RFC 7706: Decreasing Access Time to Root Servers by Running One on Loopback

A good idea or not?

Petr Špaček • petr.spacek@nic.cz • 2017-09-29

Talk outline

- Problems with RFC 7706
- Comparison with RFC 8198
  - theoretical
  - experimental
- Possible improvements
- Shameless self-promotion

RFC 7706: Root on loopback

- "Because of **the significant operational risks** described in this document, distributions of recursive DNS servers MUST NOT include configuration for the design described here."
- Is it worth the trouble?

RFC 7706: Root on loopback recap

- Primary goals
  - faster negative responses
  - preventing queries from being visible
  - ~~faster positive responses~~
- Side effects
  - higher resiliency? maybe?

RFC 8198: Aggressive cache recap

- Primary goals
  - faster negative responses
  - faster positive responses (wildcards)
- Depends on data in cache
- Side effects
  - preventing queries from being visible

RFC 7706 and 8198 overlap

- RFC 8198 almost provides what 7706 calls for
- How effective is 8198?
  - Gut feeling: good
  - Measurements?

Experimental setup

- Replay PCAP to Knot Resolver
- Log cache accesses
- Replay cache accesses to RFC 2308 & 8198 simulator
- Record hit/miss **for nodes in the root zone**

Data sets

- 4+ days of traffic in PCAP
- Public Open Resolver ran by CZ.NIC ("big")
  - 3500 q/second
  - anonymised
- Two households in Czech Republic ("small")
  - dominated by "noise"

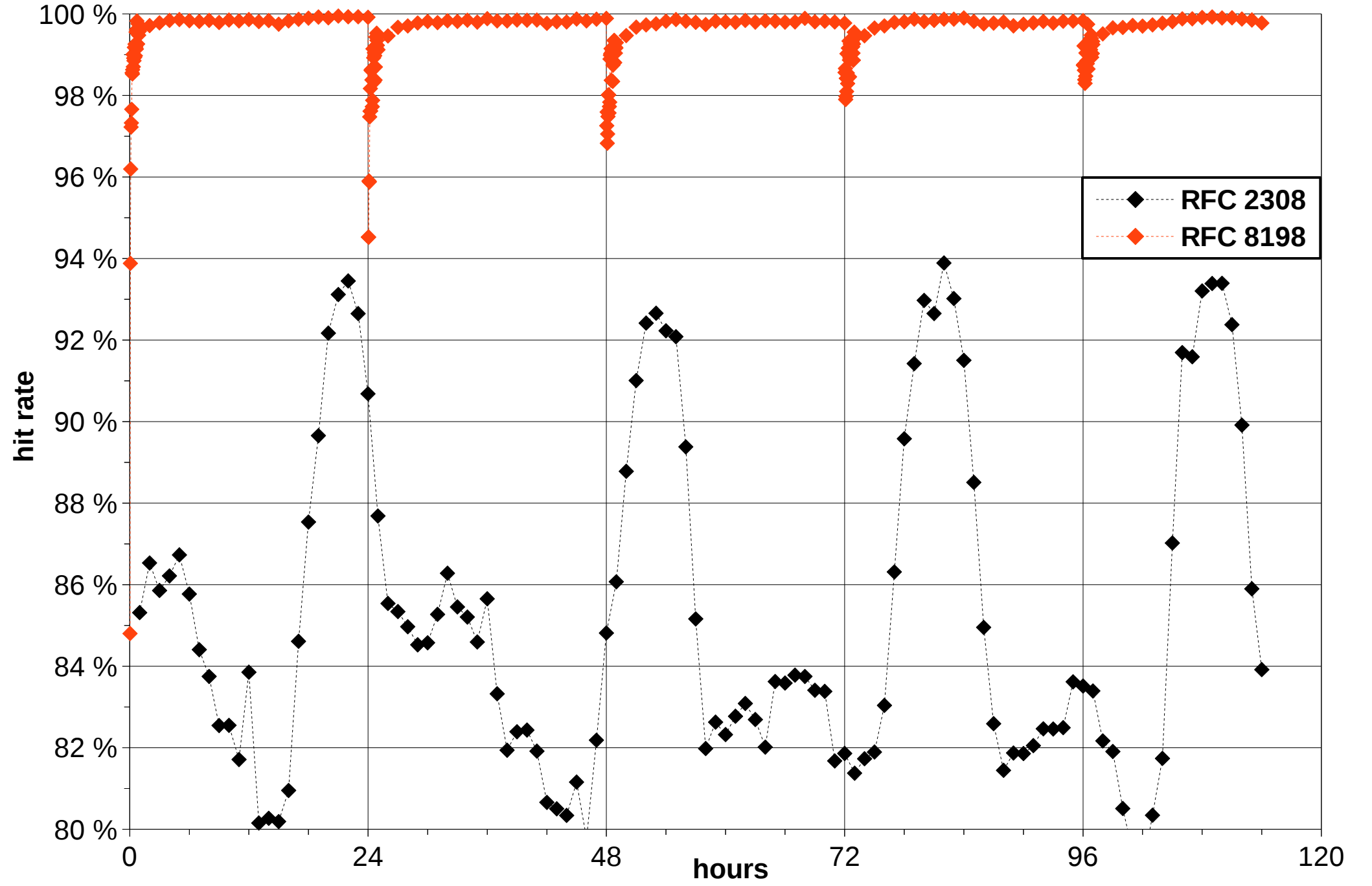
Tools

- Knot Resolver 1.3
  - patched to log cache access
- Drool to replay traffic
- RFC 2308 & 8198 simulator: https://github.com/pspacek/dnscache_simulator
  - unlimited cache size

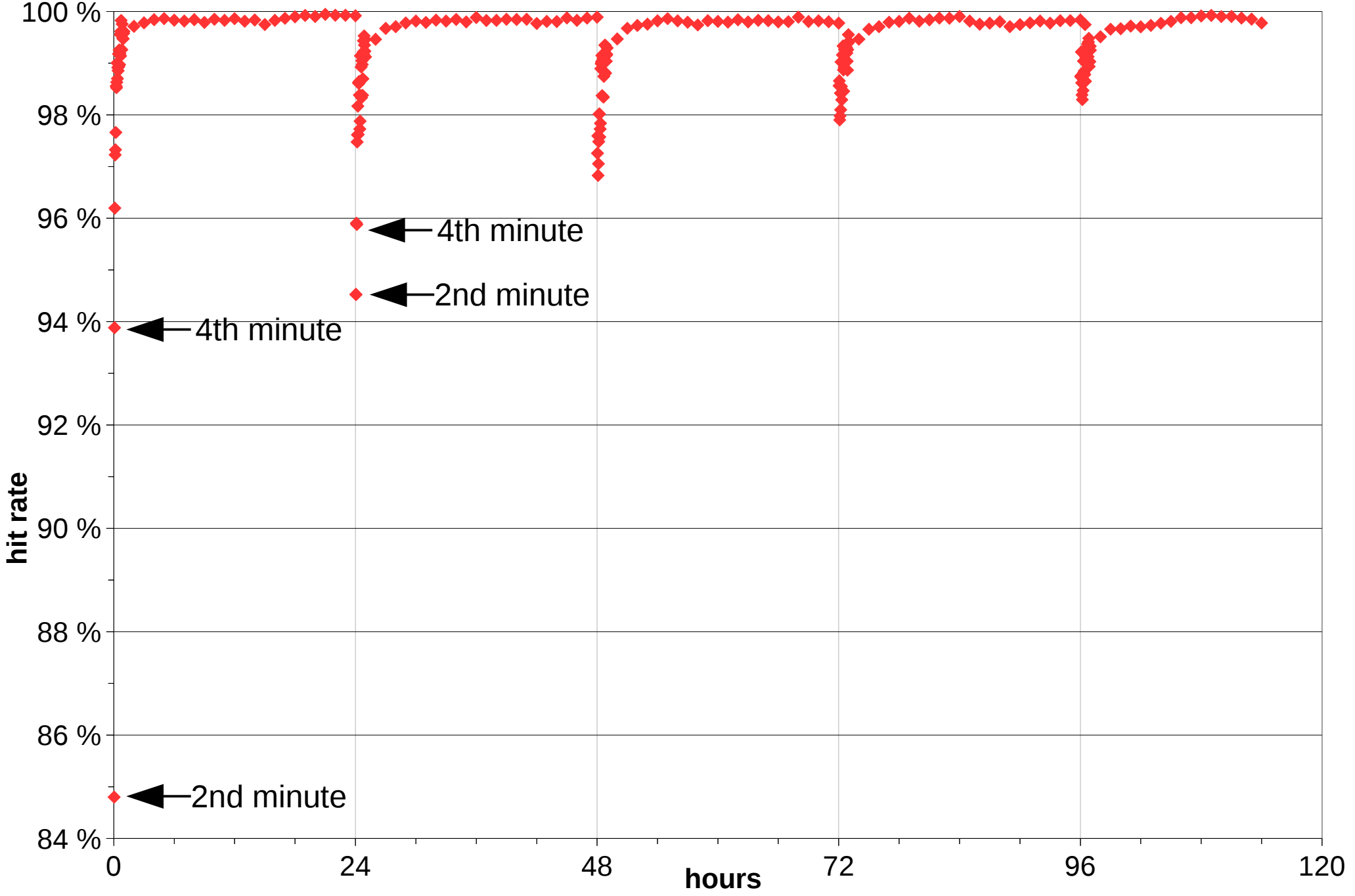
Results for root zone data

- Households = noise (no further analysis)
- Public resolver = RFC 8198 show case
  - **only 0,25 %** cache misses for root zone data
- About 3300 cache misses **per day**
  - 73 % of root zone
  - ~ 6600 UDP packets

RFC 2308 / 8198 comparison (root zone)



RFC 8198 cache hit rate (root zone)



Root zone content

- Minimal TTL = 1 day
- 1548 nodes with NSEC RR
- 4497 non-glue non-RRSIG RRs
- AXFR
  - 388 TCP packets
  - 1 363 891 bytes

RFC 7706's goals

- ☑ Faster negative responses
- ☑ Preventing queries from being visible
- ☑ **Provided by RFC 8198**
  - except for 0,25 % of queries
- ☐ Higher resiliency
  - not provided by RFC 8198 but ...

Leftovers after RFC 8198

- 0,25 % cache miss rate
  - caused by empty/expired cache
- **Pre-fill cache to get to 0 %**
  - Min TTL 1 day = 1 AXFR/day
  - AXFR/day requires just 6 % of packets for queries
- Higher resiliency
  - use a variant of draft-tale-dnsop-serve-stale-01

Is RFC 7706 worth the trouble?

- **NO!**
- Replace it with
  - RFC 8198
  - cache pre-fill
    - open question: AXFR from where?
  - a variant of draft serve-stale
- Watch out for Knot Resolver in 2018!

Thanks to Ondřej Surý!

Stay tuned for Knot news!

lead by
Daniel Salzman

lead by
Petr Špaček

Knot news for October 2017

lead by
Daniel Salzman

- **Knot DNS 2.6**
- Automatic DNSSEC algorithm rollover
- In-line signing on slave

lead by
Petr Špaček

- **Knot Resolver 2.0**
- RFC 8198 aka Aggressive Use of DNSSEC-Validated Cache