Changes since -02

• -03
  • Updated wouters-edns-tcp-keepalive reference

• -04
  • Details of connection sharing, and determination of “same server”

• -05
  • More on determination of “same server”
Changes in -06

- Supporting DNS Update on the same server is optional
- Standard DNS Queries are allowed on the connection, for names within the zone
- Supporting classes other than “IN” is optional
- Discussion on which side closes the connection, and under what circumstances
- Use of EDNS(0) TCP Keepalive timeout value to specify delay before retrying after an error
- More details of Termination Messages
Changes in -07

- Clarification of qtype ANY
- Responses MUST echo back operation ID
- Further clarification about connection closing:
  - Client should not keep idle connections open indefinitely
  - Client should not assume that a single error kills a connection
Discussion Points

• EDNS(0) TCP Keepalive option is implicit for SUBSCRIBE

• Need EDNS(0) TCP Keepalive option in all responses, or only if idle timeout changes?

• What are “the required keepalives”?

• Close connection with TCP RST vs. FIN

• Should UNSUBSCRIBE and RECONFIRM generate responses?
EDNS(0) TCP Keepalive option is implicit for SUBSCRIBE

- Saves some bytes on the wire
- But makes SUBSCRIBE a special case
EDNS(0) TCP Keepalive option in all responses, or only if idle timeout changes?

- Saves some bytes on the wire
- But TCP Keepalive spec suggests it needs to be repeated in every response
The DNS Keepalive spec doesn’t specify any way to keep a connection alive

• TCP-layer keepalives?
  • Lower overhead
  • What OS APIs exist for this?

• DNS-layer keepalives?
  • A DNS QUERY containing zero questions?
  • A DNS SUBSCRIBE containing zero questions?
  • An "empty" DNS message over the TCP connection?
  • Provides application-layer liveness check
Close connection with TCP RST vs. TCP FIN

• Does it matter?
Should UNSUBSCRIBE and RECONFIRM generate responses?

- Right now they do not
- Sending pointless responses may just be a waste of bytes
- Development and debugging can be achieved by looking at log files