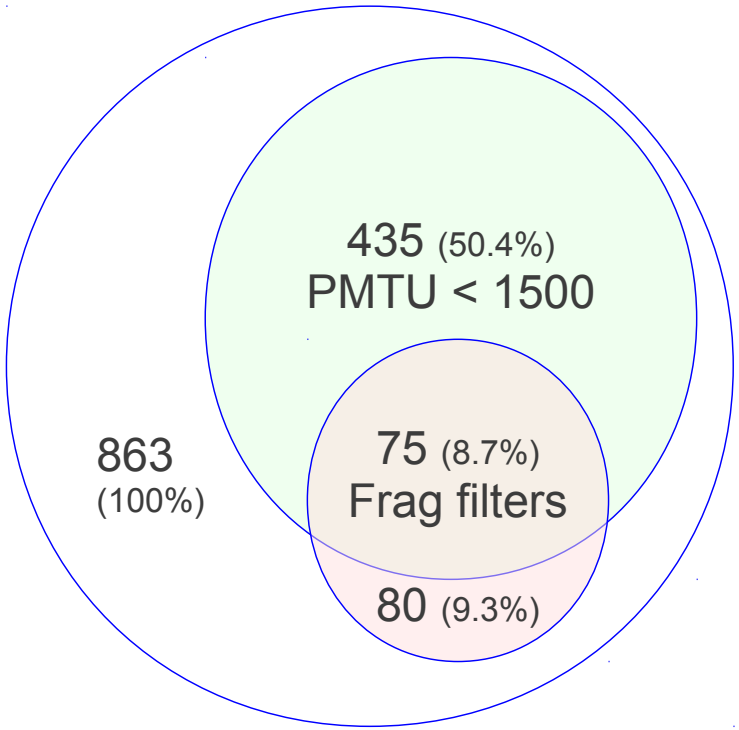
- RIPE ATLAS to query messages from 863 vantage points

measurement	message size	max packet size
baseline	1280	1280
fragment filters	1600	1280
PMTU	1600	1500

Using PMTUD for DNS Tests and Measurements

- RIPE ATLAS to query messages from 863 vantage points

measurement	message size	max packet size
baseline	1280	1280
fragment filters	1600	1280
PMTU	1600	1500

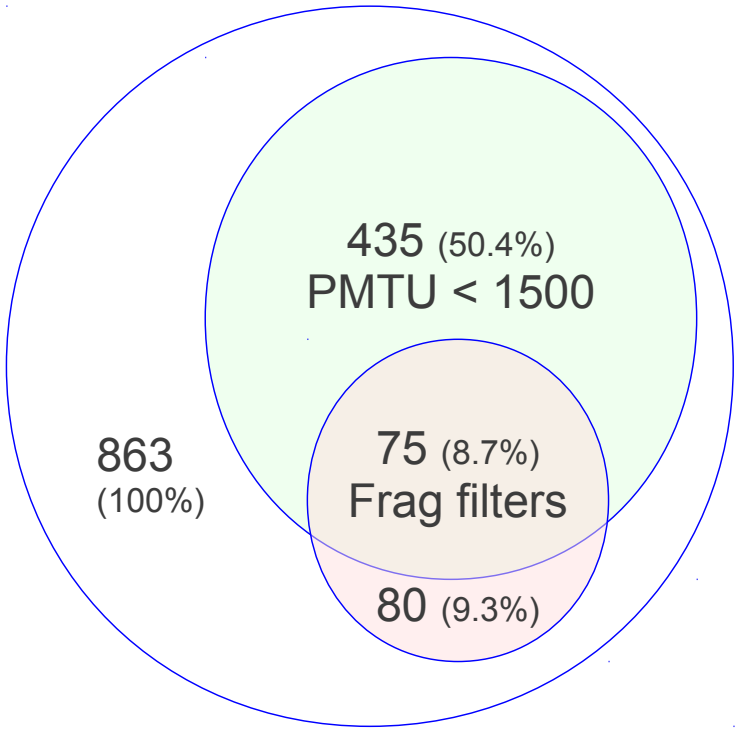


1280	115	1452	2
1300	1	1456	2
1398	1	1460	3
1400	4	1464	3
1418	1	1468	1
1420	1	1472	8
1424	1	1476	6
1428	1	1480	169
1434	3	1488	1
1440	3	1492	76
1450	4	1500	7

Using PMTUD for DNS Tests and Measurements

- RIPE ATLAS to query messages from 863 vantage points

measurement	message size	max packet size
baseline	1280	1280
fragment filters	1600	1280
PMTU	1600	1500



ICMPv6 type	#	*rtt*
address unreachable	2	0.03
administratively prohibited	18	0.03
reassembly time exceeded	13	60.09
Packet Too Big	9	0.07

Observation:

- 18 out of 80 send administratively prohibited

Using PMTUD for DNS
Relevance: Real world capture analysis

- SIDN
- Surfnet

Using PMTUD for DNS
See our blog & Questions for you

- For more info, the student report and working Proof-Of-Concept implementation see blog entry at

  `https://www.nlnetlabs.nl/pmtu4dns`

- How big are your dns answers?
- How critical are your big answers?