

# DNS Transport Signaling

## Avoiding the Chicken-and-Egg Problem

Erik Bergström   Leon Fernandez   Johan Stenstam

The Swedish Internet Foundation

September 22, 2025

# Problem Statement

The goal is to achieve privacy for DNS communication between resolver and authoritative server by helping them to migrate to an encrypted transport.

- This has been discussed for years with little progress.

There are two proposals to improve this poor state:


- RFC 9539 (now, “blind probing”)
- DELEG (in the future)

It seems reasonable to assume that DELEG will solve this in the future.

- Question #1: **When** will DELEG be widely deployed?
- Question #2: **How many trillions** of DNS queries need to be **sent in cleartext** while we wait?

# Problem Statement

The Chicken



The goal is to achieve privacy for DNS communication between **resolver** and authoritative server by helping them to migrate to an encrypted transport.

- This has been discussed for years with little progress.

There are two proposals to improve this poor state:

- RFC 9539 (now, “blind probing”)
- DELEG (in the future)

It seems reasonable to assume that DELEG will solve this in the future.

- Question #1: **When** will DELEG be widely deployed?
- Question #2: **How many trillions** of DNS queries need to be **sent in cleartext** while we wait?

# Problem Statement

The goal is to achieve privacy for DNS communication between **resolver** and **authoritative server** by helping them to migrate to an encrypted transport.

The Chicken

- This has been discussed for years with little progress.

The Egg

There are two proposals to improve this poor state:

- RFC 9539 (now, “blind probing”)
- DELEG (in the future)

It seems reasonable to assume that DELEG will solve this in the future.

- Question #1: **When** will DELEG be widely deployed?
- Question #2: **How many trillions** of DNS queries need to be **sent in cleartext** while we wait?

## Current State

RFC 9539 proposes that the resolver opportunistically ( “**blindly**” ) tries alternative transports to the authoritative server “**to see if it works**”.

- One problem is that this is not used much yet.

DELEG is a larger protocol change under discussion.

- DELEG aims to solve several problems at once, including this one.
- As DELEG requires changes at the delegation points it will take time to deploy.

This raises the question of whether it is possible to find a silver bullet that:

- ...improves the conditions for transport probing
- ...within the framework of the current DNS protocol
- ...where most of the solution can be reused by resolvers that implement DELEG-based transport signaling in the future

# Our Proposal

Let's do the following:

- Reuse the format for DELEG-based transport signal, i.e. a key/value pair inside an SVCB record (or SVCB-derivative).
- Add the SVCB record to the Additional section in responses from authoritative servers when possible: `ns.provider.example. 10800 IN SVCB 1 . alpn="doq,dot,do53"`
  - ▶ We call this an `OTS Hint`, an “Opportunistic Transport Signal”.

# Our Proposal

Let's do the following:

"My preference is first  
DNS-over-QUIC, then DNS-over-TLS  
and, finally, standard UDP/TCP"

- Reuse the format for DELEG-based transport signal, i.e. a key/value pair inside an SVCB record (or SVCB-derivative).
- Add the SVCB record to the Additional section in responses from authoritative servers when possible: `ns.provider.example. 10800 IN SVCB 1 . alpn="doq,dot,do53"`
  - ▶ We call this an **OTS Hint**, an "Opportunistic Transport Signal".

# Our Proposal

Let's do the following:

"My preference is first  
DNS-over-QUIC, then DNS-over-TLS  
and, finally, standard UDP/TCP"

- Reuse the format for DELEG-based transport signal, i.e. a key/value pair inside an SVCB record (or SVCB-derivative).
- Add the SVCB record to the Additional section in responses from authoritative servers when possible: `ns.provider.example. 10800 IN SVCB 1 . alpn="doq,dot,do53"`
  - ▶ We call this an **OTS Hint**, an "Opportunistic Transport Signal".
- The resolver chooses whether to act on the received signal or not.
  - ▶ But if it does, it may "upgrade" the next message exchange with the same authoritative server to use a transport according to the server's signaled preference.



## Example: an Authoritative DNS Response

```
godev:/src# dig @ns.provider.example p.axfr.net soa

;; opcode:  QUERY, status:  NOERROR, id:  41917
;; flags:  qr aa rd; QUERY:  1, ANSWER:  1, AUTHORITY:  1, ADDITIONAL:  3

;; QUESTION SECTION:
;p.axfr.net.  IN SOA

;; ANSWER SECTION:
p.axfr.net.      7200 IN SOA when.pigs.can.fly. hostmaster.johani.org. ...

;; AUTHORITY SECTION:
p.axfr.net.      300 IN NS ns.provider.example.

;; ADDITIONAL SECTION:
ns.provider.example.  300 IN A 77.72.230.63
ns.provider.example.  300 IN AAAA 2a01:3f0:1:2::63
ns.provider.example. 10800 IN SVCB 1 . alpn="doq,dot,do53"
ns.provider.example. 18000 IN RRSIG SVCB 13 3 ... ZdweaDo...
```

Even if the zone is unsigned, the operator's authoritative server may well be located in a signed zone, which makes validation of this SVCB possible.

## Example: an Authoritative DNS Response

Query this name server

```
godev:/src# dig @ns.provider.example p.axfr.net soa
;; opcode:  QUERY, status:  NOERROR, id:  41917
;; flags:  qr aa rd; QUERY:  1, ANSWER:  1, AUTHORITY:  1, ADDITIONAL:  3

;; QUESTION SECTION:
;p.axfr.net.  IN SOA

;; ANSWER SECTION:
p.axfr.net.  7200 IN SOA when.pigs.can.fly. hostmaster.johani.org. ...

;; AUTHORITY SECTION:
p.axfr.net.  300 IN NS ns.provider.example.

;; ADDITIONAL SECTION:
ns.provider.example.  300 IN A 77.72.230.63
ns.provider.example.  300 IN AAAA 2a01:3f0:1:2::63
ns.provider.example. 10800 IN SVCB 1 . alpn="doq,dot,do53"
ns.provider.example. 18000 IN RRSIG SVCB 13 3 ... ZdweaDo...
```

Even if the zone is unsigned, the operator's authoritative server may well be located in a signed zone, which makes validation of this SVCB possible.

## Example: an Authoritative DNS Response

Query this name server

```
godev:/src# dig @ns.provider.example p.axfr.net soa
;; opcode:  QUERY, status:  NOERROR, id:  41917
;; flags:  qr aa rd; QUERY:  1, ANSWER:  1, AUTHORITY:  1, ADDITIONAL:  3

;; QUESTION SECTION:
;p.axfr.net.  IN SOA

;; ANSWER SECTION:
p.axfr.net.      7200 IN SOA when.pigs.can.fly. hostmaster.johani.org. ...

;; AUTHORITY SECTION:
p.axfr.net.      300 IN NS ns.provider.example.

;; ADDITIONAL SECTION:
ns.provider.example.  300 IN A 77.72.230.63
ns.provider.example.  300 IN AAAA 2a01:3f0:1:2::63
ns.provider.example. 10800 IN SVCB 1 . alpn="doq,dot,do53"
ns.provider.example. 18000 IN RRSIG SVCB 13 3 ... ZdweaDo...
```

Name server  
is in  
NS RRset

Even if the zone is unsigned, the operator's authoritative server may well be located in a signed zone, which makes validation of this SVCB possible.

## Example: an Authoritative DNS Response

Query this name server

```
godev:/src# dig @ns.provider.example p.axfr.net soa
;; opcode:  QUERY, status:  NOERROR, id:  41917
;; flags:  qr aa rd; QUERY:  1, ANSWER:  1, AUTHORITY:  1, ADDITIONAL:  3

;; QUESTION SECTION:
;p.axfr.net.  IN SOA

;; ANSWER SECTION:
p.axfr.net.  7200 IN SOA when.pigs.can.fly. hostmaster.johani.org. ...

;; AUTHORITY SECTION:
p.axfr.net.  300 IN NS ns.provider.example.

;; ADDITIONAL SECTION:
ns.provider.example.  300 IN A 77.72.230.63
ns.provider.example.  300 IN AAAA 2a01:3f0:1:2::63
ns.provider.example. 10800 IN SVCB 1 . alpn="doq,dot,do53"
ns.provider.example. 18000 IN RRSIG SVCB 13 3 ... ZdweaDo...
```

Name server  
is in  
NS RRset

OTS hint

Even if the zone is unsigned, the operator's authoritative server may well be located in a signed zone, which makes validation of this SVCB possible.

# Disadvantages of the Proposal: No Guarantees

- It is opportunistic. That is, there is no guarantee that the transport signal exists, and even if it does, it can theoretically be deleted by an "on-path attacker".
  - ▶ Statistically, this is rarely a problem.
  - ▶ **The goal here is to improve privacy for Internet users at large**, not to guarantee encrypted transport for any specific domain name.
- It is not secure (i.e. DELEG will be an improvement in the future).
  - ▶ But it is as secure as what we have today.
  - ▶ **The goal is privacy**, and we get privacy with this mechanism.
- The very first query to an authoritative server will use "traditional" DNS transport.
  - ▶ Given the consolidation of DNS operators, this is not much of a problem. The first query to an authoritative server is usually followed by a very large number of subsequent queries.

## Advantages of the Proposal: Simplicity

- It **does not change the relationship between registrant/registrar/registry** or the DNS ecosystem in any way.
- It **does not change anything in the parent zone.**
  - ▶ Usually not in the child zone either.
- It is trivial to implement on the authoritative server side.
- Less trivial to implement on the resolver side.
  - ▶ But **90% of resolver complexity is the same as required to support RFC 9539 and/or DELEG** (so it must be implemented anyway).
- The only change to DNS records is the addition of the **SVCB record in the zone where the name server is located**
  - ▶ That **SVCB** record can be synthesized automatically if desired.

# Advantages Compared to “Waiting For DELEG”

- Privacy **as soon as possible** is very important.
  - ▶ Because the proposal is so simple, there is a realistic chance to get encrypted transport to and from global DNS operators **much sooner** than the 10+ years likely needed for wide spread deployment of DELEG.
- Good to get **early experience with DNS transport signaling**.
  - ▶ The signaling mechanism is intended to be the same as for DELEG.
  - ▶ Should it turn out that this is not optimal for some reason, there is a chance to fix this in the work with DELEG.

# Advantages Compared to “Only Doing RFC 9539”

Compared to RFC 9539 “blind probing”, the proposal has two important advantages, both of which are about the resolver becoming “less blind”:

- There is a difference between **supporting a transport**, e.g. DoQ, and **advertising this support**.
  - ▶ With only RFC 9539, a resolver will note when a name server enables DoQ for testing. But there is no way to distinguish between “test” and “production”.
- Without a transport signal, there is no way to **upgrade the reliability** of the signal.
  - ▶ If there is a signal (the SVCB record with the `OTS Hint`), the resolver can choose to explicitly query for the signal. Then the signal becomes as reliable as if DELEG had been used, i.e. no longer “opportunistic”.



# Open Questions

Of course, there are some open questions:

- There needs to be a mechanism for the resolver to either signal
  - ▶ "I **DO NOT** want any OTS Hint" (OPT-OUT)
  - ▶ "I **DO** want an OTS Hint if possible" (OPT-IN)
  - ▶ Right now we support both (via a new EDNS(0) option), but hopefully either should be enough.
- What should be the resolver caching behaviour for the OTS Hint?
- Should the resolver proactively improve the reliability of the signal by explicitly querying for this SVCB?

We are working on these questions, and it does seem that this proposal is generating quite a bit of interest.

# Implementation Status

There is a working implementation of this proposal for opportunistic DNS transport signaling:

- The authoritative side is largely complete.
  - ▶ It was mostly trivial to add to the existing experimental authoritative server we use.
- The resolver side “works”, but will need more work to be robust.
  - ▶ Our proof-of-concept recursive server is. . . obviously very limited.
  - ▶ We would love to see support implemented in a “real” resolver.
- The goal is to have a fully working implementation of both the authoritative and recursive server later this year.
- The implementation is written in Go and fully open source (using the same codebase, `tdns`, as we use for our other DNS projects).

# Implementation Status

In addition to the resolver side being a bit more complex, we also had to implement a recursive server since we didn't have one.

There is a working implementation of this proposal for opportunistic DNS transport signaling:

- The authoritative side is largely complete.
  - ▶ It was mostly trivial to add to the existing experimental authoritative server we use.
- The resolver side “works”, but will need more work to be robust.
  - ▶ Our proof-of-concept recursive server is... obviously very limited.
  - ▶ We would love to see support implemented in a “real” resolver.
- The goal is to have a fully working implementation of both the authoritative and recursive server later this year.
- The implementation is written in Go and fully open source (using the same codebase, `tdns`, as we use for our other DNS projects).

# Implementation Status

In addition to the resolver side being a bit more complex, we also had to implement a recursive server since we didn't have one.

There is a working implementation of this proposal for opportunistic DNS transport signaling:

- The authoritative side is largely complete.
  - ▶ It was mostly trivial to add to the existing experimental authoritative server we use.
- The resolver side “works”, but will need more work to be robust.
  - ▶ Our proof-of-concept recursive server is... obviously very limited.
  - ▶ We would love to see support implemented in a “real” resolver.
- The goal is to have a fully working implementation of both the authoritative and recursive server later this year.
- The implementation is written in Go and fully open source (using the same codebase, `tdns`, as we use for our other DNS projects).

Nudge, nudge,  
wink, wink.

# Summary

- We propose a simple mechanism for opportunistic DNS transport signaling.
  - ▶ The goal is to improve privacy for Internet users at large **as soon as possible**.
- It requires no changes to the DNS protocol, it is purely operational.
- It has advantages compared to only doing RFC 9539 and/or waiting for DELEG.
- It is as secure as what we have today.
- It is simple to implement on the authoritative server side.
- It is more complex to implement on the resolver side, but that is inherent to the problem and code may be re-used in the future.

Questions?

## Thank You & Contact Information

```
erik.bergstrom }
leon.fernandez  } @internetstiftelsen.se
johan.stenstam  }
```

**Source code**      <https://github.com/johanix/tdns>