Kobayashi Maru: Packet Sizes

A No-Win Scenario in EDNS

DNS OARC 44
2025-02-06
Atlanta, Georgia, USA

Shane Kerr <shane.kerr@ibm.com>
Back-end Engineer

IBM

Customer Issue

IBM NS1 Connect is an authoritative DNS hosting platform.

A customer started getting increased TXT query counts at their zone apex.

They had added enough TXT RR that we started truncating.

Debugging

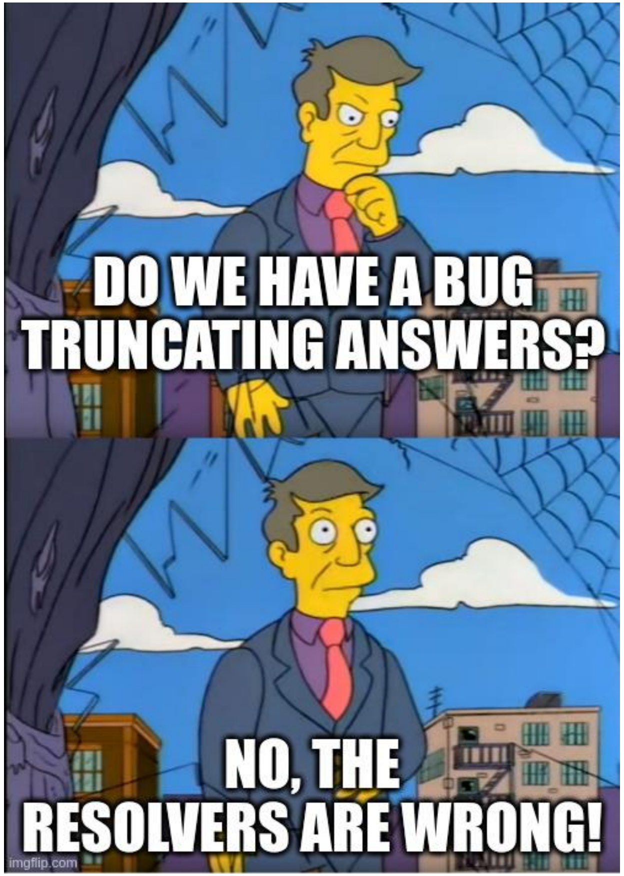
Most resolvers worked fine.

Found some open resolvers having issues. These were *very* old (one running BIND 9.3.4 from 2007).

Was it because of how we truncate?

When we truncate, we respond without any RR other than OPT (where EDNS information sits). Maybe we have to include as much as we can?

But RRL uses this technique, as do other authority servers…


DO WE HAVE A BUG TRUNCATING ANSWERS?
NO, THE RESOLVERS ARE WRONG!
imgflip.com

MOAR Debugging!

It turns out that we were not getting TCP connection requests from the resolvers. They were just trying UDP over and over, and if they were trying TCP we were not seeing any SYN packets.

Customer reduced the size of the TXT record, and the resolvers accepted the answer.

Hypothesis: Resolvers with broken TCP will send a lot of UDP trying to get an answer.

No Way to Win with EDNS Buffer Sizes

Since DNS Flag Day 2020 we are supposed to limit DNS messages to 1232 bytes.

This prevents fragmentation, and gives the best chance of UDP actually working.

This works around networks with broken fragmentation support.

But… resolvers that have networks with *working* fragmentation support, but do not have working TCP, will now have *more* problems.

While we don't care *too* much about resolver operators who have broken setups, we do care about extra queries.

Can We ~~Fix~~ Hack It?

It might be possible to adjust the EDNS buffer size if we detect this behavior.

But our platform does not track resolver behavior, and it would mean adding some system to track state across an entire PoP, or change our load balancing to go to a single server.

We could randomly either respect the resolver EDNS buffer size request *or* limit it to 1232 (or do both at the same time!).

This would mean extra retries, but eventually every resolver would figure it out.

But the whole point of DNS Flag Day was to stop such hacks!

The extra query load isn't enough to cause us performance problems, and our customer had no complaints about reachability.

Our "solution" was to stop charging the customer for these queries. 🙈

Discussion

It seems that at least one authoritative DNS hoster quietly rolled back the DNS Flag Day 2020 values, and now have some higher EDNS buffer size limit (if any limit at all).

Some authoritative DNS hosters are protected because they track resolvers for other reasons, and coincidentally end up blocking resolvers that repeat queries.

Is the DNS Flag Day 2020 recommendation the best possible? RFC 9715 (IP Fragmentation Avoidance in DNS over UDP) discusses larger potential sizes, for example.

Have other operators seen this? What approaches have been taken?

THIS DOCUMENT IS DISTRIBUTED “AS IS” WITHOUT ANY WARRANTY, EITHER EXPRESS OR IMPLIED. IN NO EVENT, SHALL IBM BE LIABLE FOR ANY DAMAGE ARISING FROM THE USE OF THIS INFORMATION, INCLUDING BUT NOT LIMITED TO, LOSS OF DATA, BUSINESS INTERRUPTION, LOSS OF PROFIT OR LOSS OF OPPORTUNITY.

Client examples are presented as illustrations of how those clients have used IBM products and the results they may have achieved. Actual performance, cost, savings or other results in other operating environments may vary.

Not all offerings are available in every country in which IBM operates.

Any statements regarding IBM’s future direction, intent or product plans are subject to change or withdrawal without notice.