Persistent DNS connections for improved performance

Submitted to IFIP Networking 2019

Baptiste Jonglez, Sinan Birbalta, Martin Heusse

18 January 2019

Université Grenoble Alpes, Laboratoire d'Informatique de Grenoble
PhD advisors: Martin Heusse, Bruno Gaujal

DNS and latency

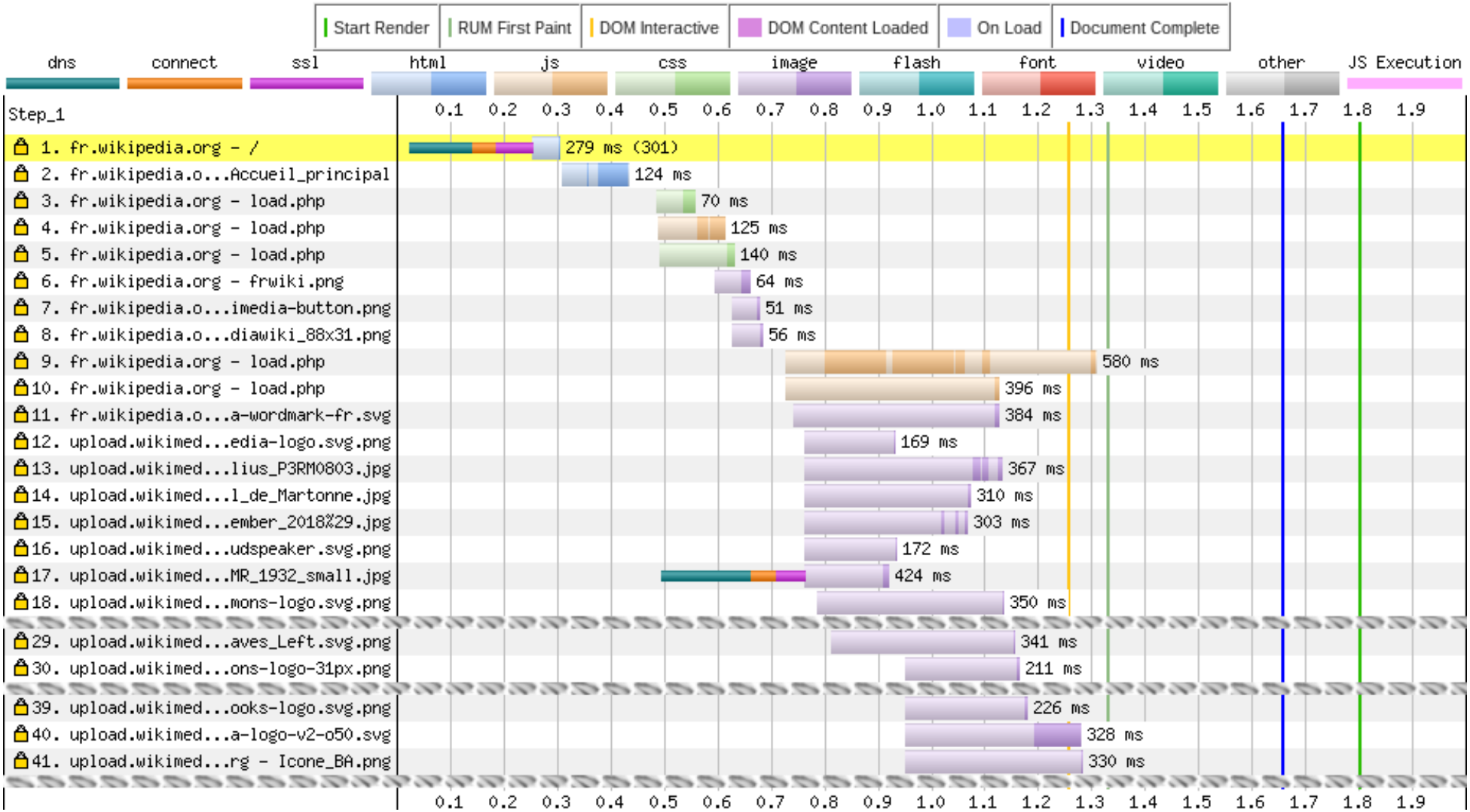


Figure: Loading `fr.wikipedia.org` using webpagetest.org from Strasbourg (5Mbps/1Mbps connection with 28 ms RTT, chrome). DNS took 312 ms, that is 24% of “Time to First Interactive”.

Why does DNS suffer from latency?

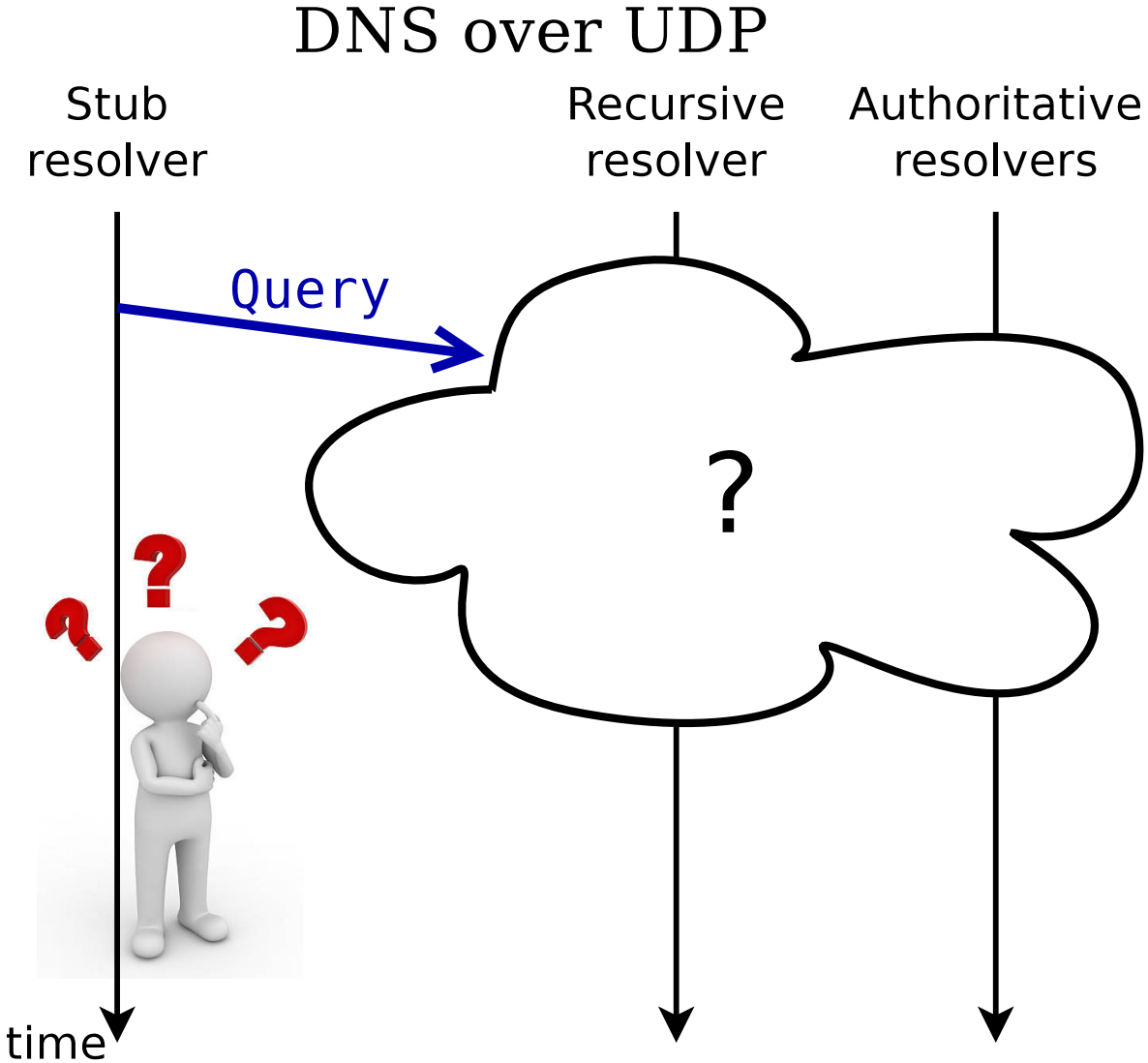


Figure: DNS resolution seen from the stub resolver.

Why does DNS suffer from latency?

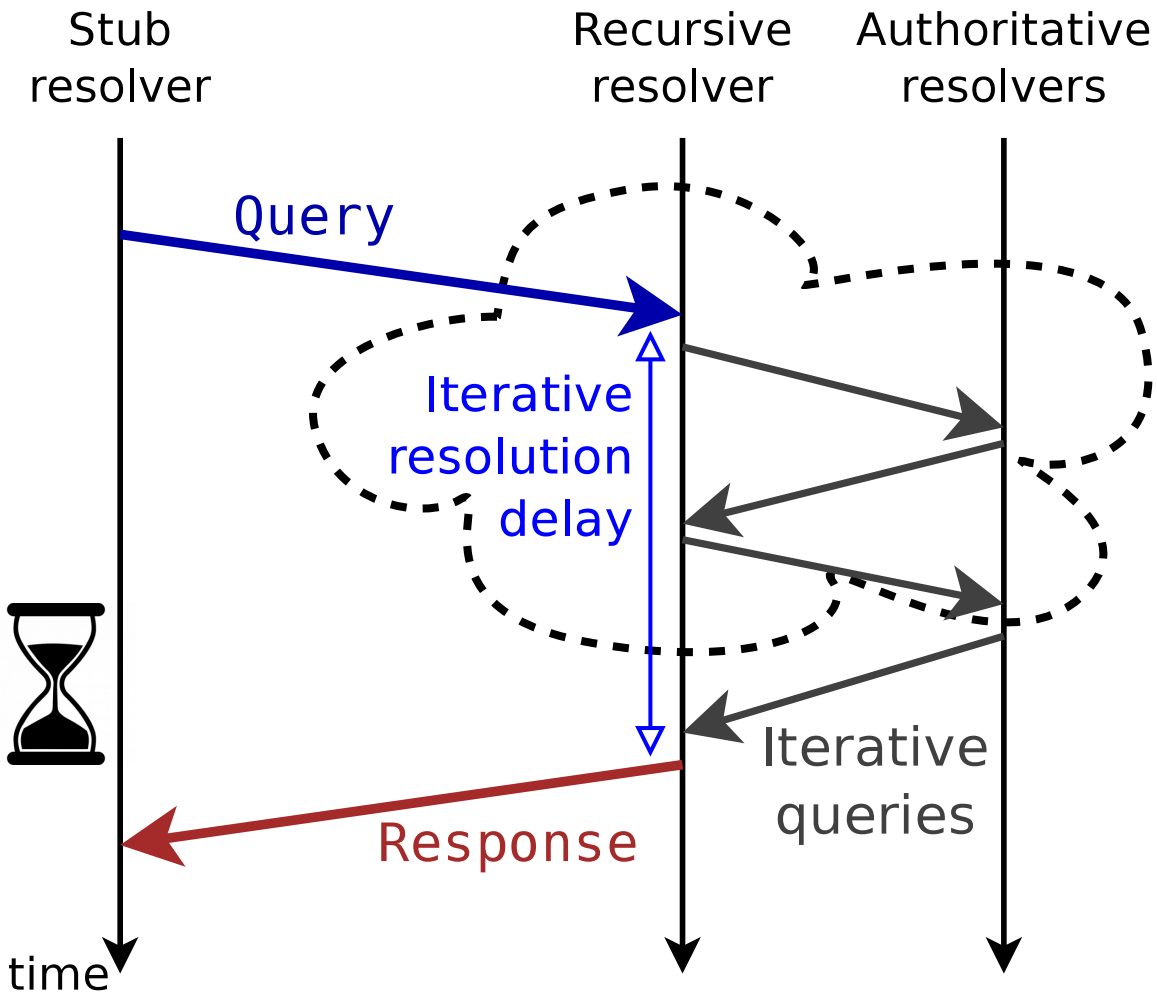


Figure: Iterative DNS resolution process.

Why does DNS suffer from latency?

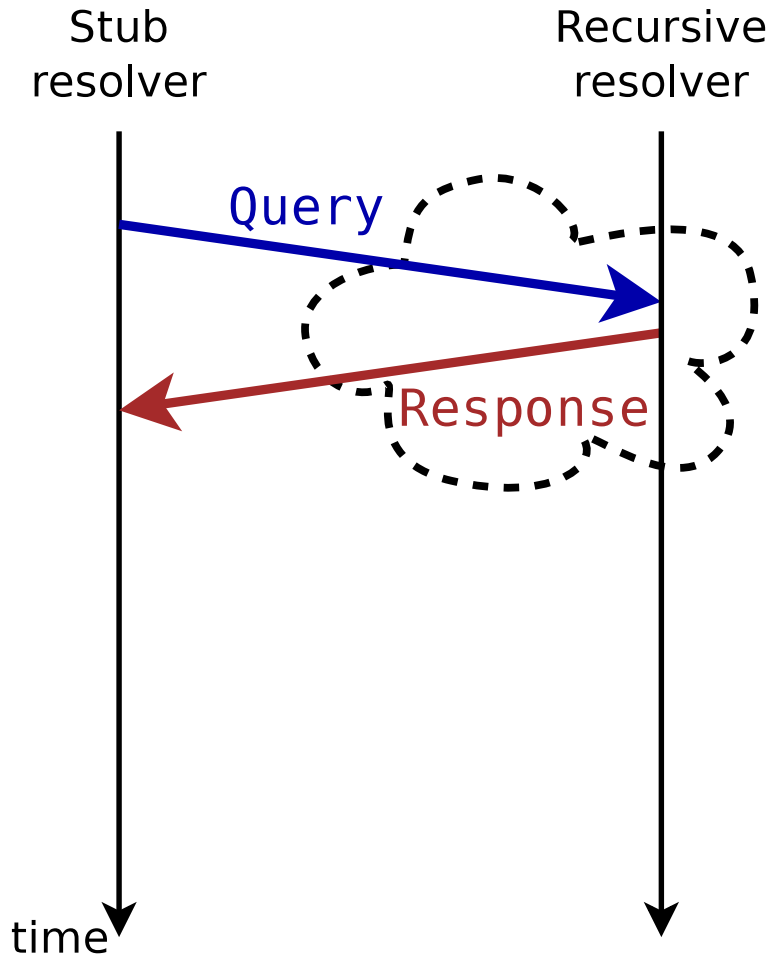
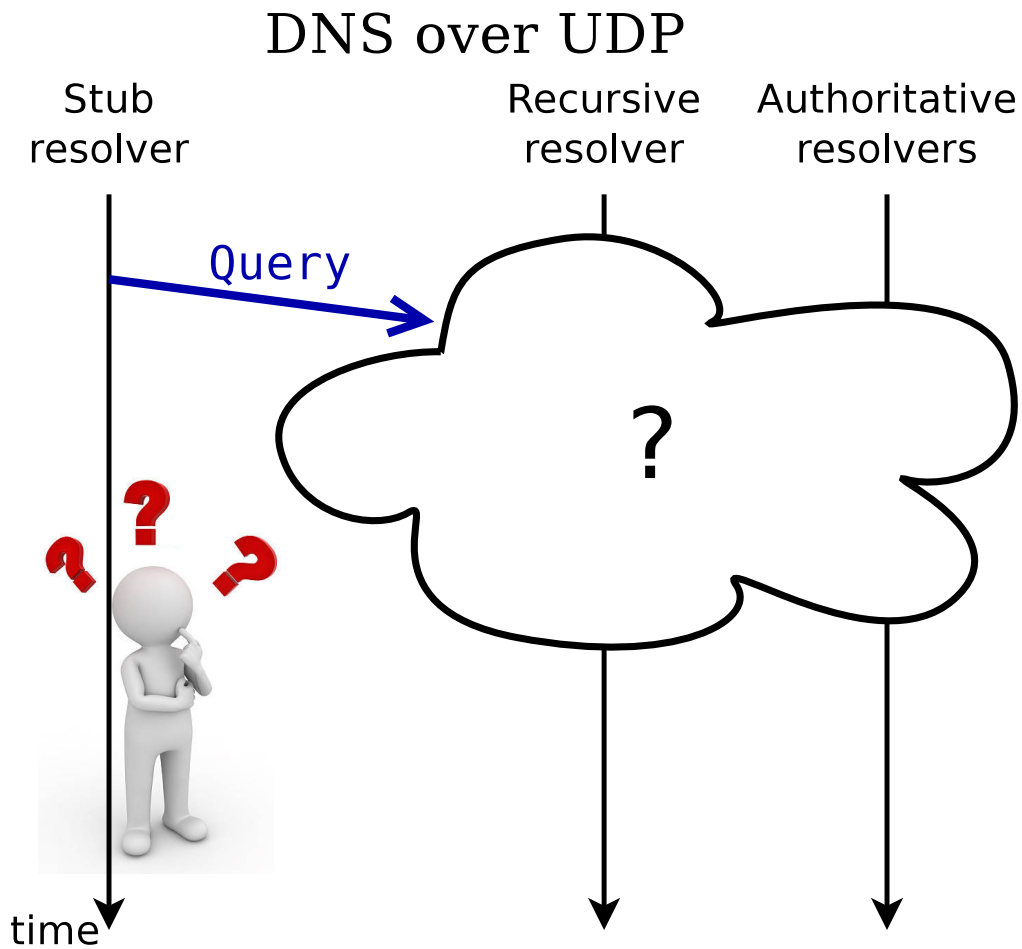


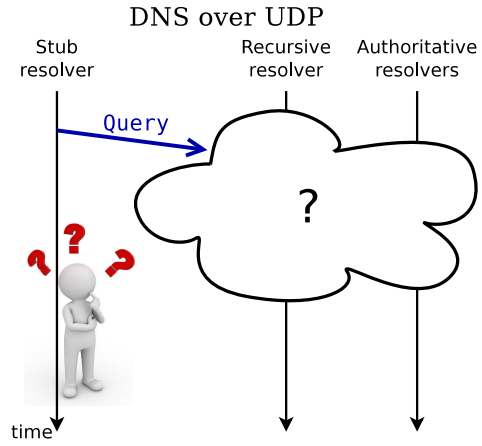
Figure: DNS response served from cache.

When things go wrong



But what if no response comes back?

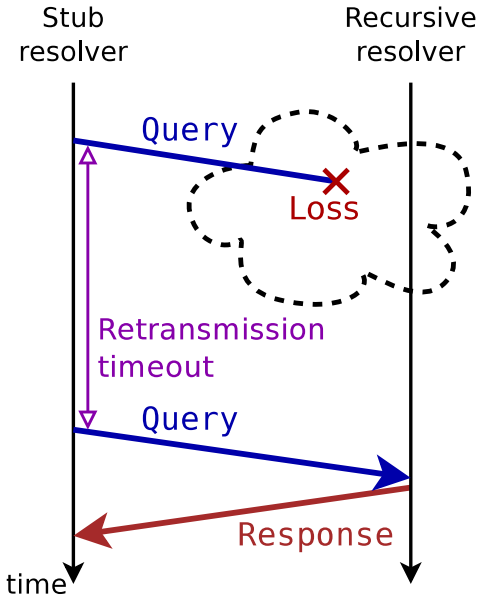
When things go wrong

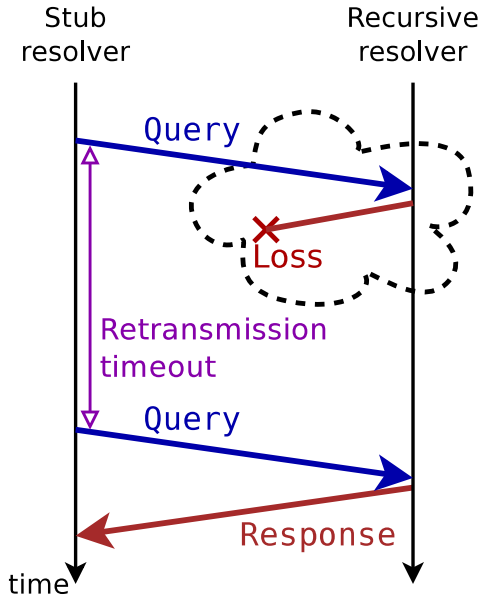


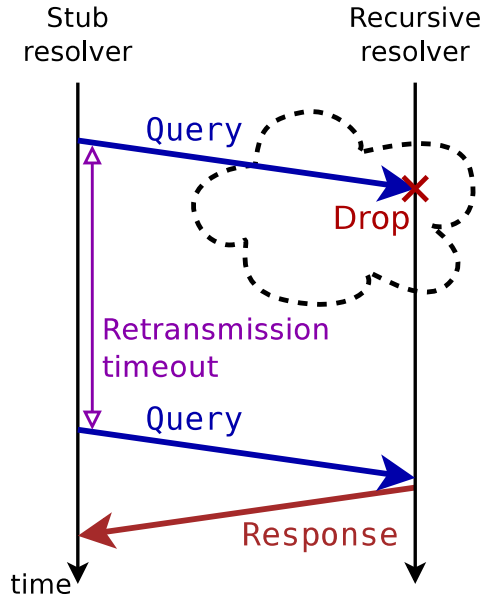
Stub resolver	First retrans. timeout	Retransmission strategy	Time before application failure
Glibc 2.24 (Linux)	5 seconds	Constant interval	40 seconds
Bionic (android 7.1)	5 seconds	Constant interval	30 seconds
Windows 10	1 second	Exponential backoff	12 seconds
OS X 10.13.6	1 second	Exponential backoff	30 seconds
IOS 11.4	1 second	Exponential backoff	30 seconds

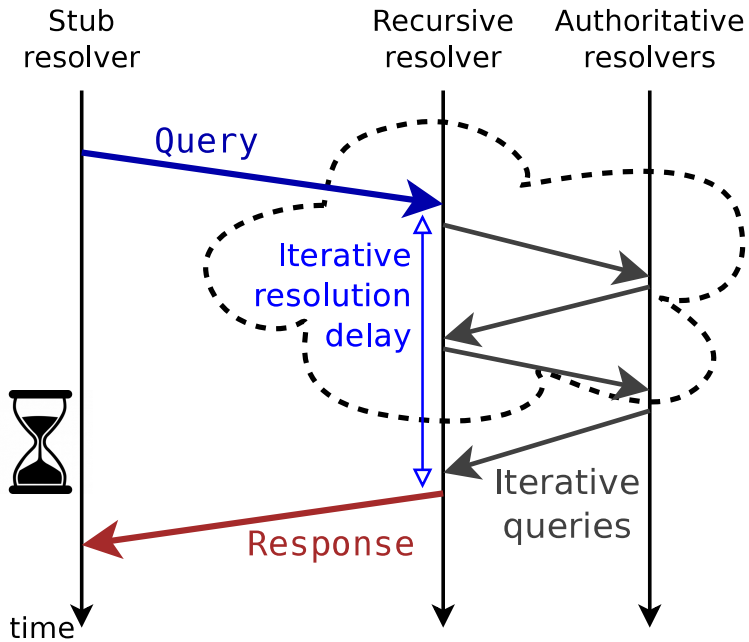
Table: Retransmission behaviour of widely used stub resolvers.

What could have gone wrong?









No way to distinguish the four cases!

Persistent DNS connections to the rescue

Persistent connections

- Replace UDP with TCP, TLS, QUIC...
- Reuse connection for several queries
- Gain: decouples application from transport

The network security community hates UDP anyway!

Decoupling application from transport

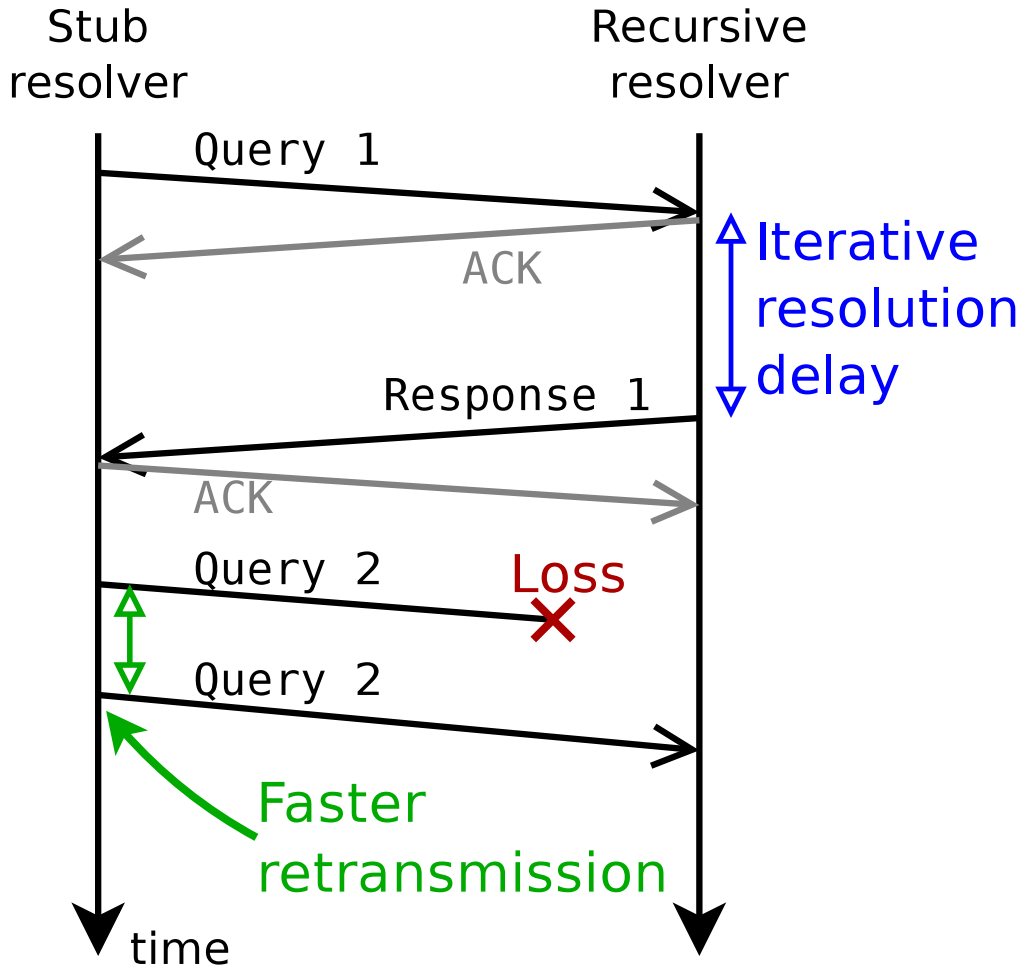


Figure: DNS over TCP: retransmission can happen much faster thanks to RTT estimation.

Contributions

Contributions

- We show that DNS-over-TCP can yield lower latency than UDP (testbed experiment)
- We study the performance impact of TCP/TLS on recursive resolvers (large-scale experient on Grid'5000) and find it is manageable

First experiment: focus on latency

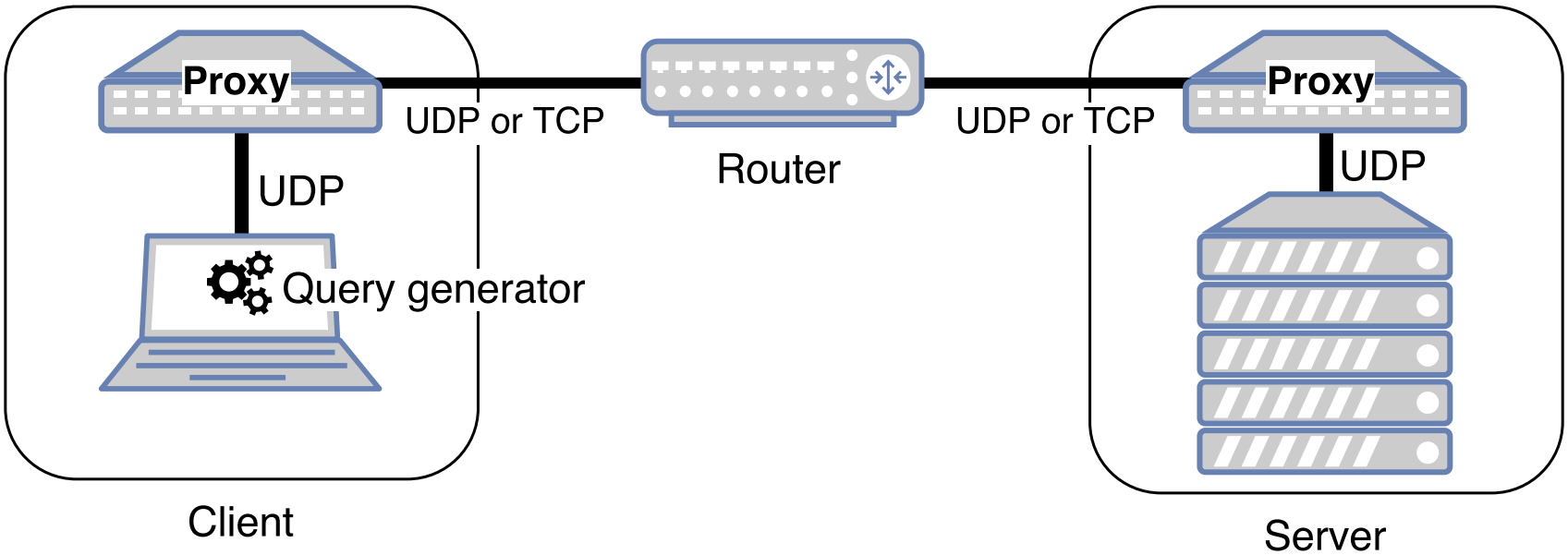


Figure: Experimental platform to compare UDP and TCP.

First experiment: focus on latency

Client parameters

- Inter-query time distribution (between 50 ms and 300 ms)
- Number of queries sent simultaneously (1, 3)
- Retransmission timeout for UDP (3s)
- TCP variants: Early Retransmit, Tail Loss Probe, Low Latency, Thin Linear Timeout...

Network parameters

- Emulated loss (0.5%, 1%, 2%, 5%, 10%)
- Emulated delay (RTT of 20 ms, 60 ms, 200 ms)

First experiment: latency results

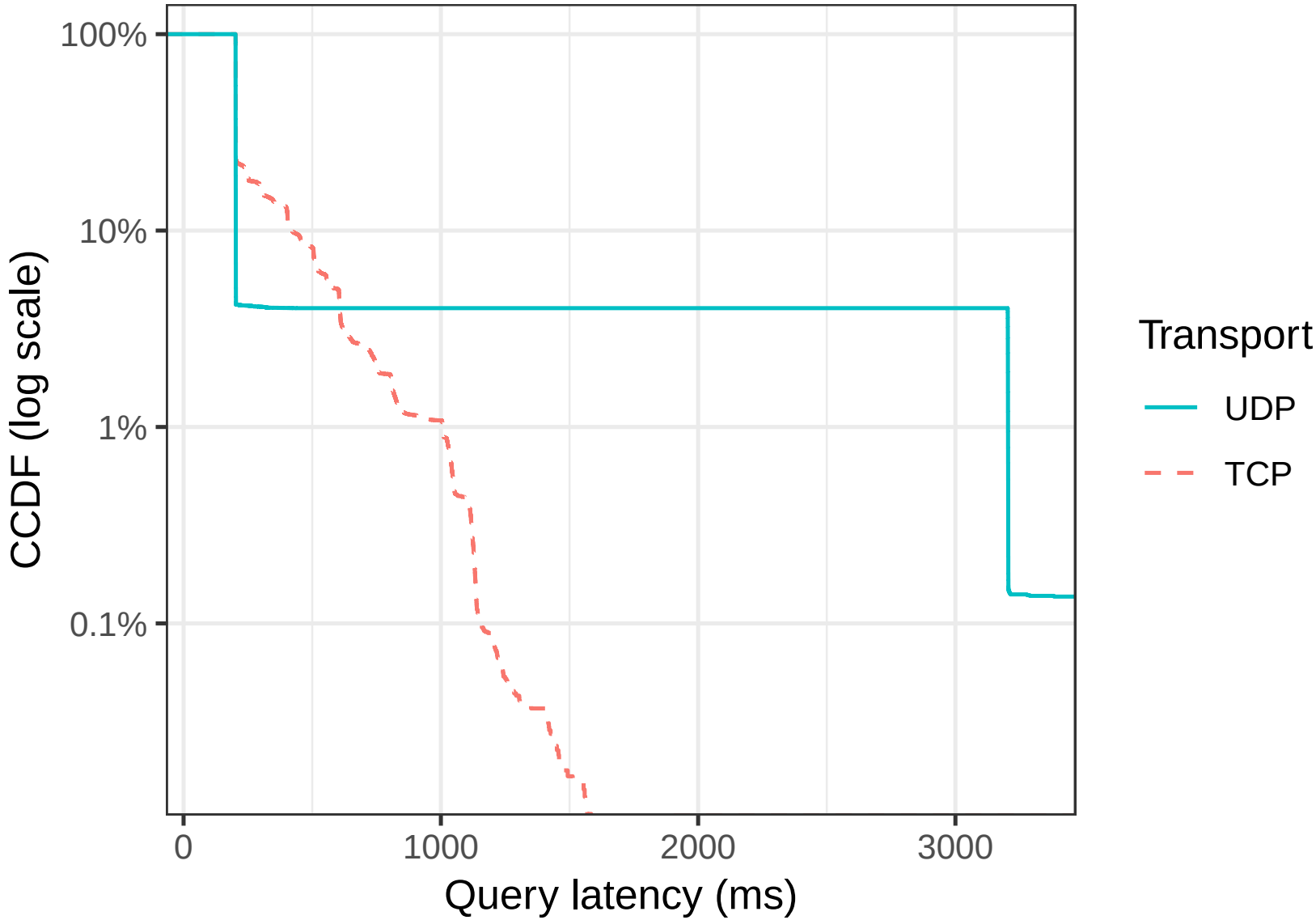


Figure: 2% packet loss, 200 ms RTT

First experiment: head-of-line blocking

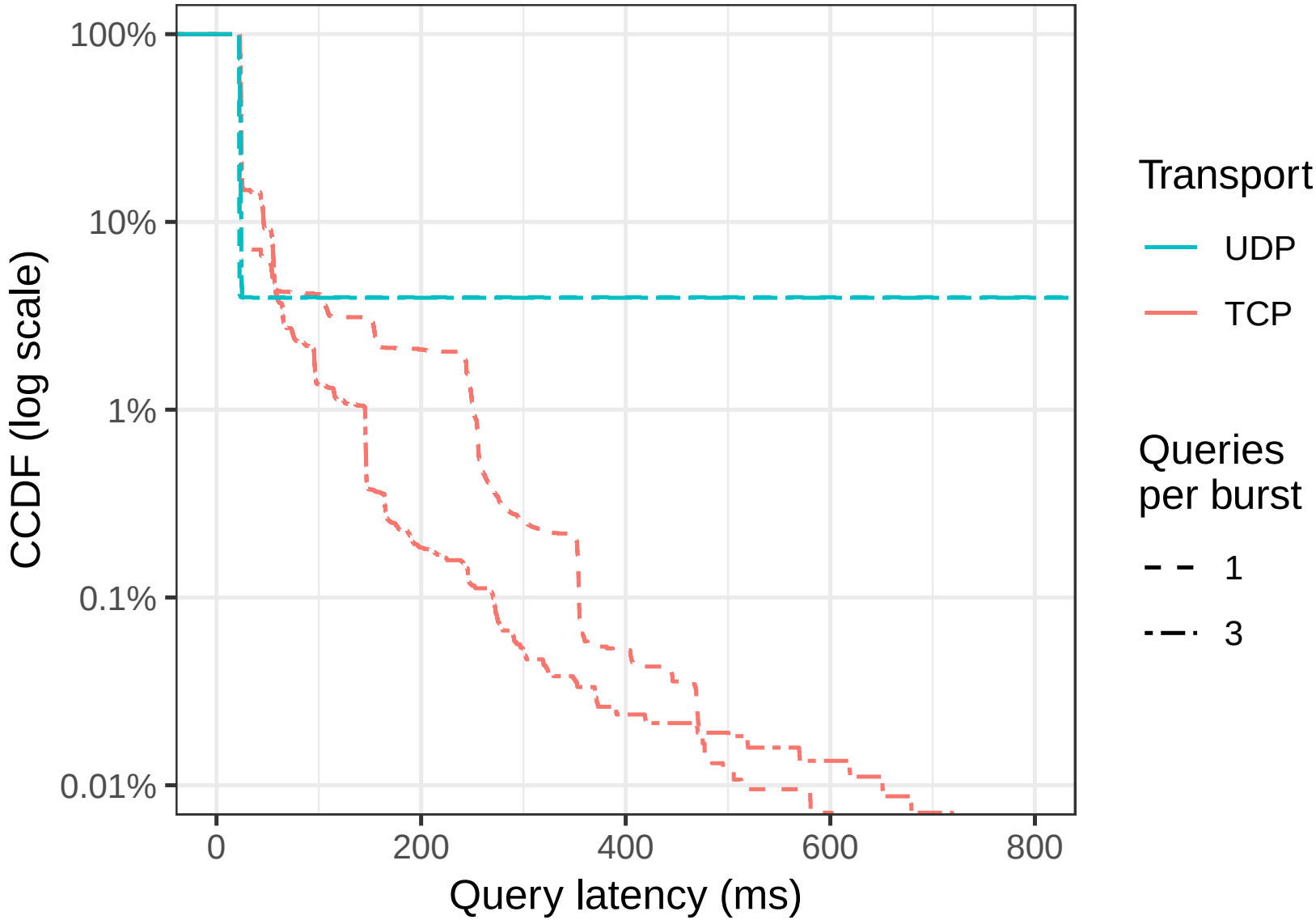


Figure: 2% packet loss, 20 ms RTT

First experiment: main results

Main results

- TCP reduces worst case latency: p99 reduced from 3200 ms to 1006 ms, p99.9 reduced from 6200 ms to 1157 ms
- head-of-line blocking issue, especially when the RTT is much larger than the inter-query time
- TCP variants have no significant impact!

Further work: DNS-over-QUIC to avoid head-of-line blocking

From theory to real-world deployment

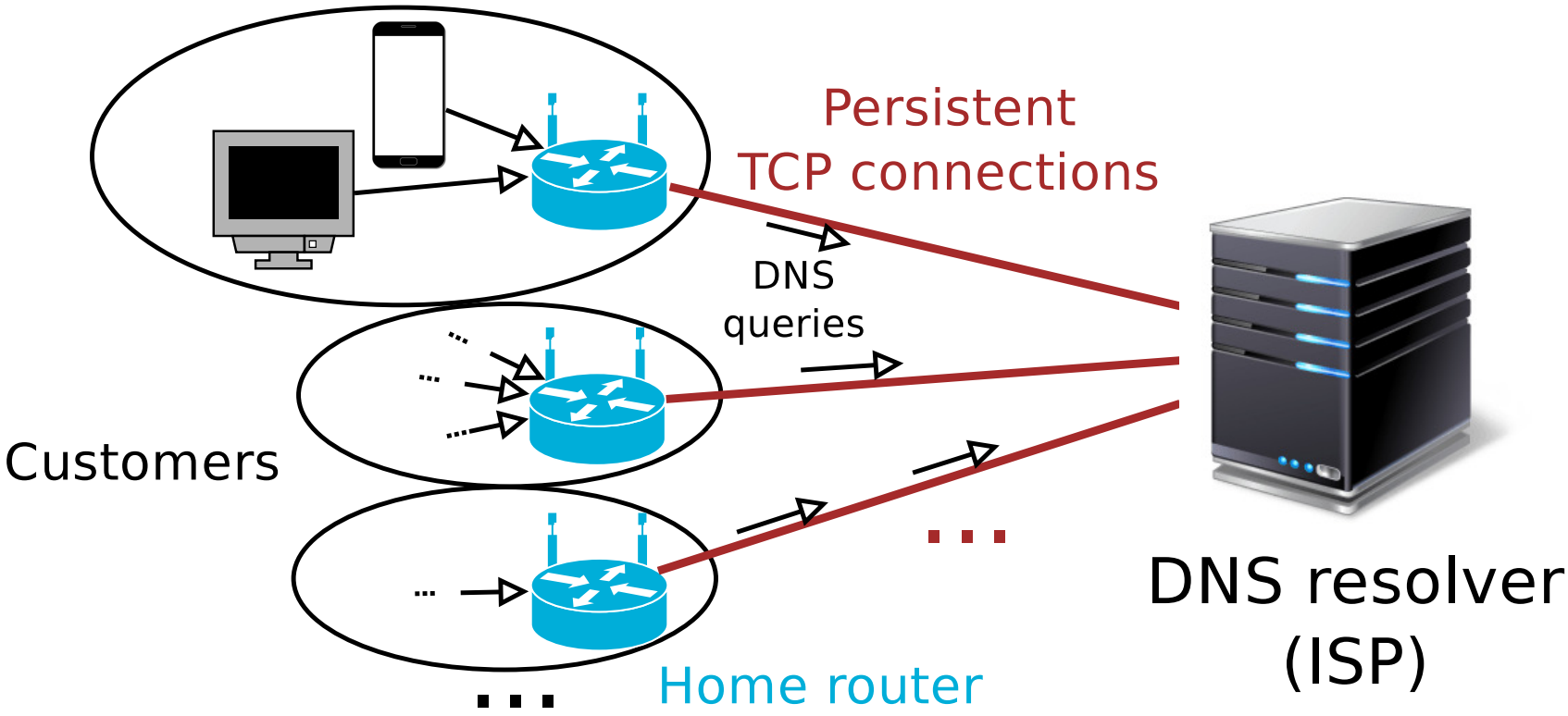


Figure: Deployment model of persistent DNS connections.

Second experiment: large-scale

Experiment goals and challenges

- Analyze the performance impact of persistent connections on **recursive resolvers**;
- Compare UDP, TCP, TLS;
- Large-scale: millions of DNS clients;
- No simulation: real recursive resolver software.

Grid'5000 fits all the needs!

Second experiment: large-scale

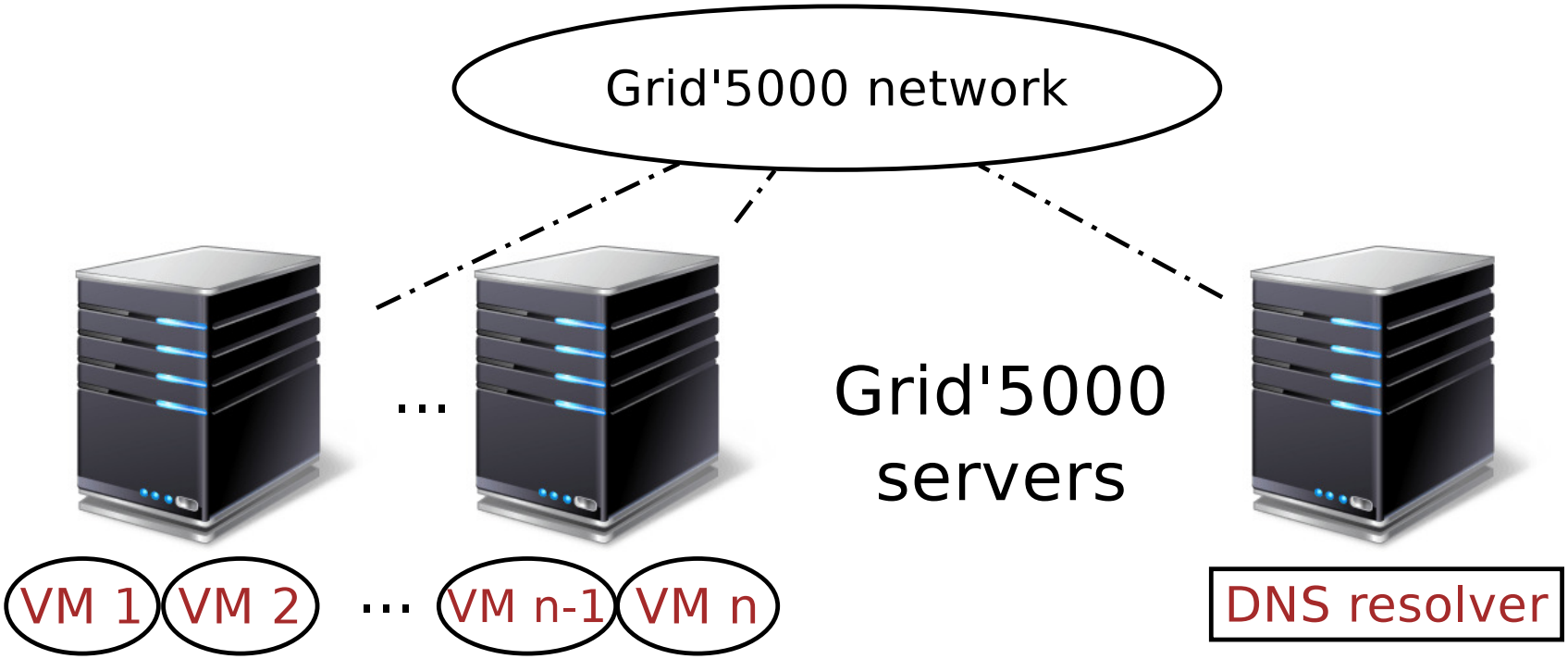


Figure: Practical setup using Grid'5000. Each VM opens several persistent connections to the recursive resolver.

Methodology: peak performance estimation

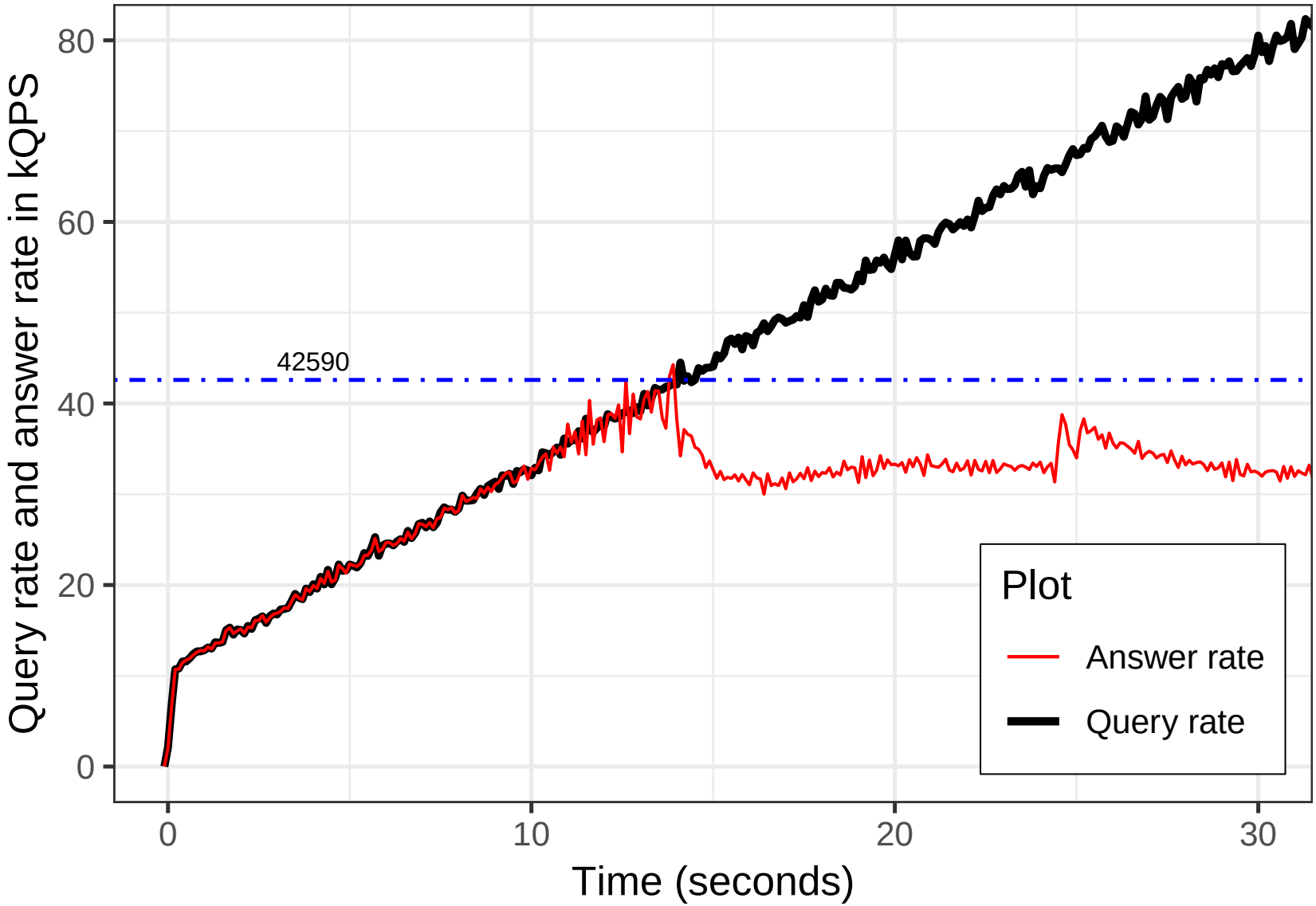


Figure: `unbound` with 1 thread, 24 VMs, 250 TLS connections per VM.

Methodology: peak performance estimation

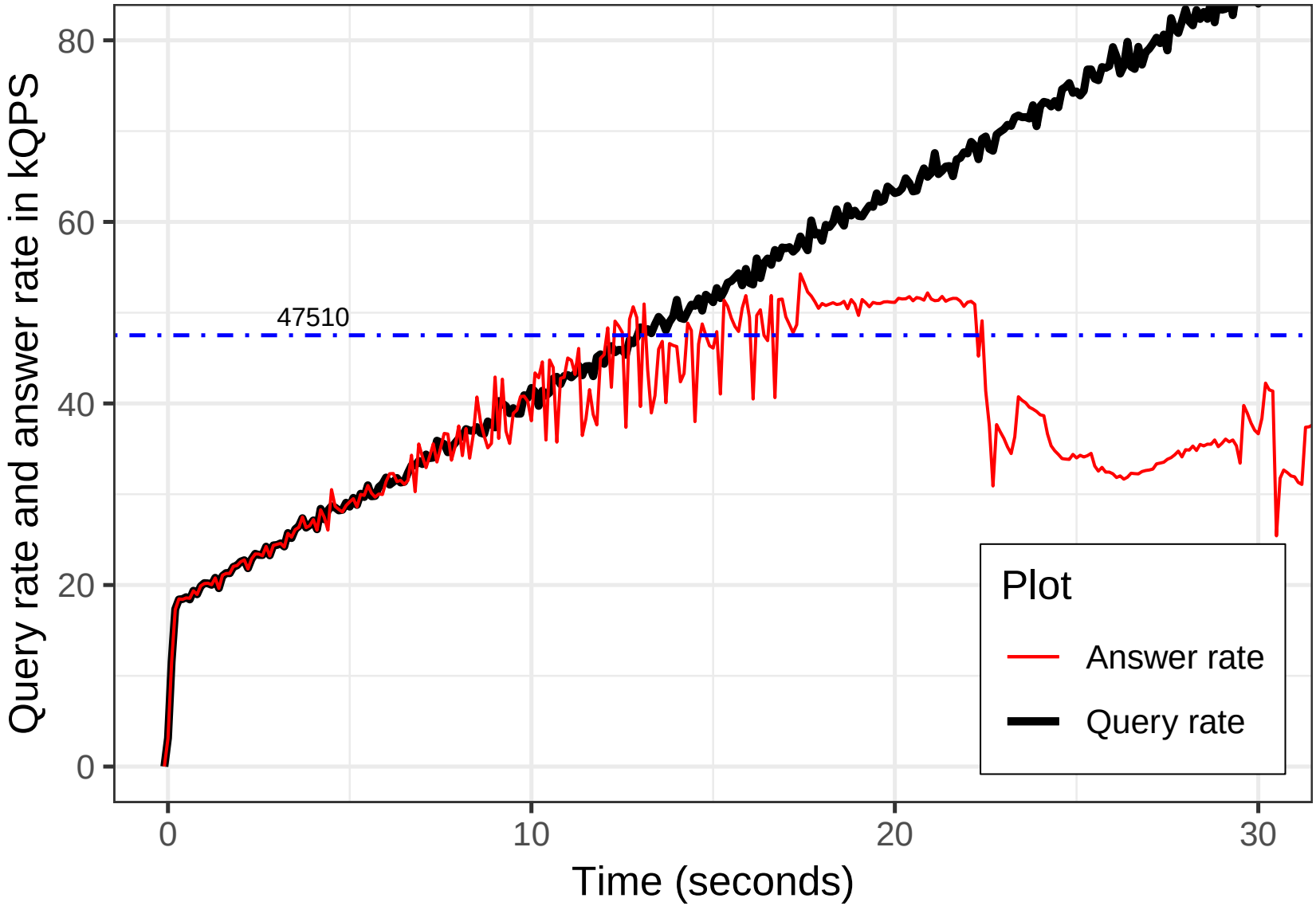


Figure: Bind with 1 thread, 24 VMs, 125 TCP connections per VM.

Main results

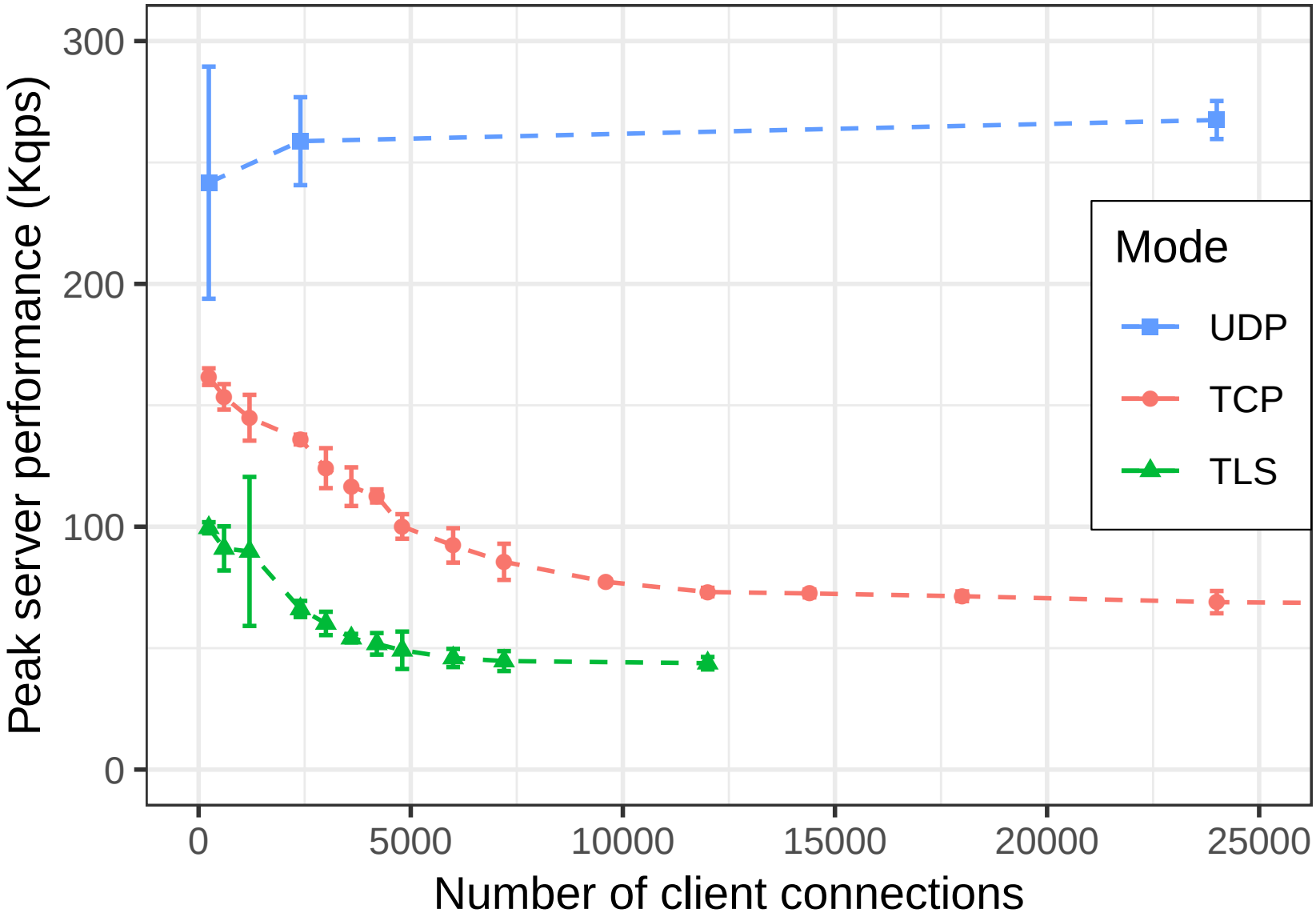


Figure: Performance comparison of UDP, TCP, TLS (unbound).

Main results

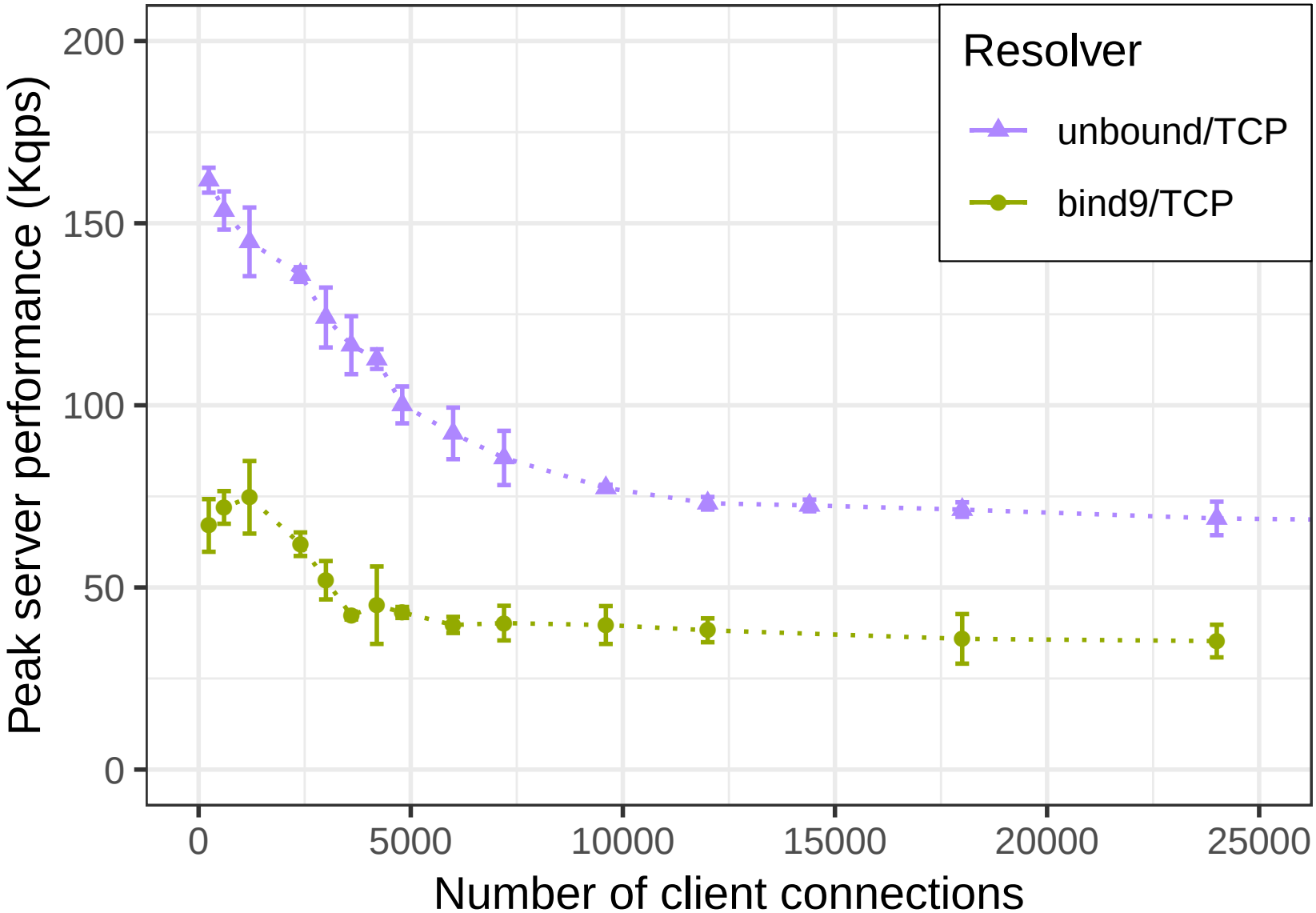


Figure: Performance comparison of bind and unbound (TCP).

Main results

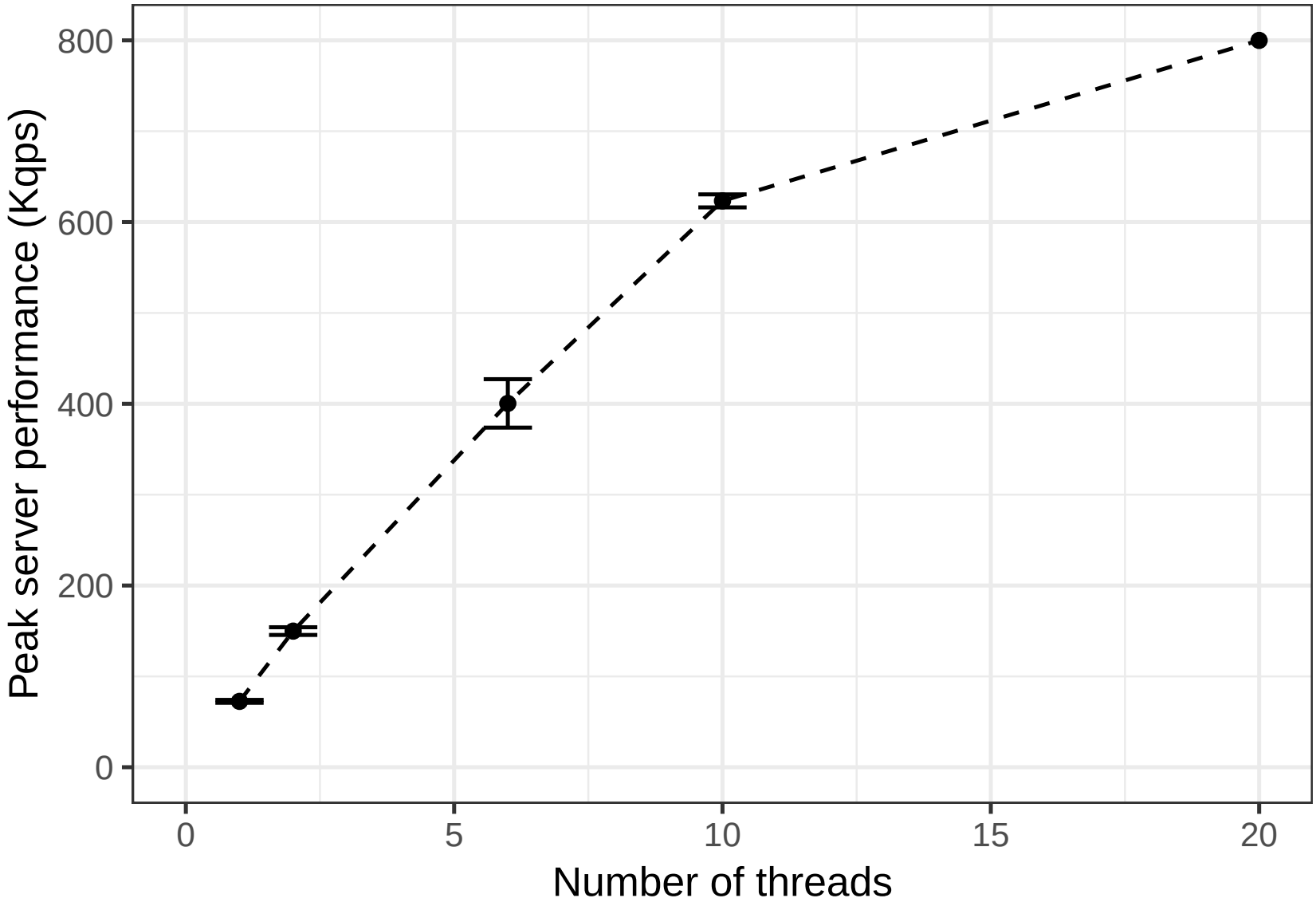


Figure: Scaling on multiple CPU cores.

Conclusion

Conclusion

- Persistent DNS connections can reduce latency on lossy networks
- Recursive resolver performance is manageable, even with TLS (but see below)
- Grid'5000 is useful for large-scale, scripted experiments

Sharp edges

- client-side: Head-of-line blocking with TCP and TLS
- server-side: cost of new TLS sessions (churn)

Thank you!

Latency

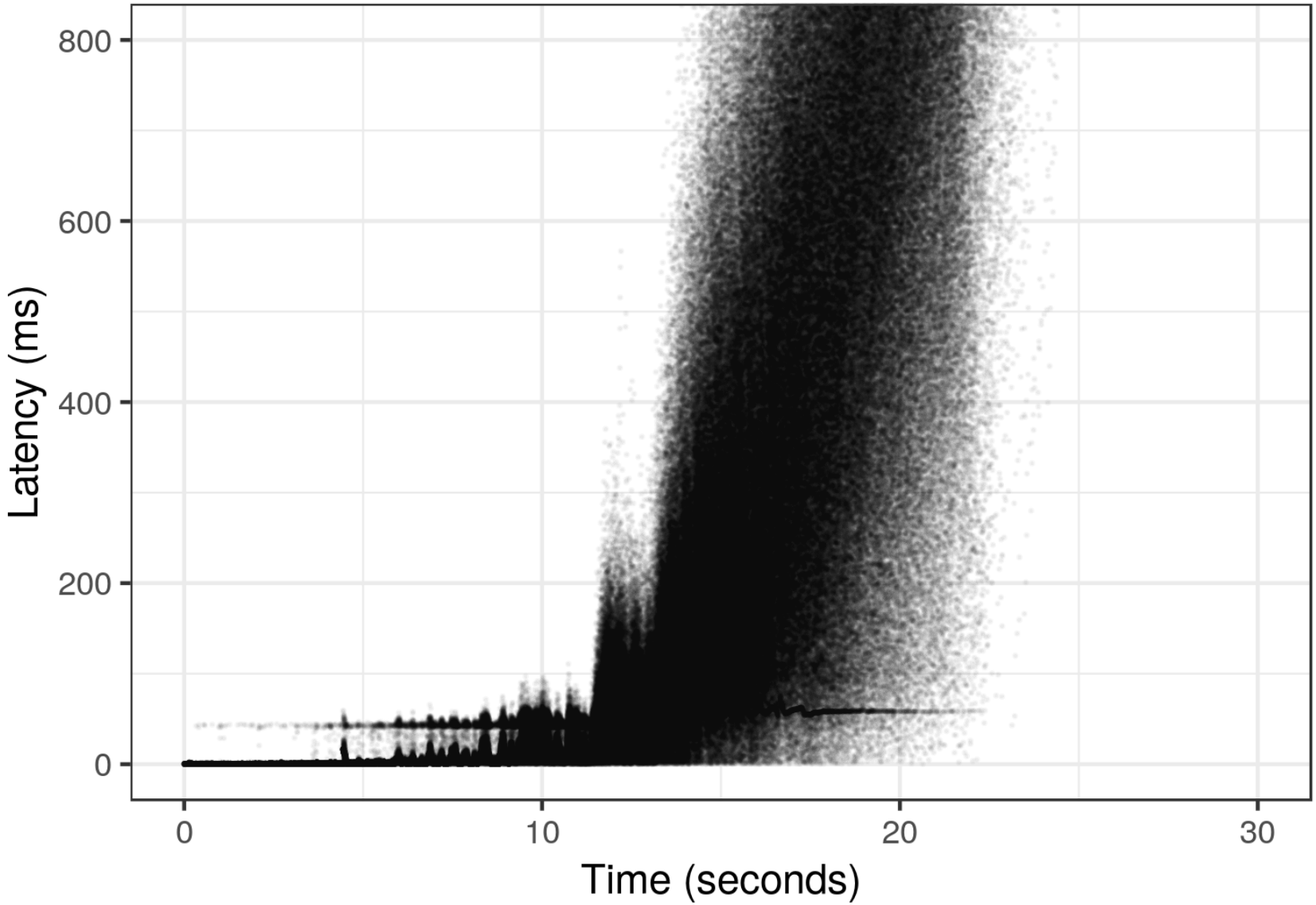


Figure: Latency of each query during an experiment (Bind/TCP)