Persistent DNS connections for improved performance

Baptiste JONGLEZ, Sinan BIRBALTA, Martin HEUSSE
Univ. Grenoble Alpes, CNRS, Grenoble INP, LIG, 38000 Grenoble, France

THE CASE FOR DNS-OVER-TLS

DNS runs primarily over UDP today, but this raises security and privacy issues:
1. No source address validation, enabling massive DDoS reflection attacks;
2. Unreliable IP fragmentation, which is required when sending large DNS answers;
3. No encryption, allowing pervasive surveillance and censorship through DNS traffic manipulation.

DNS-over-TLS and DNS-over-HTTPS [2], as recently specified, both solve all these problems. We focus on DNS-over-TLS between stub resolver and recursive resolver, and study its performance at scale.

CHALLENGES

Deploying DNS-over-TLS at scale comes with a number of challenges. The cost of opening a new TLS connection is significant, both in terms of latency and CPU load. Persistent connections thus become necessary to amortize this cost. However, it means that servers need to manage tens of thousands of TLS connections, consuming resources.

We focus on the following questions:
1. What is the impact of DNS-over-TLS on query latency?
2. What is the performance impact of large-scale DNS-over-TLS on recursive resolvers?

This poster presents experiments which answer these two questions.

PERSISTENT CONNECTIONS CAN IMPROVE LATENCY

- **Without losses**: once established, a persistent connection provides the same latency as UDP. 1 RTT [3].
- **With losses**: UDP performs very poorly and exhibits large latency. A stub resolver has to guess whether the iterative resolution process is taking longer than expected (left), or if a message was lost and a retransmission is required (middle). Most implementation use a simple timeout:
  - Constant interval
  - Constant interval
  - Exponential backoff
  - Exponential backoff

With a persistent connection, the stub resolver can measure the RTT and retransmit much more efficiently (right):

RESULT 1: UDP vs. TCP

![UDP vs. TCP graph]

RESULT 2: BIND vs. UNBOUND

![BIND vs. UNBOUND graph]

RESULT 3: MULTICORE SCALING

![MULTICORE SCALING graph]

CONCLUSION

- DNS-over-TLS runs at about 20% of the performance of DNS-over-UDP;
- Performance scales linearly with the number of CPU cores, up to a hardware limit;
- With lots of CPU cores, performance gets close to UDP (common bottleneck);
- DNS-over-TLS improves worst-case latency when messages are lost;

Further work: study the impact of client churn, alleviate the head-of-line blocking problem.

REFERENCES

