Scaling DNS resolvers

for UDP, TLS and HTTPS traffic

Petr Špaček • petr.spacek@nic.cz • 2020-09-28

KNOT RESOLVER

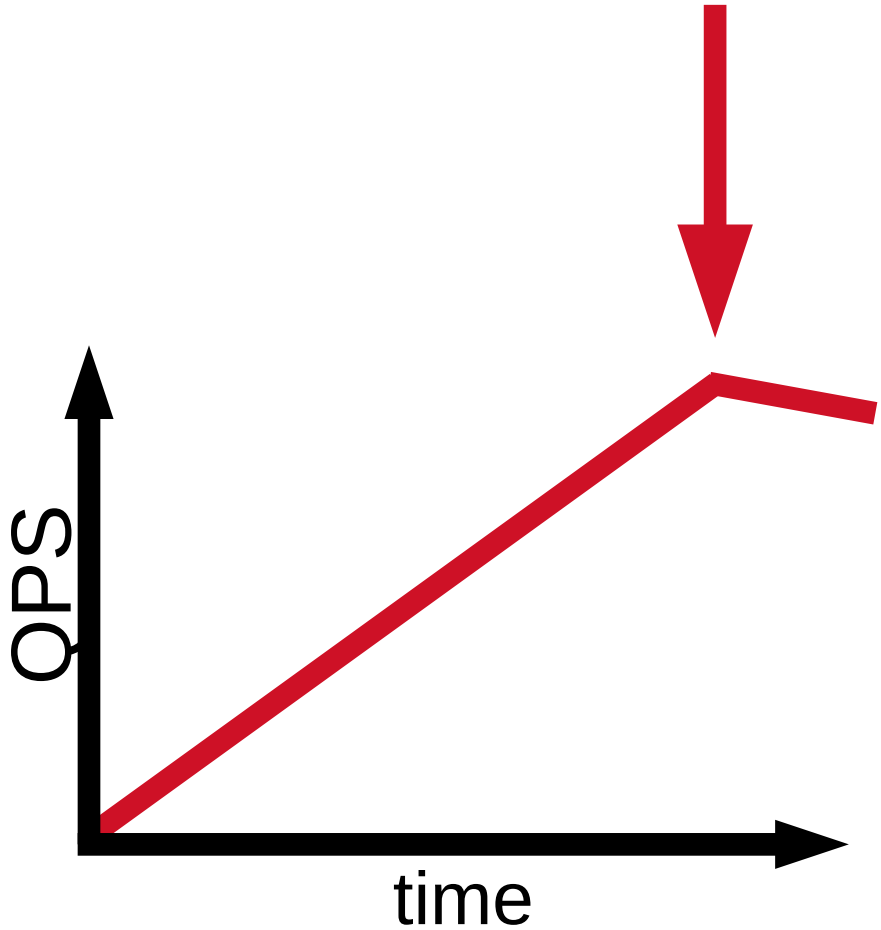
cz.nic | CZ DOMAIN REGISTRY

Outline

- Why not resperf
- DNS Shotgun introduction
- Measurement results
- Take aways

Resperf: QPS QPS QPS!

- `$ man resperf`
- Searches for max. QPS
- Text query list
  - tcpdump ⇒ text
  - **information loss**
    - timing, flags, EDNS, …



Why not resperf

- No timing information
  - ⇒ unrealistic cache hit rate
- No connection information
  - TCP, TLS, DoH!
- Ramp-up
  - ⇒ unrealistic
- **Over-focuses on QPS!**

DNS Shotgun: Introduction

- Toolset for **realistic** DNS resolver benchmarking
  - Work on user interface underway
- https://gitlab.nic.cz/knot/shotgun/
  - automation https://gitlab.nic.cz/knot/resolver-benchmarking/
  - based on https://github.com/DNS-OARC/dnsjit

COMCAST | INNOVATION FUND

DNS Shotgun: Different approach

- *How many clients can the resolver handle?*
  - Not just QPS

- Different clients = different behavior
  - IoT, mobil, desktop, mail server, …

DNS Shotgun: Principle

- Step 1: Analyze PCAPs
  - Slice traffic e.g. by source IP address
- Step 2: Simulate N clients
- Step 3: Compare results

DNS Shotgun: Client simulation

- DNS query stream replay
- Query timing ± 1 second
  - Realistic hit rate
  - ⇒ Variable QPS
- Configurable transport
  - TCP conn idle timer, TLS, HTTPS2, …

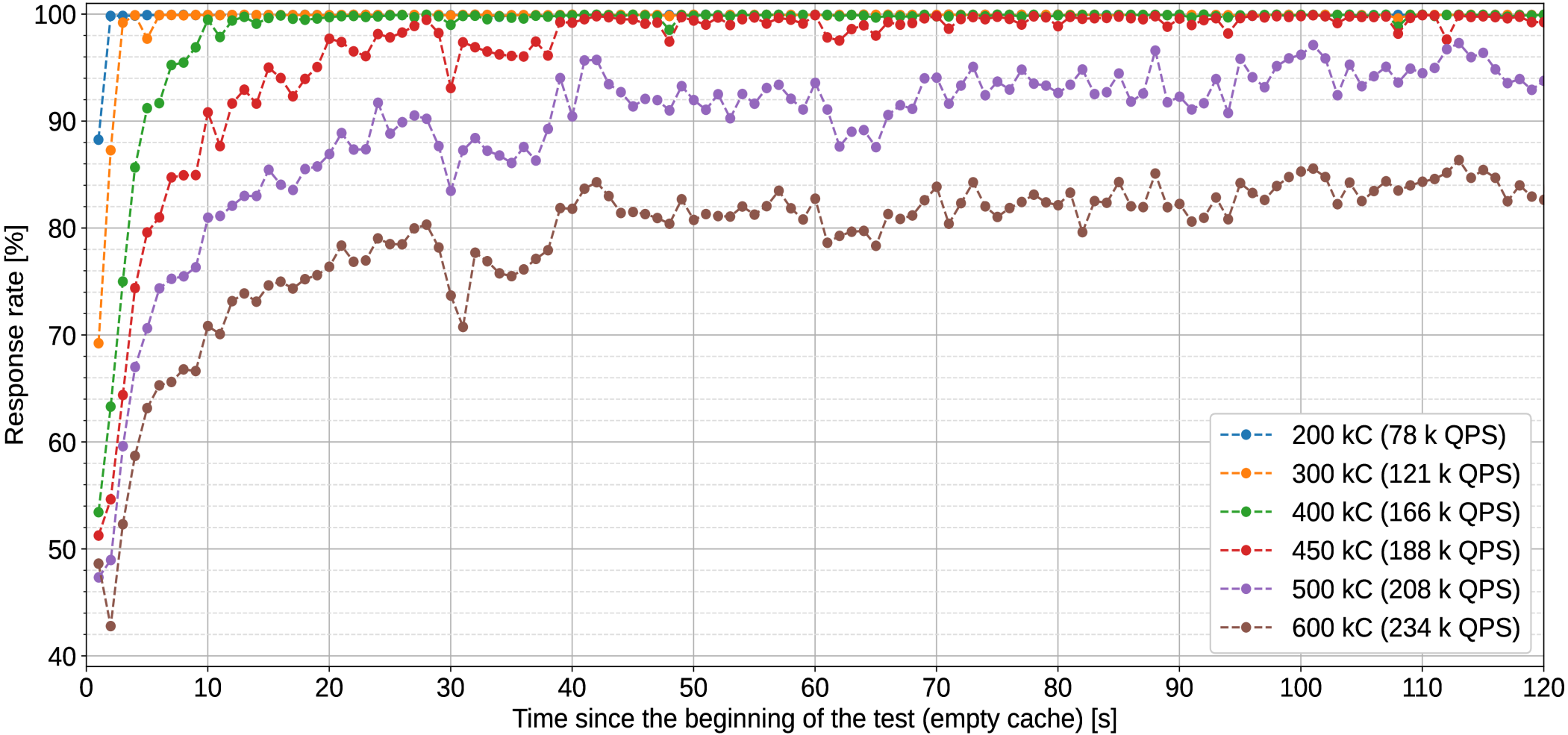
Experimental setup

- PCAP: university resolvers, normal traffic (no attacks)
- Empty cache
- Measurement length 120 s
- Monitor NOERROR/NXDOMAIN/SERVFAIL, latency
- **Increase number of clients**
- 8 CPU, 32 GB RAM, 10 ms latency client ⇔ resolver

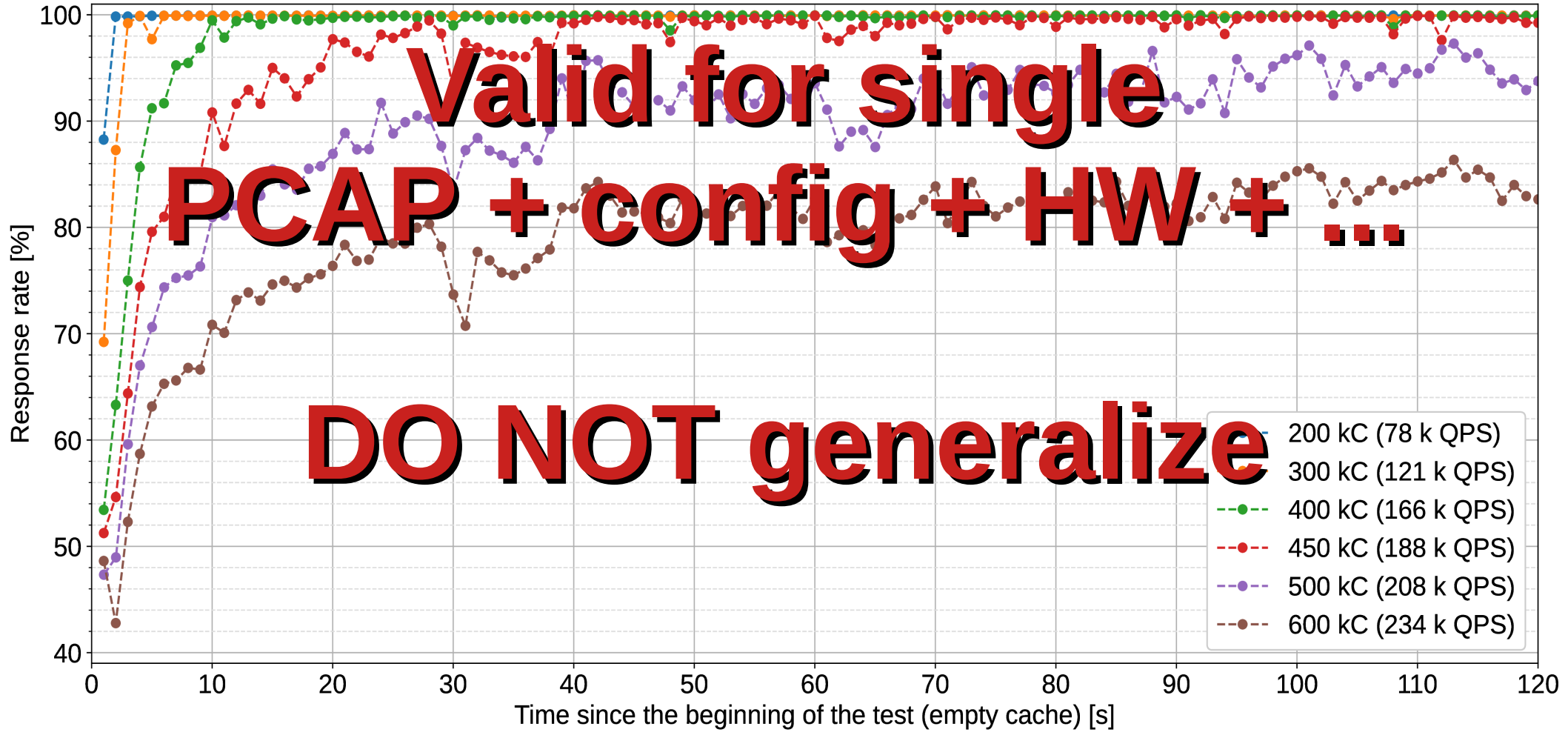
Performance tuning

- Resolver configuration
- ulimit -n
- https://www.kernel.org/doc/html/latest/networking/device_drivers/ethernet/intel/ixgb.html#improving-performance
- https://gitlab.nic.cz/knot/resolver-benchmarking/-/tree/master/roles/tuning

Knot Resolver 5.2.0 dev, UDP



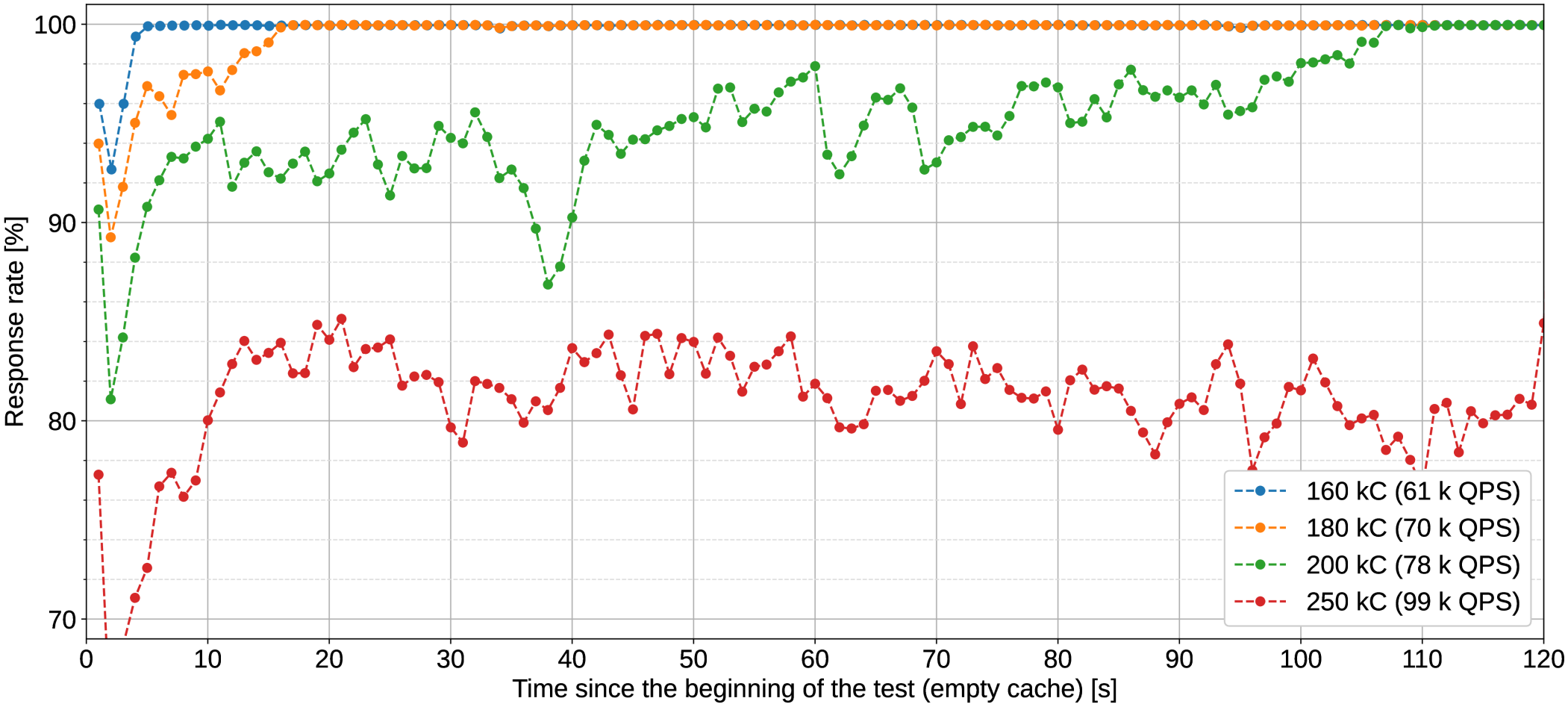
Knot Resolver 5.2.0 dev, UDP



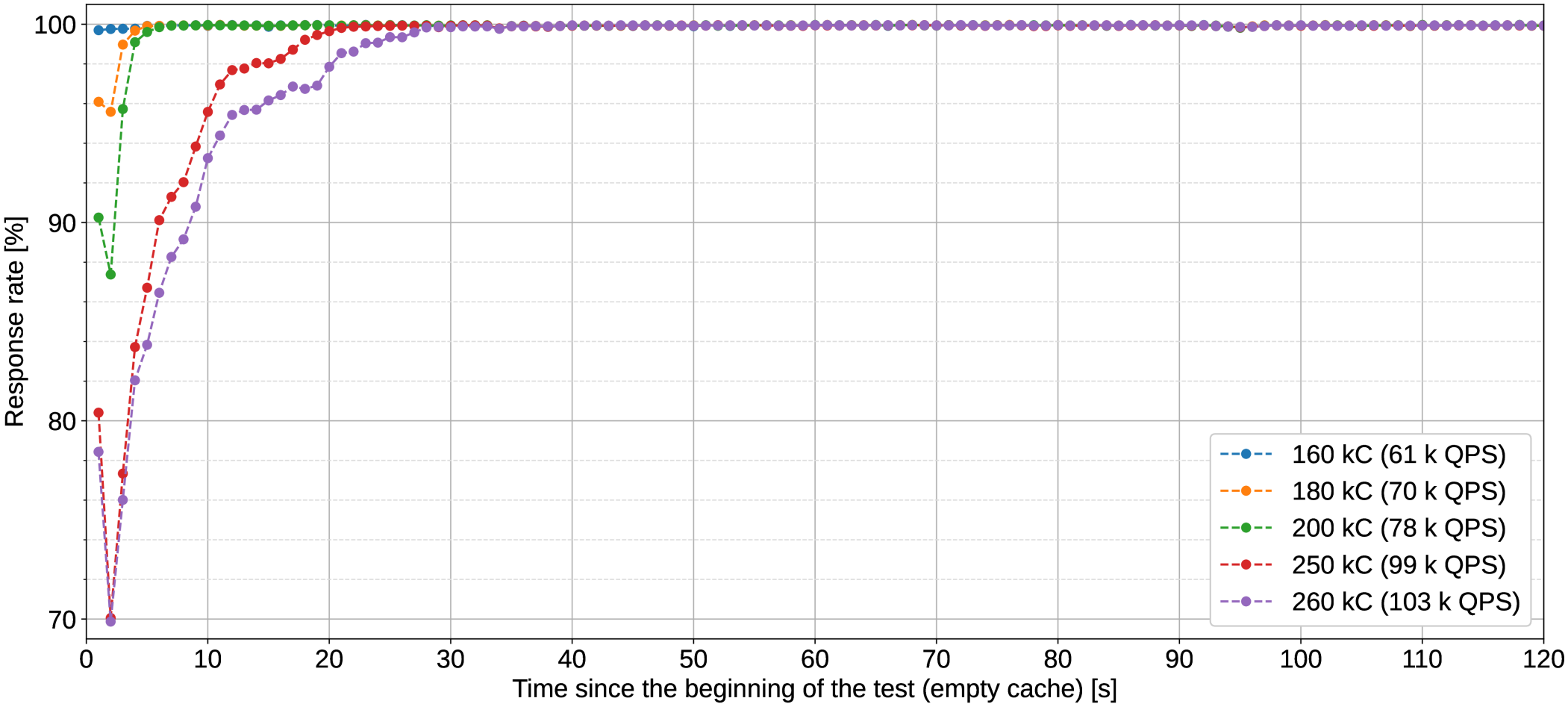
TCP setup

- Multiple source IP addresses
  - for clients / the Shotgun machine
- Resolver config
  - tcp-clients / max-tcp-clients / incoming-num-tcp / etc.
- Idle timer

Knot Resolver 5.2.0 dev, TCP, no idle connections



Knot Resolver 5.2.0 dev, TCP, idle limit 10 s



Knot Resolver 5.2.0 dev, UDP – TCP

Protocol	Clients served	Performance loss to UDP
UDP	400 k	1
TCP no idle connections	180 k	2.2
TCP idle limit 10 s	250 k	1.6

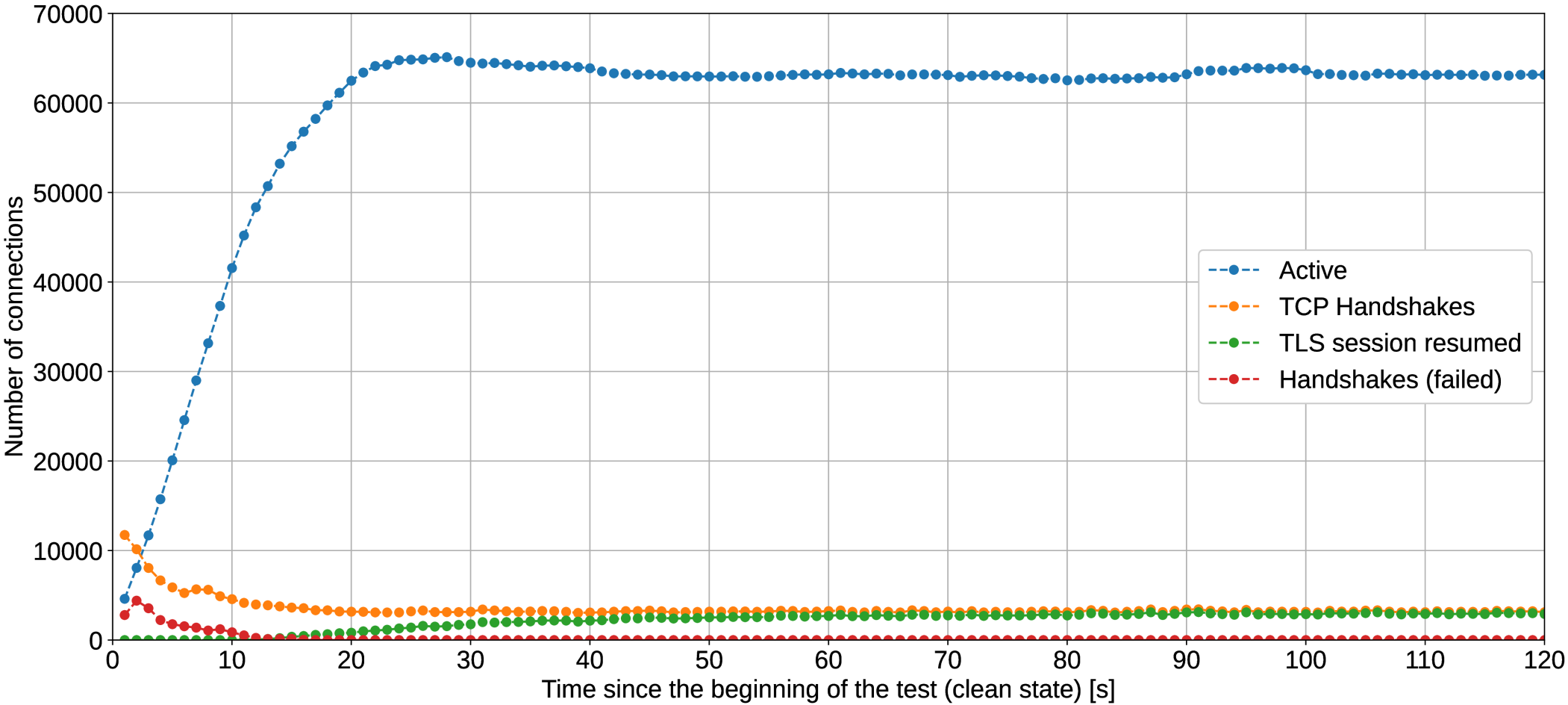
TLS setup

- TLS version
  - 1.3
- Cert signing algorithm
  - RSA 2k, RSA 3k, P256, Ed25519, …
- TLS session resumption
  - New handshake – latency, CPU
  - TCP idle timer – memory

UDP – TCP – TLS, idle limit 10 s

Protocol	Clients served	Performance loss to UDP
UDP	400 k	1
TCP, idle limit 10 s	250 k	1.6
TLS RSA 2k	80 k	5
TLS RSA 3k	20 k	20
TLS P256	140 k	2.9
TLS Ed25519	140 k	2.9

TLS connections, idle limit 10 s, 140 kC



UDP – TCP – TLS, no idle connections

Protocol	Clients served	Performance loss to UDP
UDP	400 k	1
TCP	180 k	2.2
TLS Ed25519	40 k	10

DoH

- TCP + TLS + HTTP
- HTTP version
  - version 2 recommended
  - GET / POST
- HTTP compression
- Headers
  - Content-type …
- Tons of additional options

UDP – TCP – TLS – HTTP2, idle limit 10 s

Protocol	Clients served	Performance loss to UDP
UDP	400 k	1
TCP	250 k	1.6
TLS Ed25519	140 k	2.9
HTTP2 + TLS Ed25519	120 k	3.3

Other resolvers

- BIND 9.16.6
  - no DoT or DoH
- PowerDNS 4.3.4 + dnsdist 1.5.0
  - problems measuring DoT and DoH
- Unbound
  - DoT 1.11.0
  - DoH https://github.com/NLnetLabs/unbound/pull/255

All resolvers: Thousands clients served

- Better ignore the absolute values!

Protocol	BIND	Knot Resolver	PowerDNS	Unbound
UDP	160	400	250	450
TCP 0s	80	180	120	250
TCP 10s	120	250	140	300
TLS Ed25519		140		160
HTTP2 + TLS		120		140

All resolvers: Protocol performance penalty

- Compared to max. UDP throughput of a given resolver

Protocol	BIND	Knot Resolver	PowerDNS	Unbound
UDP	1	1	1	1
TCP 0s	2	2.2	2.1	1.8
TCP 10s	1.3	1.6	1.8	1.5
TLS Ed25519		2.9		2.8
HTTP2 + TLS		3.3		3.2

Take aways

- TLS – use P256 / Ed25519, avoid RSA
- Protocol performance penalty
  - Similar across implementations?
- **Do not generalize – client populations differ**
- **Measure it yourself**
  - https://gitlab.nic.cz/knot/shotgun
- Need help measuring? Contact us
  - knot-resolver@labs.nic.cz