draft-ietf-dnsop-serve-stale

- Goal is to increase resiliency when authorities can’t be reached
- Records past expired TTL used only if can’t be refreshed in timely manner
- Drop-in for existing infrastructure
- Possible EDNS option for signalling / debugging
- Some open issues need attention
TTL REDEFINITION

• This is the Standards Track part of the document
• Obvious bit: clarify that TTL-expired data can be used in some circumstances
• Additional bits relevant to 32 bits of TTL field:
  • RFC 2181 says to treat a set high order bit as equivalent to 0. Propose to treat as max
  • $2^{31}$ seconds = 68 years. Suggest more sane configurable default cap
  • Major implementations cap to 1 week or less by default
EDNS OPTION BACKGROUND

• Feature expected to be usable by default, with no signalling necessary
• Option proposed mainly for explicit information and debugging purposes
• Stale-aware client could also use to ignore stale answers if desired
• Subtle point: client can disable stale fallback by asking with recursion disabled
• Mukund Sivaraman suggested non-EDNS method: overload TTL high order bit
EDNS OPTION 1

• More featureful: explicit feedback about which RRSets are stale
• Multiple RRSets can appear in an response, some stale and some not
• More complex code: track which are stale, including through reordering

EDNS OPTION 2

• Simpler: only says whether answer includes any stale records
• Can explicitly signal client does not want stale data, without disabling recursion
**TIMER VALUE RECOMMENDATIONS**

- All should be configurable – recommending defaults
- Client Response Timer, 1.8 seconds
  - How long to attempt resolution before falling back to stale data
- Query Resolution Timer, 10-30 seconds
  - Existing resolver behaviour, bounds how long a given resolution attempt takes
- Maximum Stale Timer, 1 week
  - Hard cap on TTL, after which records cannot be used even for stale responses
- TTL on returned stale records, 30 seconds
  - Balances getting fresh data versus unhelpful work
  - Typically ignored by stubs, mainly useful when forwarding from a cacheful resolver
SUPPORT / ANTIPATHY

• Authors’ impression is that this general idea is widely supported

• However, it has been called:
  • “shockingly poor engineering”
  • “only commercial benefits which will impose costs on all internet users for the sake of a few large companies…”

• If opposed, now’s the time to speak up

• Hoping for WGLC after previous issues are addressed