Announcing CDS/CSYNC Scanner Existence

draft-berra-announce-scanner-00

Erik Bergström Leon Fernandez Johan Stenstam

Swedish Internet Foundation

July 25, 2025

Is It Better To Be Late Than Never?

In the draft for generalized notifications (that is in the RFC Editors queue) there are specifications for how to announce the existence and location of a **notification receiver**:

```
_dsync.parent.example. IN DSYNC CDS NOTIFY {port} {target}
```

- i.e. a service for receiving notifications from children that there may have been changes that should trigger a CDS or CSYNC scanner to take an immediate look at the child
- eg. the CDS may have changed

Is It Better To Be Late Than Never?

In the draft for generalized notifications (that is in the RFC Editors queue) there are specifications for how to announce the existence and location of a **notification receiver**:

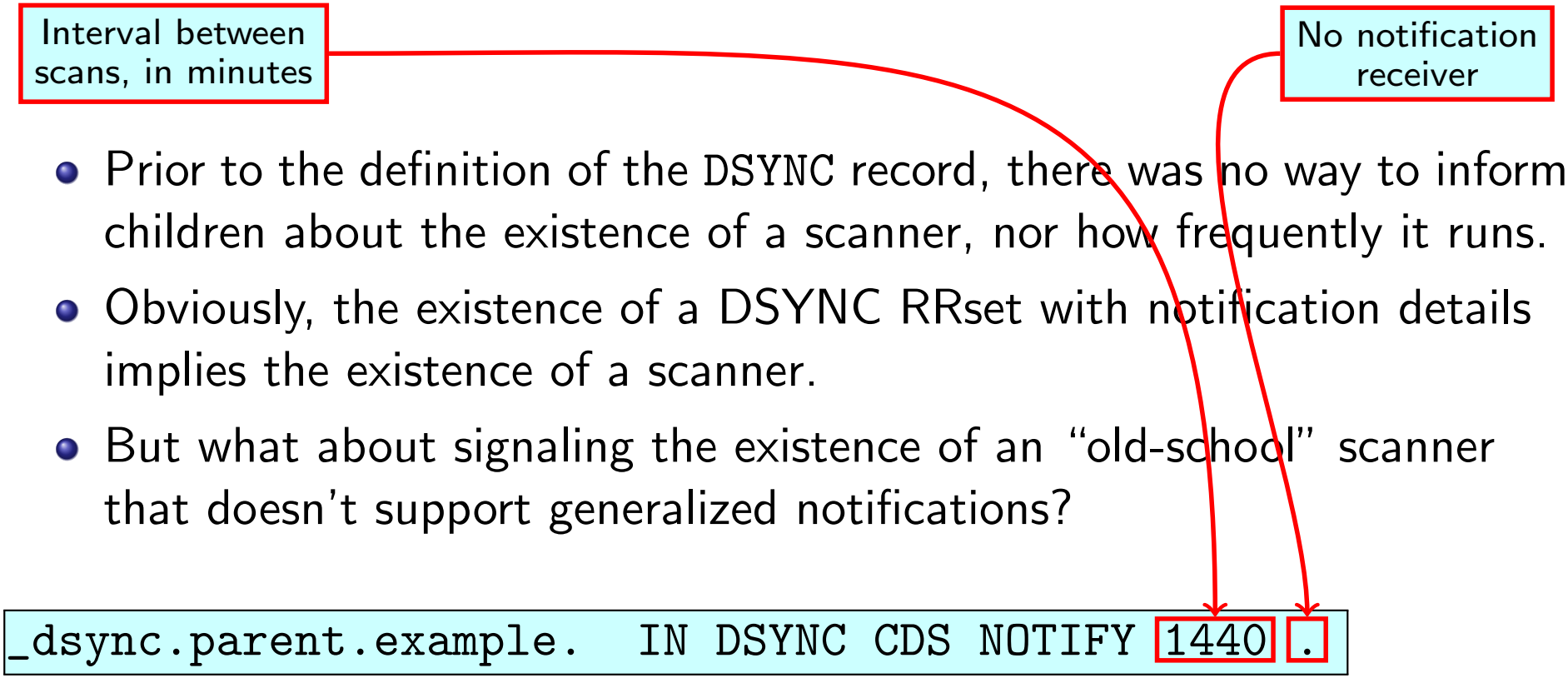
```
_dsync.parent.example. IN DSYNC CDS NOTIFY {port} {target}
```

- i.e. a service for receiving notifications from children that there may have been changes that should trigger a CDS or CSYNC scanner to take an immediate look at the child
- eg. the CDS may have changed

I think we may have missed (or almost missed) an opportunity to provide some additional useful information to the child (or child operator).

The question is whether we should and can fix this somewhat quickly.

Missed Opportunity #1: Signaling Scanner **Frequency**



- Prior to the definition of the `DSYNC` record, there was no way to inform children about the existence of a scanner, nor how frequently it runs.
- Obviously, the existence of a DSYNC RRset with notification details implies the existence of a scanner.
- But what about signaling the existence of an "old-school" scanner that doesn't support generalized notifications?

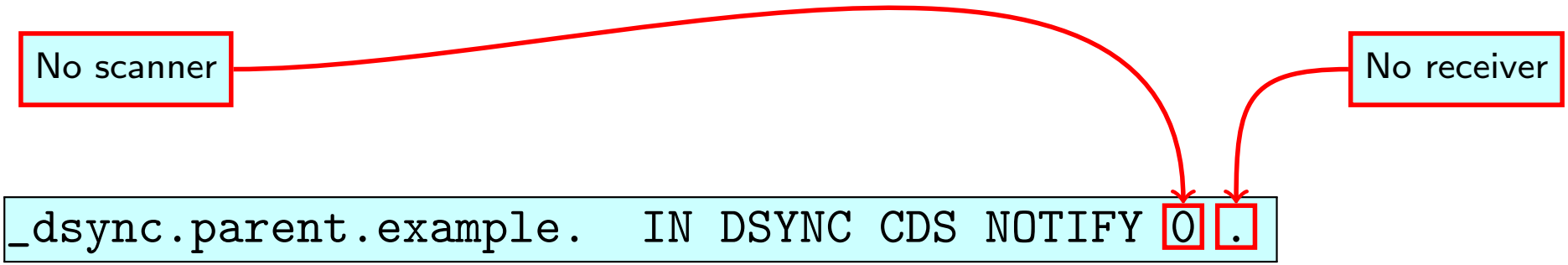
We propose that when `target="."` then the `DSYNC` record does not describe a receiver but the scanner itself.

- In that case the `port` field is reused to signal the scanner frequency

Missed Opportunity #2: Signaling **Absence** of Scanner

What about the case where there is no scanner at all?

- It is possible to overload extra semantics on the `port` field to signal the absence of any scanner.



We propose that when `port="0"` (and `target="."`) then the `DSYNC` record signals the absence of both the notification receiver **and** a scanner.

There are a Couple of Caveats

- The signal is primarily intended for use by the child operator.
  - **"Aha, the reason my `CDS` has not been picked up yet is because the scanner only runs once per day."**
  - **"Bummer, there is no `CSYNC` scanner at all. I want a better parent."**
- The signal provided this way **must not modify the behaviour of the child** (other than that if `target="."` then there's no point in trying to send notifications).
- If the parent is a registry, with registrars, then it would be counterproductive to signal the absence of a scanner unless it is certain that no registrar is scanning.
- Overloading semantics of previously defined fields is not exactly elegant, but it is the only viable option at this time.
  - I do **not** suggest that we open up the generalized notifications draft to redesign the `DSYNC` record.

Opportunistic DNS Transport Signaling

A Proposal for Enhanced Privacy of Authoritative DNS Communication

`draft-johani-dnsop-transport-signaling`

Erik Bergström Leon Fernandez Johan Stenstam

Swedish Internet Foundation

July 25, 2025

Problem Statement

We have talked about how to add privacy to the DNS communication between resolver and authoritative server for years without much real progress (AFAIK).

There are two approaches to improve this quite unsatisfactory state:

- `RFC 9539` (now)
- `DELEG` (in the future)

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

- The issue is when `DELEG` will be widely deployed, i.e. how long will we have to wait.
- And how many gazillions of clear-text DNS packets will be sent in the meantime.

`RFC9539` is an attempt at doing something within the current DNS protocol, in this case by doing "blind probing" (opportunistically just try alternative transports to the auth server to see if it works).

- The issue is that it isn't being used much by resolvers yet (but we hope that will change).

Problem Statement, cont'd

Our position is that we would like to improve the privacy of Internet users as soon as possible, via encryption of the DNS traffic to and from authoritative nameservers (i.e. the same view as expressed in RFC9539).

This prompts the question whether it is possible to:

- Change something so that we improve the success rate of probing.
- While staying within the current DNS protocol.
- Where most of the solution may be re-used by resolvers implementing `DELEG`-based transport signaling in the future.

Essentially the goal is to **change the incentive** for resolvers by improving on the RFC9539 "blind probing" so that it has a higher likelihood of success.

Proposal

Let's do the following:

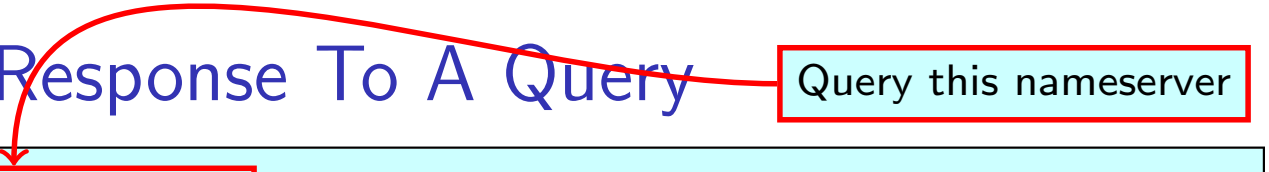
- Re-use the format of the `DELEG`-based transport signal (i.e. a key/value pair inside an `SVCB` or `SVCB`-derivative.
- Add a new `SVCB` record to the Additional Section in responses from authoritative servers when possible.
  - This is referred to as the `OTS Hint`.
- The resolver may or may not choose to act on the received transport signal, but if it does act on it it may "upgrade" the next communication with the same server to use a transport according to that servers signaled preference.

Example: an Auth Response To A Query

```
godev:/src# dog @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...
```

Note that while the zone may be unsigned, the provider auth server may well be located in a signed zone, making validation of the SVCB possible (but we cannot depend on this).

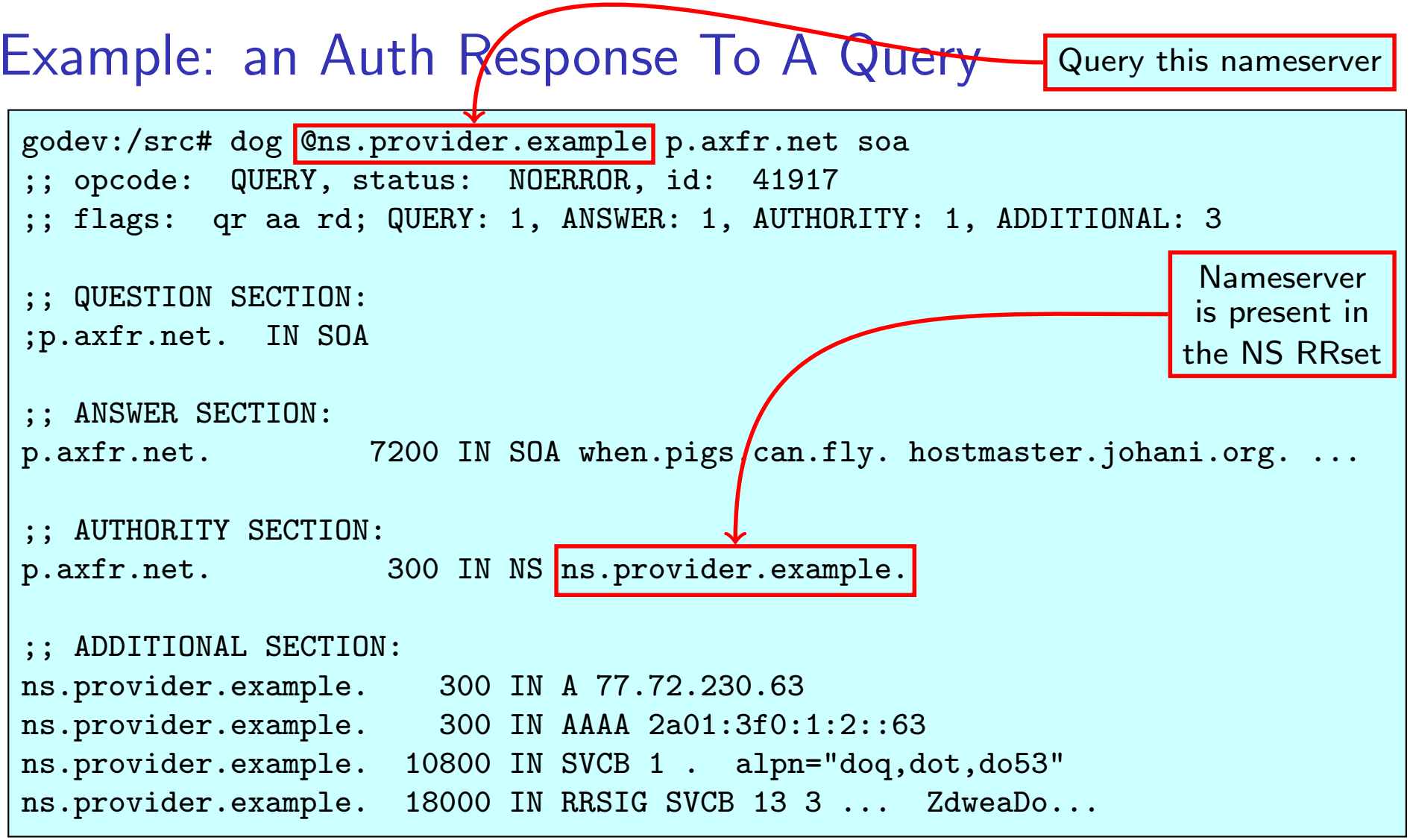
Example: an Auth Response To A Query




```
godev:/src# dog @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...
```

Note that while the zone may be unsigned, the provider auth server may well be located in a signed zone, making validation of the SVCB possible (but we cannot depend on this).

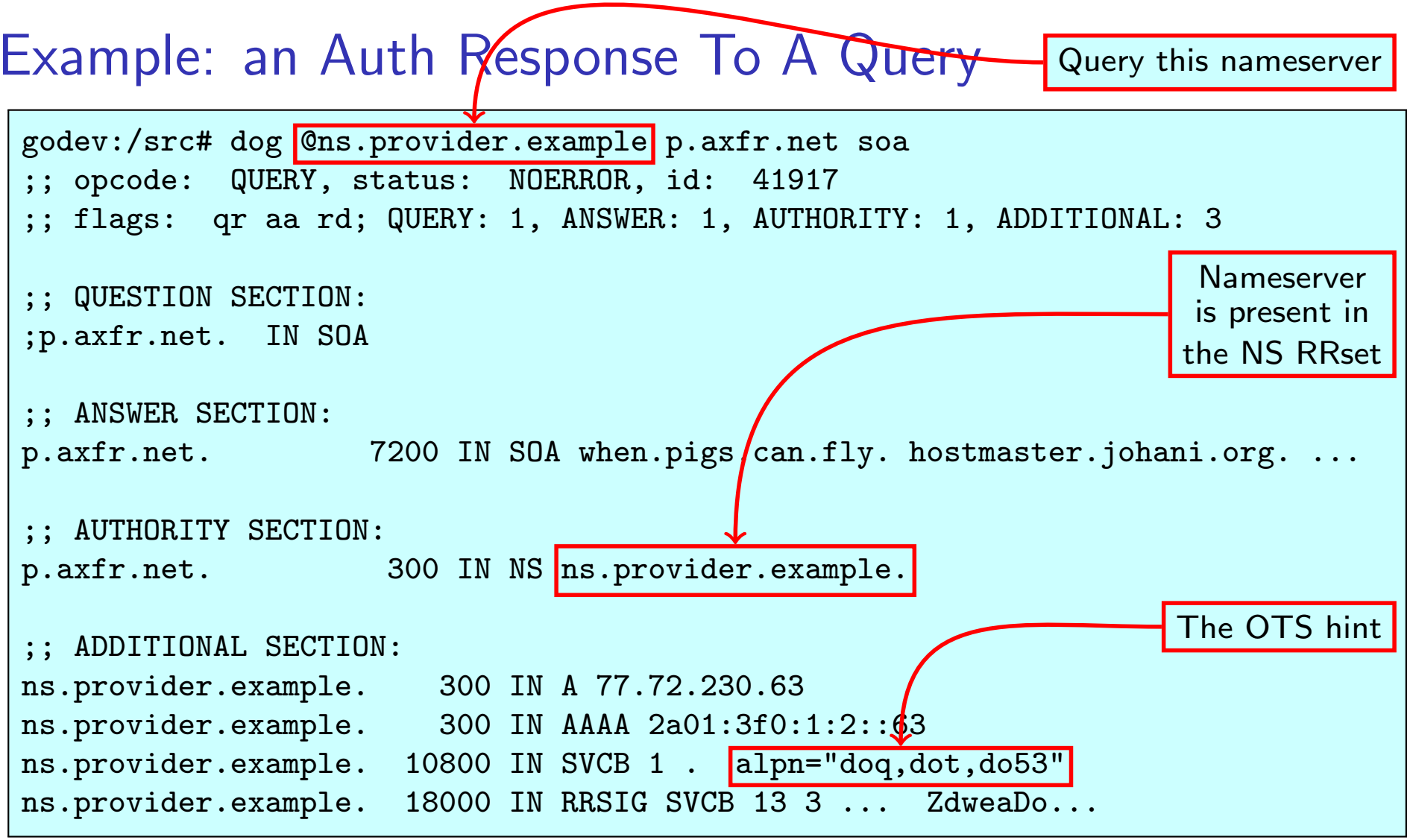
Example: an Auth Response To A Query




```
godev:/src# dog @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...
```

Note that while the zone may be unsigned, the provider auth server may well be located in a signed zone, making validation of the SVCB possible (but we cannot depend on this).

Example: an Auth Response To A Query




```
godev:/src# dog @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...
```

Note that while the zone may be unsigned, the provider auth server may well be located in a signed zone, making validation of the SVCB possible (but we cannot depend on this).

Advantages of the Proposal: Simplicity

- It **does not touch the registrant/registrar/registry relationship** or eco system in any way.
- It **does not touch the parent zone** and usually not even the child zone.
- It is trivial to implement in the authoritative server end.
- Less trivial to implement in the resolver end, but **90% of the resolver complexity is mostly identical to the DELEG-support that will need to be implemented regardless.**
- The only change to DNS records is the addition of the **SVCB record in the zone where the nameserver is located** (and that record may be synthesized automatically if wanted).

Advantages, cont'd

In addition to the privacy goals, being able to **gain experience with auth DNS transport signaling prior to the deployment of** `DELEG` is another plus.

Finally, when comparing RFC 9539 "blind probing" with this proposal, the proposal has two key advantages:

- There is a difference between having the ability to support, eg. DoQ, and announcing that ability.
- Without a transport signal, there is no way of upgrading the reliability of the signal. If there is a signal, then the resolver can choose to query for the signal explicitly.

This means that the resolver can choose to ignore the signal and continue using the current "blind probing" mechanism.

Disadvantages of the Proposal: No Guarantees

- It is opportunistic. I.e. there is no guarantee that the transport signal will be present, and even if it is, it could be stripped away by an on-path attacker.
- It is not secure (i.e. `DELEG` will be an improvement in the future).
  - However, it is just as secure as what we have today. The goal is **privacy**, and we do get privacy with this mechanism.

Open Questions

There are open questions:

- There needs to be a mechanism for the resolver to either signal "I do not want any OTS Hint" (`OPT_OUT`) or "Please provide an OTS Hint if possible" (`OPT_IN`).
  - Right now we support both (via an EDNS(0) option), but either one should be enough.
- What is the right caching behavior for the OTS Hint?
- Should the resolver improve the reliability of the signal by querying for the SVCB explicitly?

Implementation Status

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

- The authoritative side is essentially complete.
  - Almost trivial to add to the existing experimental auth server we use.
- The resolver side is "working", but will need more work to be robust.
  - Apart from the resolver side being a bit more complex, we also needed to implement a recursive server as we didn't have that. The recursive server we built as a proof-of-concept is... very limited.
- The internet-draft will be updated shortly.
- Aiming to have a fully functioning implementation of both auth side and resolver side later this year.
- The implementation is written in Go and completely open source.

Questions?

Thanks & Contact Information

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

Code https://github.com/johanix/tdns