# Benchmarking DNS resolvers

## using realistic workload

Petr Špaček • petr.spacek@nic.cz • 2019-10-16

KNOT RESOLVER

CZ.NIC | CZ DOMAIN REGISTRY

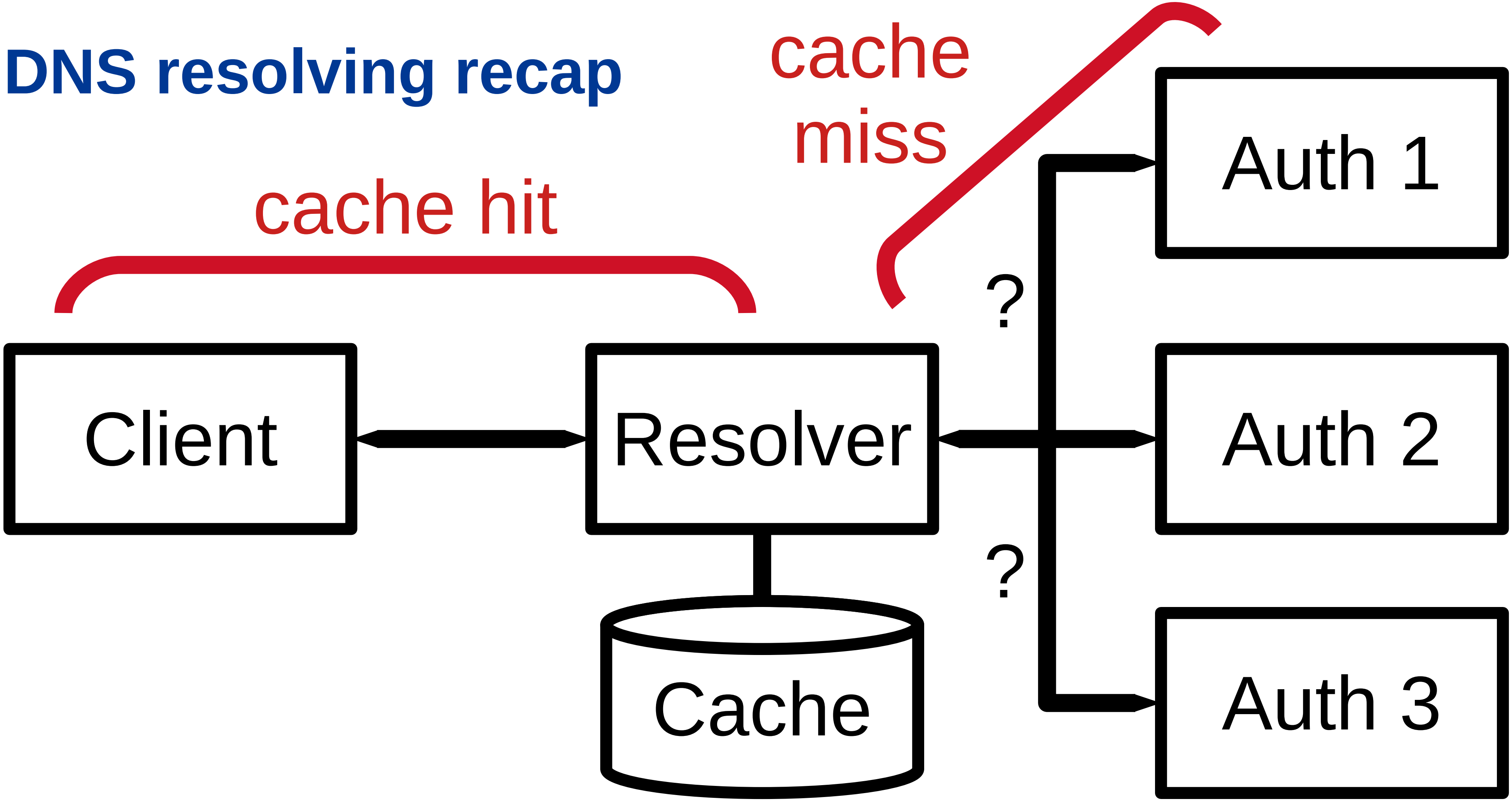
# Talk outline

- Motivation
- Classic approach
- Classic pitfalls
- DNS Shotgun – tool for realistic benchmarking

# Motivation

- Running DNS resolver ⇒ power, cooling
- Power, cooling ⇒ €€€
- Benchmarking ⇔ optimization
  - ⇒ cost reduction

# DNS resolving recap



cache miss

cache hit

Client

Resolver

Cache

Auth 1

Auth 2

Auth 3

?

?

# Inside of a DNS resolver: Cache hit

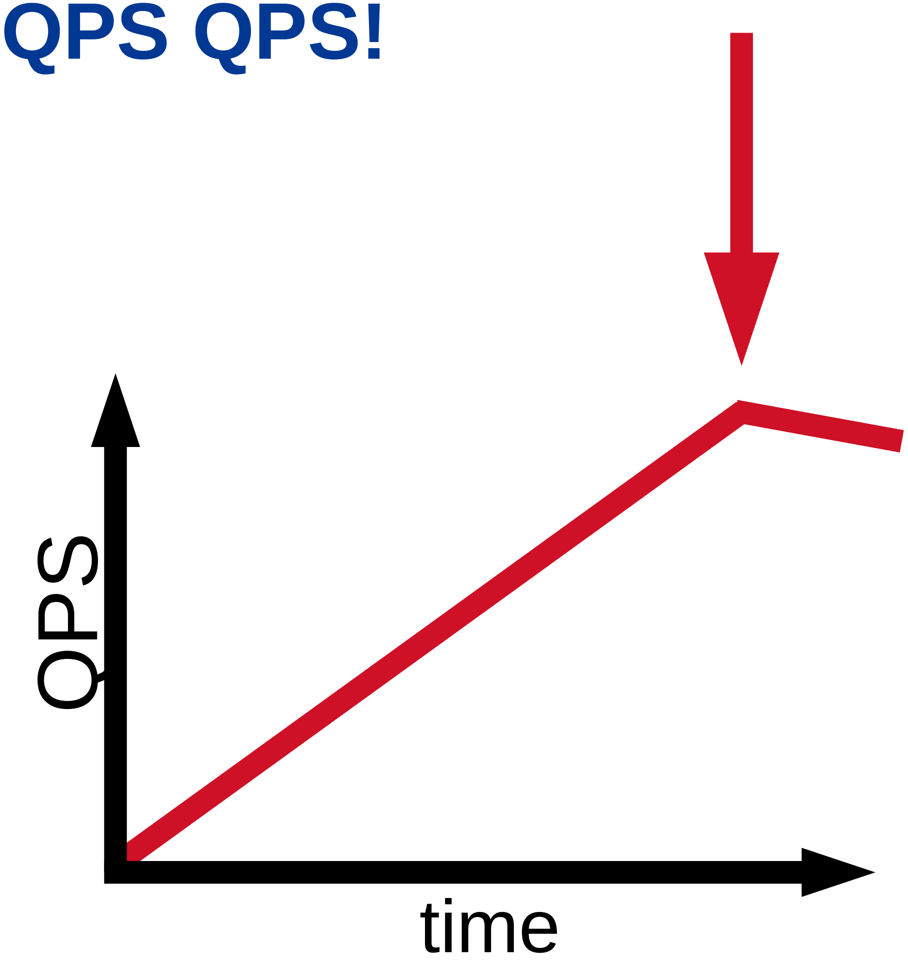
- Query parsing
- Cache search
- Answer serialization

# Inside of a DNS resolver: Cache miss

- Authoritative server selection – who to ask?
- Retransmit strategy
- DNSSEC validation
- Socket management – reuse? randomization?
- Policy engine
- Cache write & eviction

# Classic benchmarking: QPS QPS QPS!

- `$ man resperf`
- Query list: tcpdump -> text
- Ramp-up query traffic
- Find max QPS
  - Response rate drops



# Classic pitfalls 1/2

- No query timing
  - Ignores TTL ⇒ **unrealistic cache hit rate**
- Text query list
  - EDNS info lost ⇒ **unrealistic** TCP fallbacks
- QPS ramp-up
  - Waits for cache hit rate increase ⇒ **unrealistic**
  - Resolver restart!

# Classic pitfalls 2/2

- Small # of clients
  - Affects workload distribution
- No fallback to TCP
  - TrunCated bit
- No connection management
  - TCP, TLS, DoH!
- **Over-focuses on QPS!**

# DNS Shotgun: Introduction

- New toolset
  - Based on `dnsjit` by DNS-OARC
  - https://www.dns-oarc.net/tools/dnsjit
- Realistic DNS benchmarking
- Open-source
  - https://gitlab.labs.nic.cz/knot/shotgun/

# DNS Shotgun: Client-based approach

- **How many clients can the resolver handle?**

- Performance depends on clients
  - IoT, mobile, desktop, mail server, …

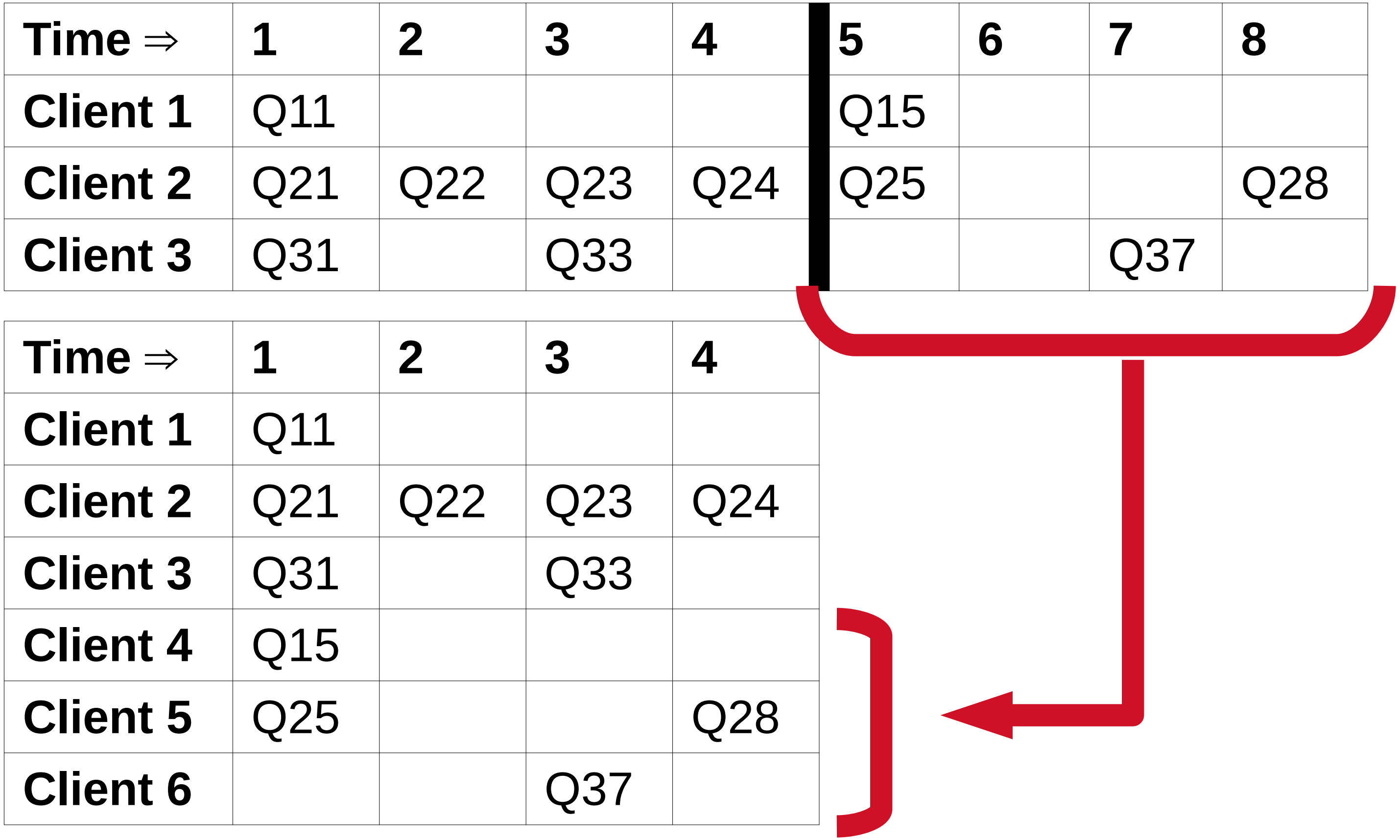
# DNS Shotgun: Principle

- Phase 1: Analyze traffic patterns in PCAPs
- Phase 2: Simulate ***N*** *of* ***your*** clients

# DNS Shotgun: Traffic analysis

- Query stream for each IP/DNS client
  - IoT – mobile – desktop – mail server …
  - Beware! NAT!
- Pre-generate test data
  - ***N*** clients with ***S*** seconds
  - ***S*** = 60 seconds
  - ***N*** = 100k, 200k, 300k, …, 1M

# DNS Shotgun: 3 => 6 clients – generation



| Time ⇒ | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|---|---|---|---|---|---|---|---|---|
| Client 1 | Q11 | | | | Q15 | | | |
| Client 2 | Q21 | Q22 | Q23 | Q24 | Q25 | | | Q28 |
| Client 3 | Q31 | | Q33 | | | | Q37 | |

| Time ⇒ | 1 | 2 | 3 | 4 |
|---|---|---|---|---|
| Client 1 | Q11 | | | |
| Client 2 | Q21 | Q22 | Q23 | Q24 |
| Client 3 | Q31 | | Q33 | |
| Client 4 | Q15 | | | |
| Client 5 | Q25 | | | Q28 |
| Client 6 | | | Q37 | |

# DNS Shotgun: Client simulation

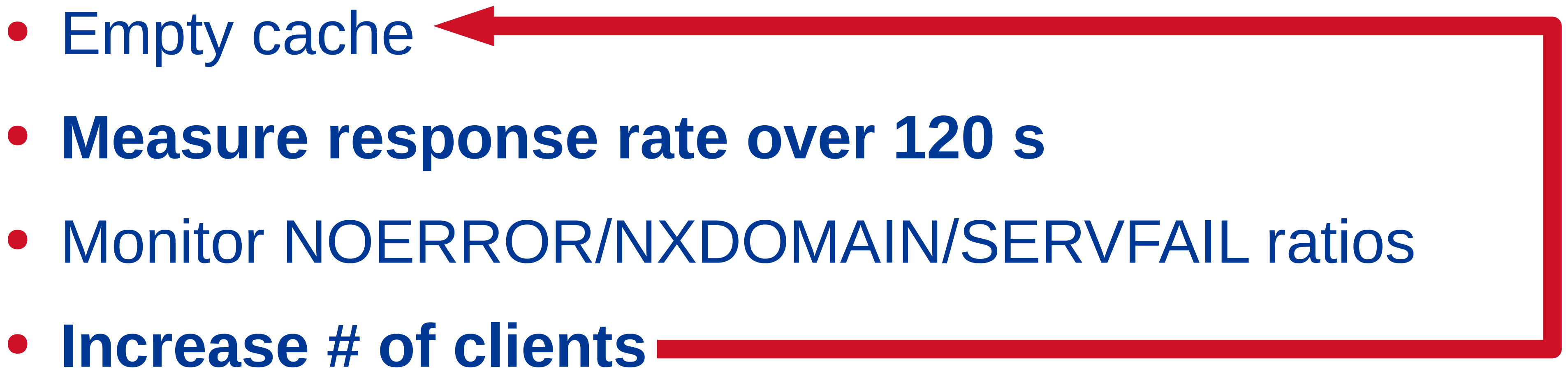
- Replay pre-generated traffic
- Socket/connection per query/client
- Keep ± 1 second query timing
  - Realistic cache hit rate
  - ⇒ QPS varies over time
- Want higher "QPS"? Add clients!

# DNS Shotgun: Performance testing

- Simulate ***N*** clients
  - Analyze respose rate + RCODEs
  - Monitor resource usage
- Increase ***N***
  - … as long as resolver can keep up
- ***N*** = maximum # of clients
  - for given input PCAP & connection parameters

# DNS Shotgun: Experiment

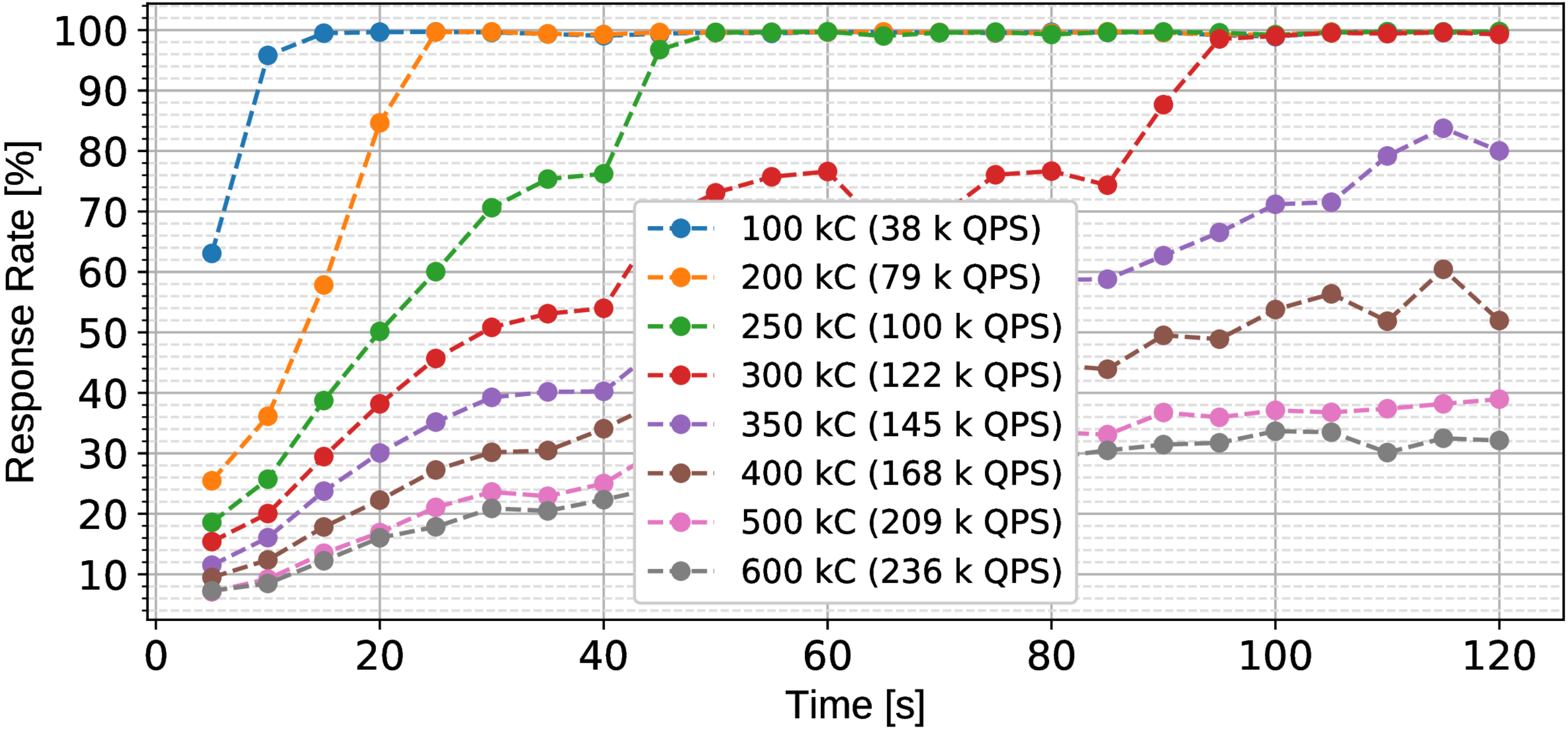
- Input: anonymized traffic from a Czech university



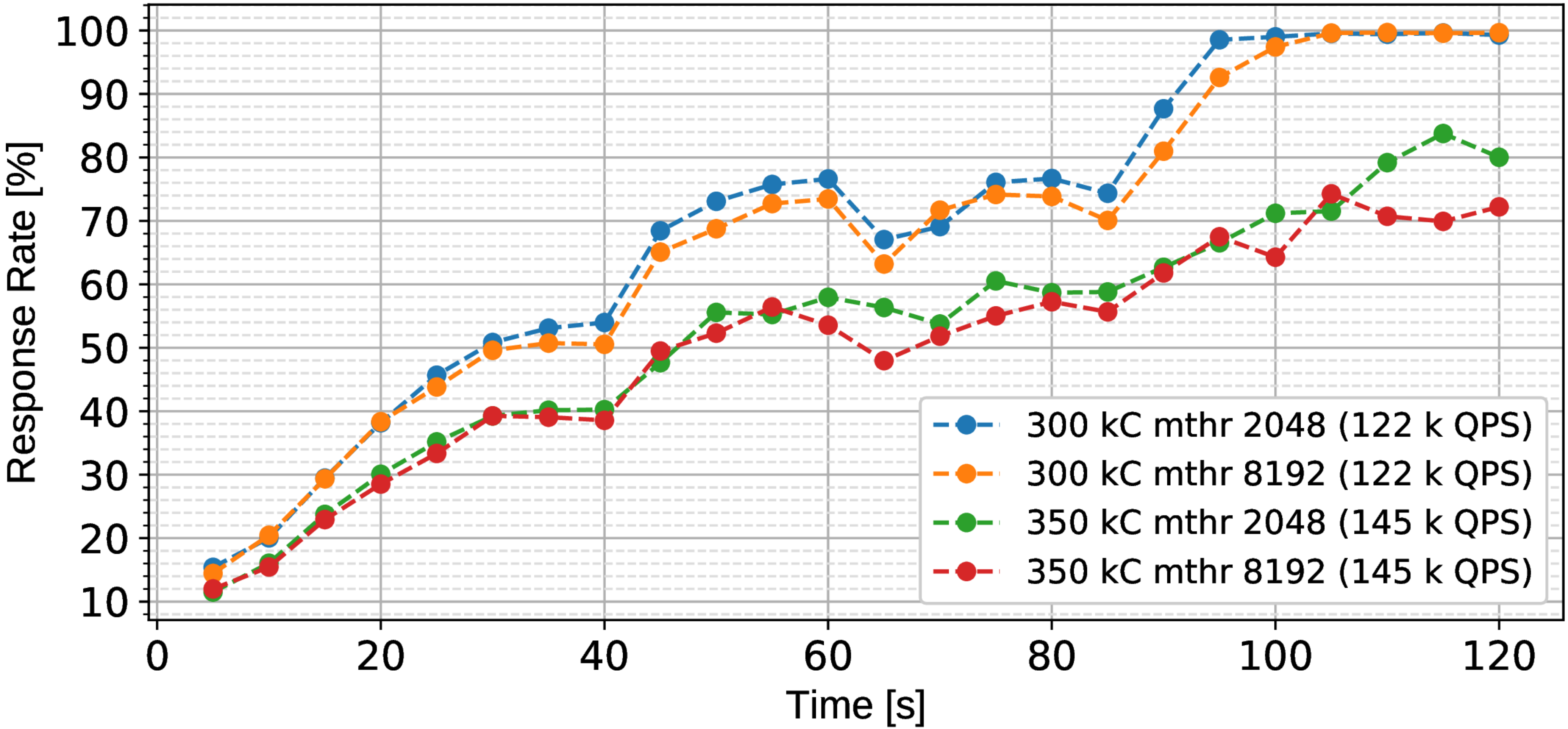
- Empty cache
- **Measure response rate over 120 s**
- Monitor NOERROR/NXDOMAIN/SERVFAIL ratios
- **Increase # of clients**

- 4 CPUs, no qname minimization, same cache params

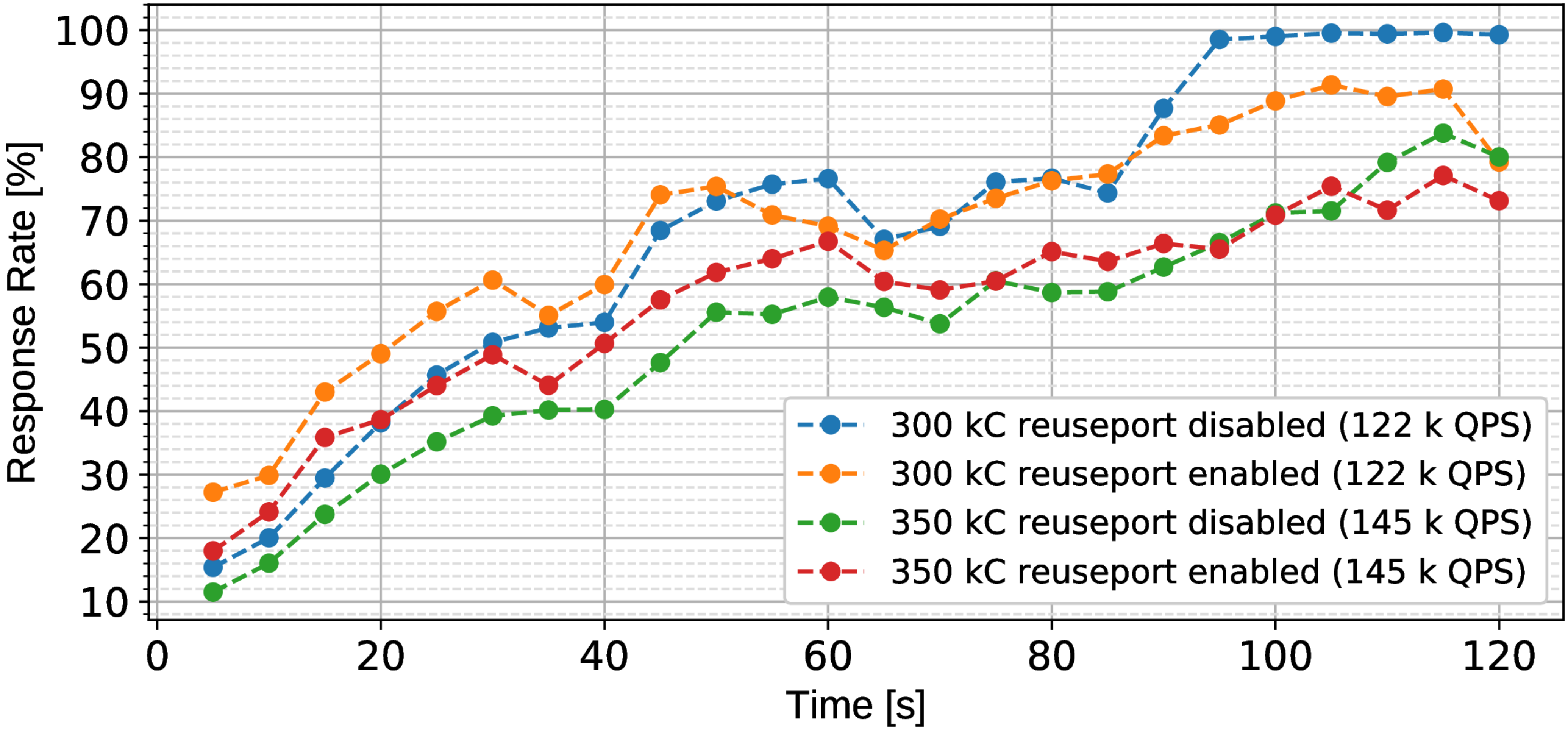
# PowerDNS Recursor 4.2.0: defaults



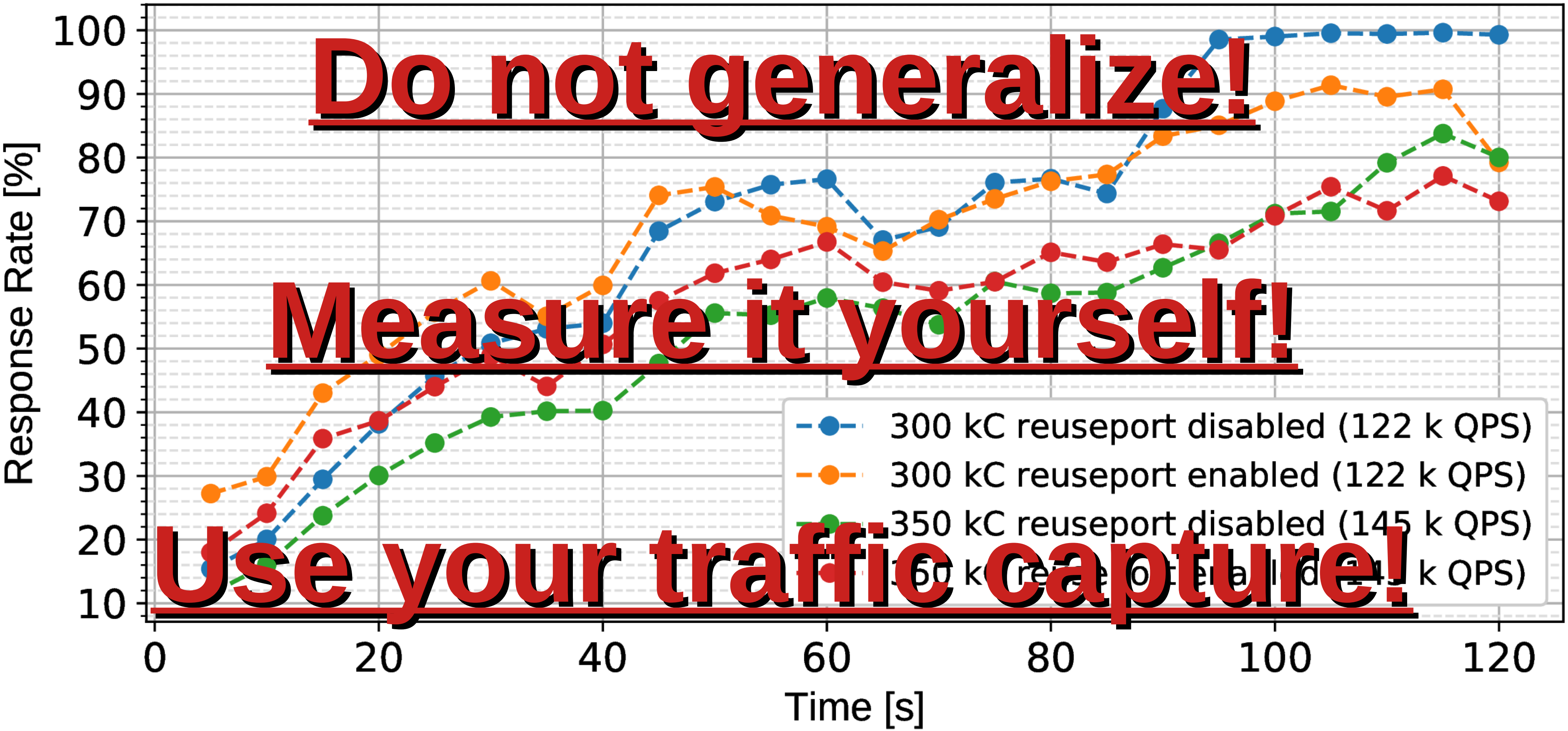
# PowerDNS Recursor 4.2.0: max-mthreads?



# PowerDNS Recursor 4.2.0: reuseport?



# PowerDNS Recursor 4.2.0: reuseport?

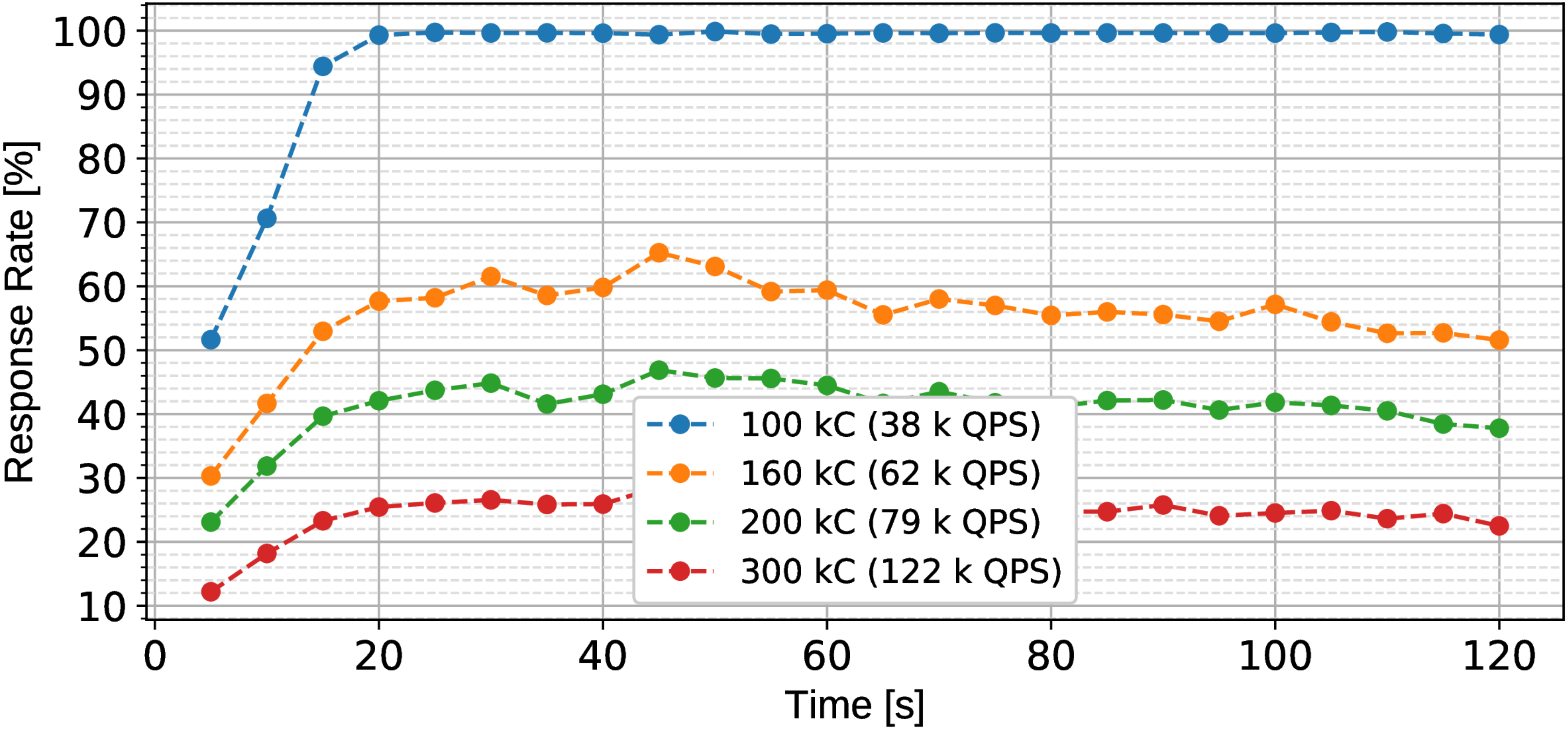


**Do not generalize!**

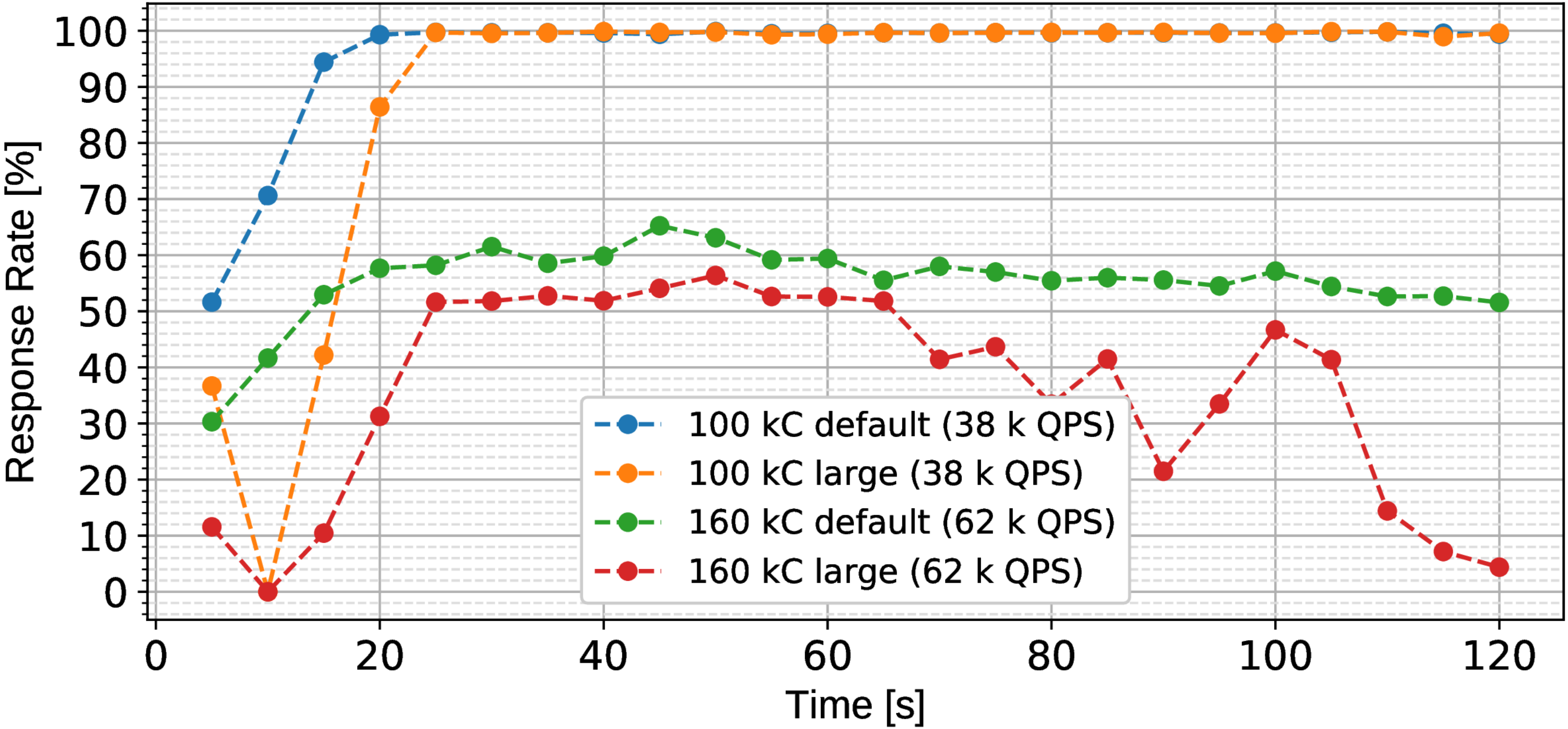
**Measure it yourself!**

**Use your traffic capture!**

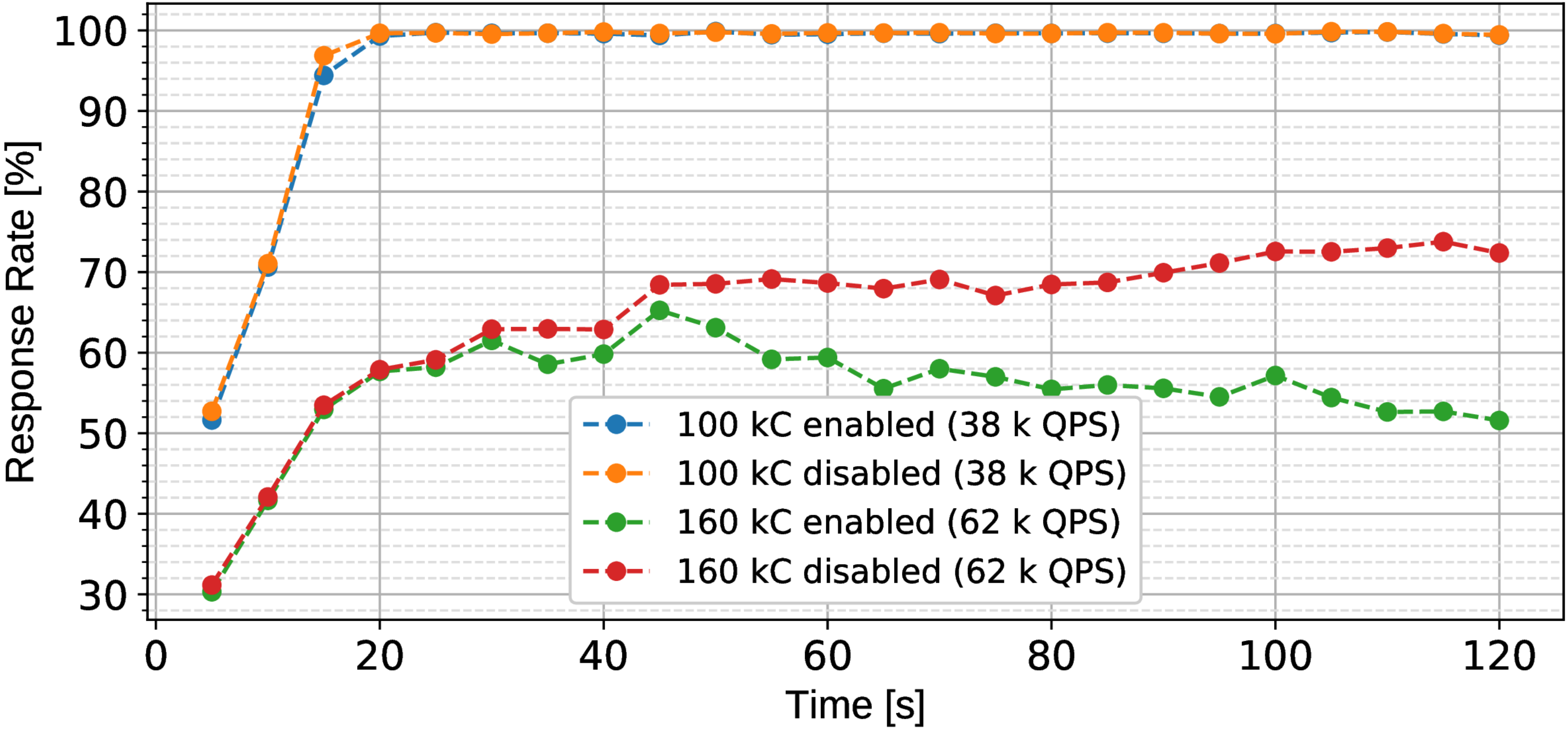
# BIND 9.14.6: --tuning=default



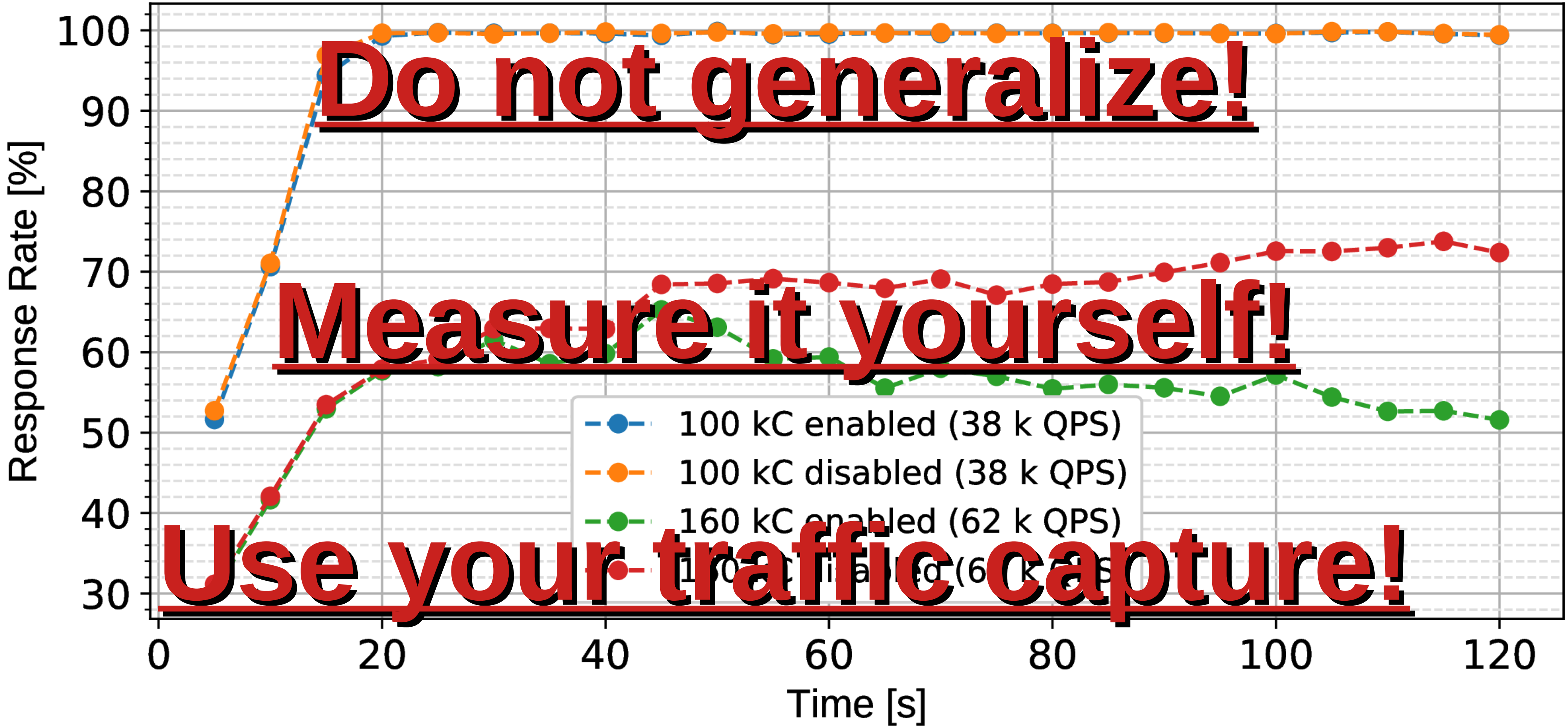
# BIND 9.14.6: --tuning=?



# BIND 9.14.6: --tuning=default, synth-from-dnssec ?



# BIND 9.14.6: --tuning=default, synth-from-dnssec ?

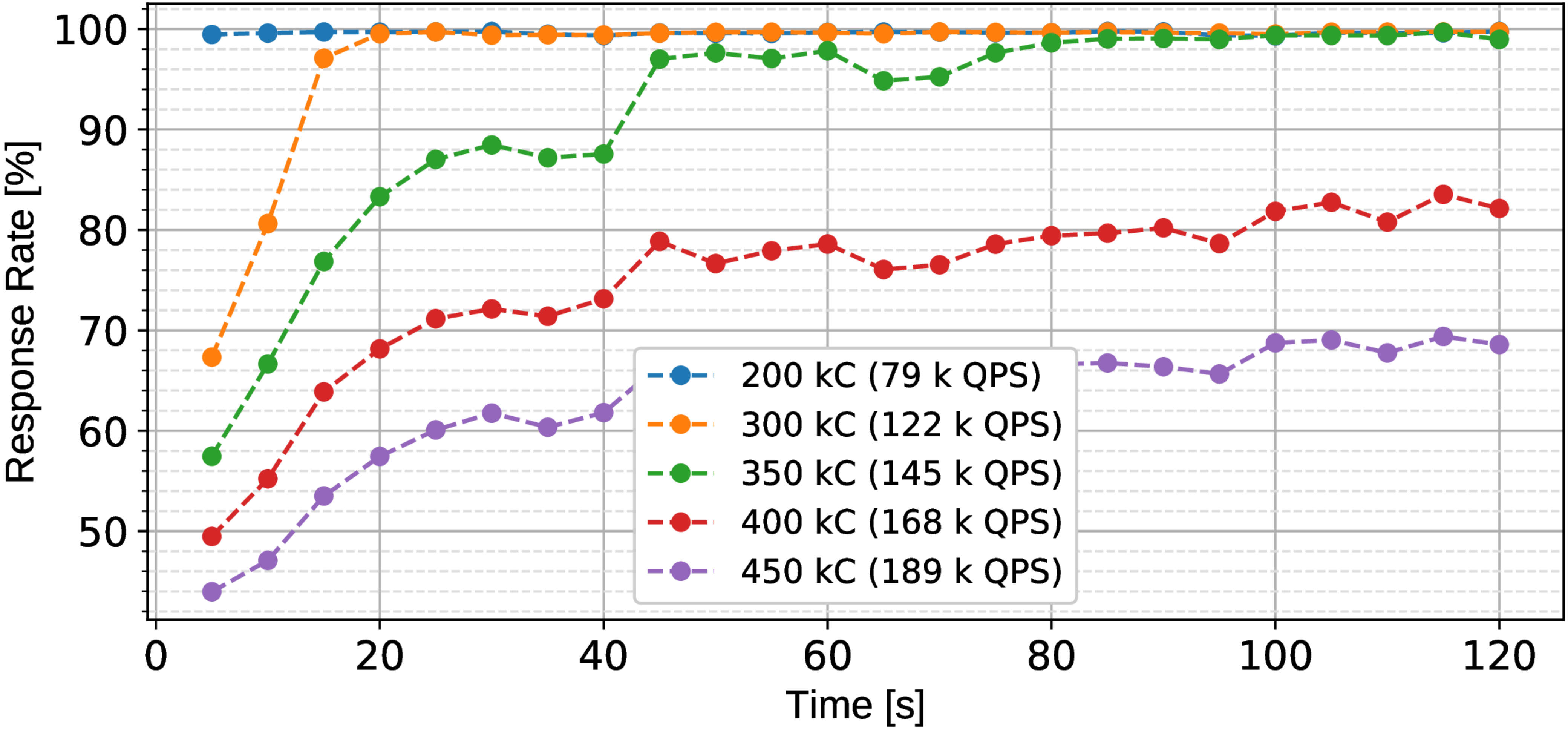


Do not generalize!

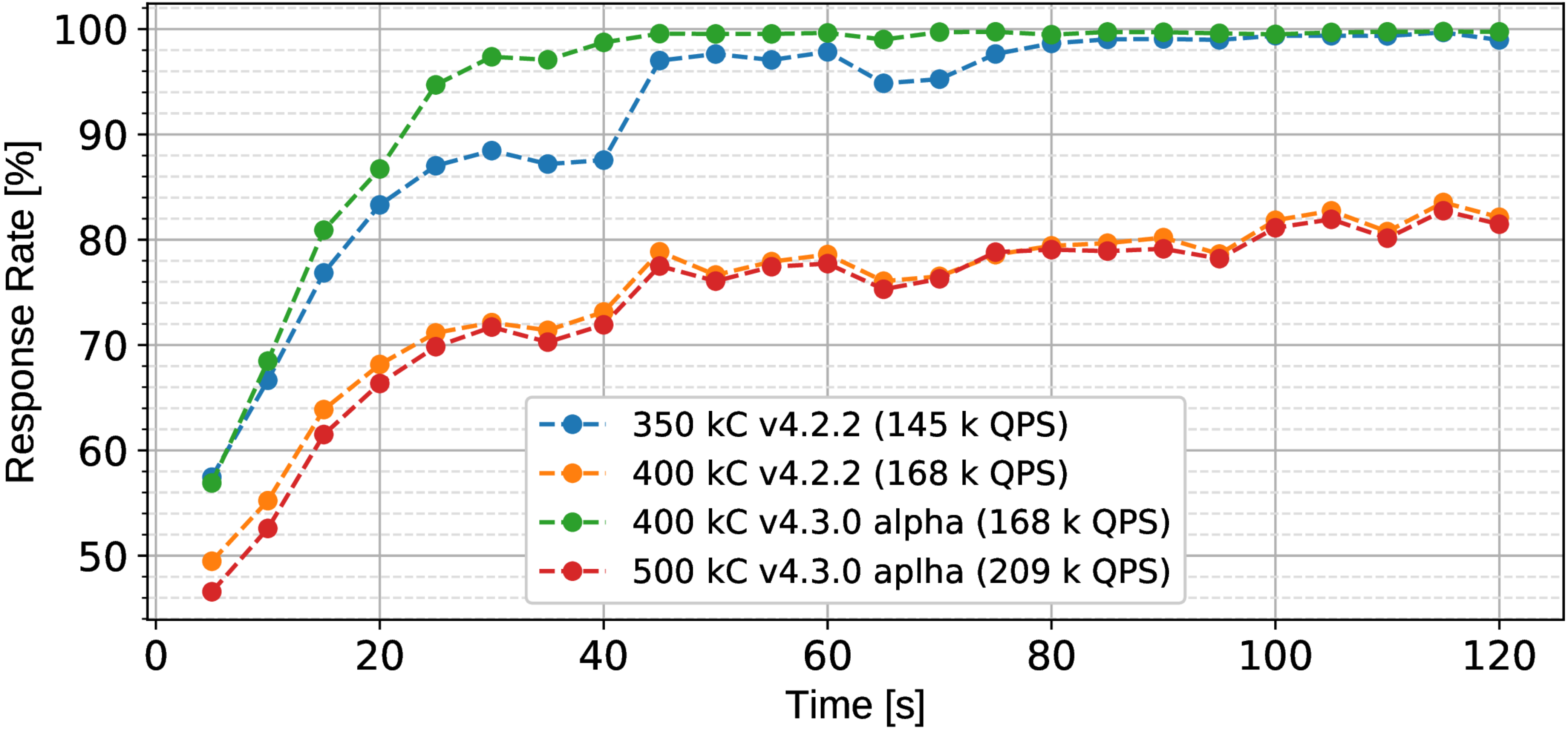
Measure it yourself!

Use your traffic capture!

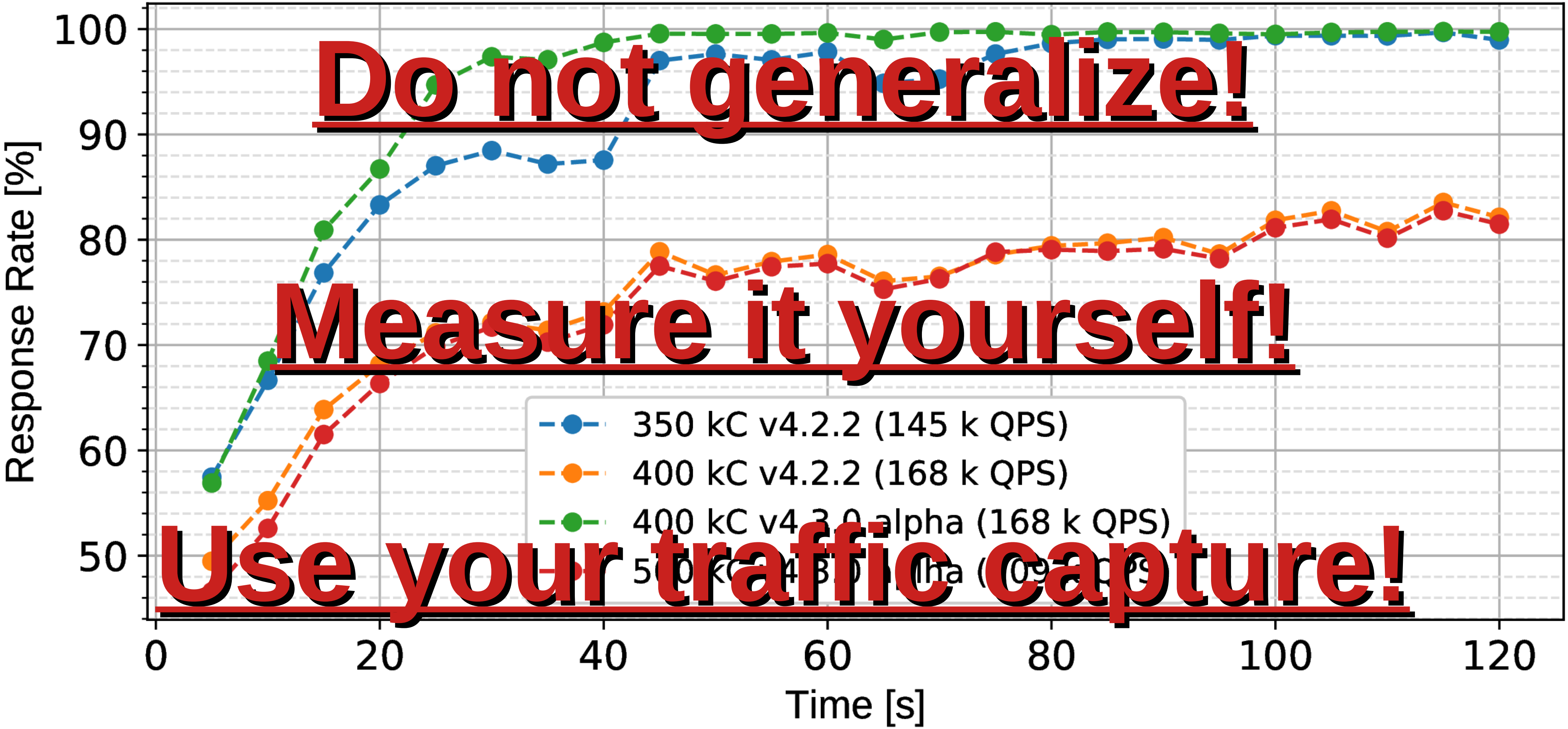
# Knot Resolver 4.2.2 defaults



# Knot Resolver 4.2.2 vs. to-be-4.3.0



# Knot Resolver 4.2.2 vs. to-be-4.3.0

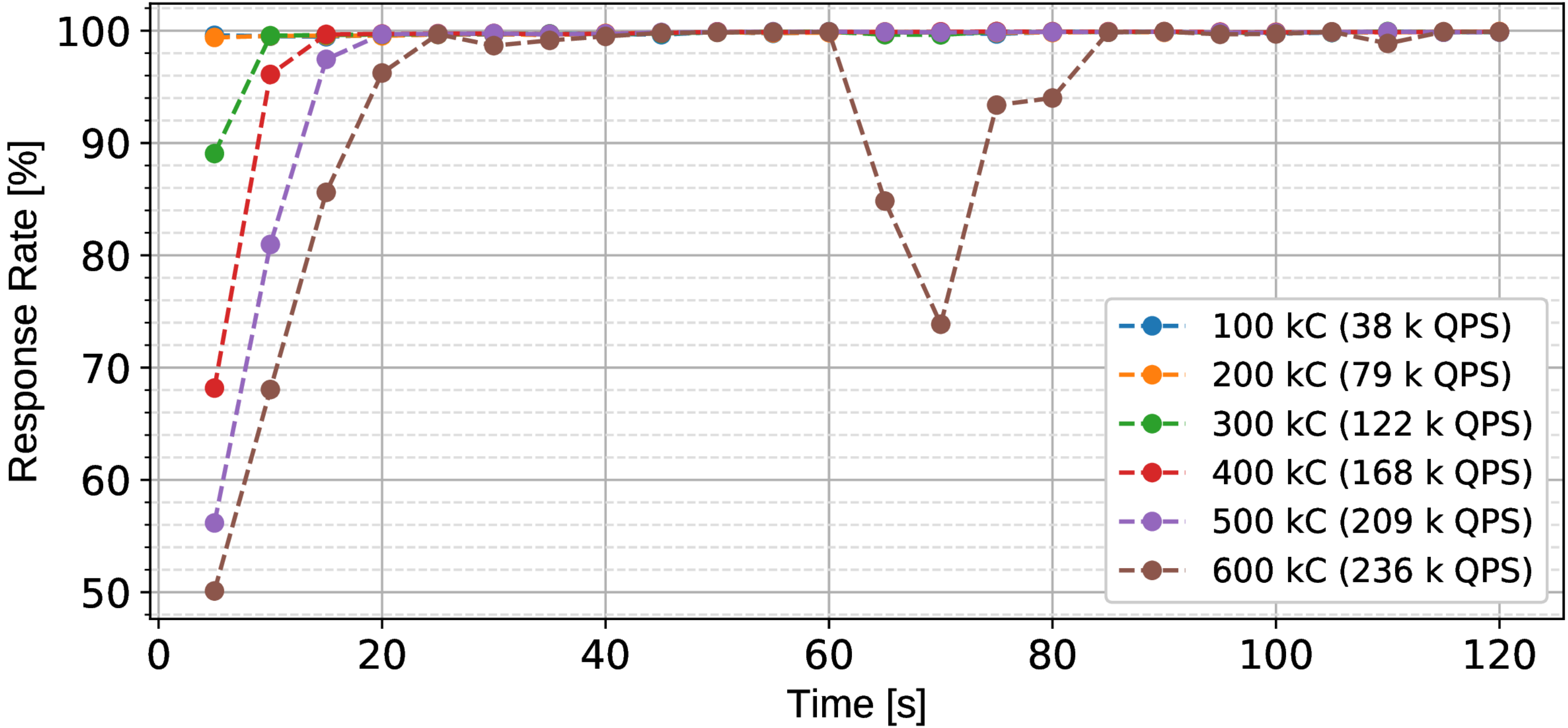


Do not generalize!

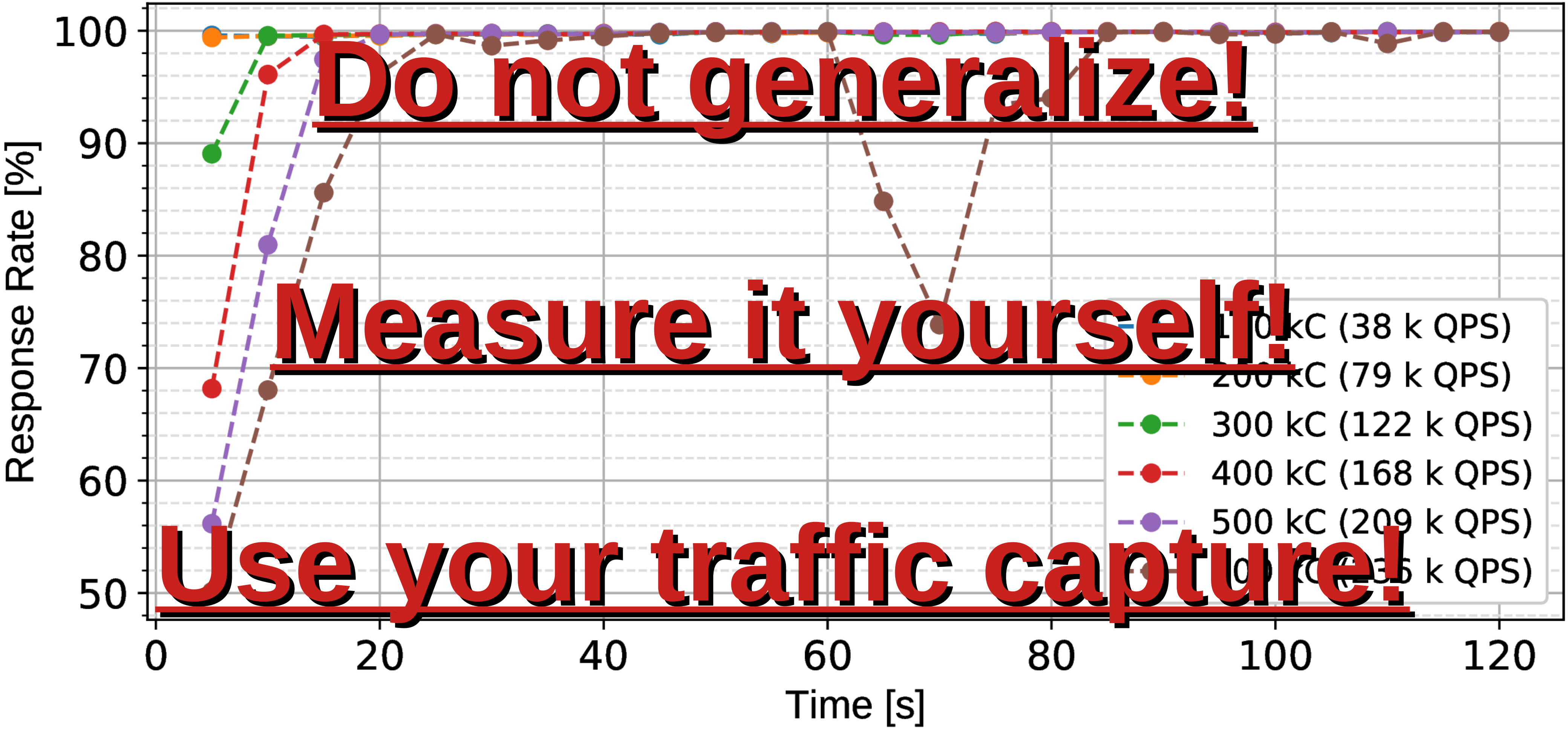
Measure it yourself!

Use your traffic capture!

# Unbound 1.9.4


Response Rate [%]
100
90
80
70
60
50
Time [s]
0
20
40
60
80
100
120
100 kC (38 k QPS)
200 kC (79 k QPS)
300 kC (122 k QPS)
400 kC (168 k QPS)
500 kC (209 k QPS)
600 kC (236 k QPS)

# Unbound 1.9.4



Do not generalize!

Measure it yourself!

Use your traffic capture!

# DNS Shotgun: Limitations

- Requires **a lot** of PCAPs
  - 1 hour, 1k clients
    = 6 minutes, 10k clients (simulated)
- Results depend on input traffic capture
  - … simulates **your own clients**
- TCP/TLS/DoH not supported *yet*

# DNS Shotgun: Try it

- Very much work-in-progress
  - Here be dragons! :-)
- Try it anyway
  - https://gitlab.labs.nic.cz/knot/shotgun
- **Sponsors needed!**
  - TCP/TLS/DoH support
  - Configurable connection reuse (pipelining, keepalive)

# Closing remarks

- DNS micro-benchmarks do not reflect real world
- HW & OS changes invalidate results
- Generalization is hard
  - Compare using **your config** and **your traffic**
- Interested in benchmarking? Get in touch
  - petr.spacek@nic.cz
  - https://gitlab.labs.nic.cz/knot/shotgun

CZ DOMAIN
REGISTRY