Locating the Target

for NOTIFY(CDS), NOTIFY(CSYNC), DNS UPDATE, etc

Johan Stenstam and Peter Thomassen

November 7, 2023

Problem Statement

Both draft-ietf-dnsop-generalized-notify and draft-johani-dnsop-delegation-mgmt-via-ddns rely on the availability of information in the parent zone for the child to know where to send the information.

- a `NOTIFY` or an `UPDATE` in these cases
- there may be more cases coming

Essentially neither draft proposes anything new, **except for how to locate the target** of the `NOTIFY`/`UPDATE`/etc.

- `NOTIFY(CDS)`, `NOTIFY(CSYNC`, etc, are already allowed by the protocol.
- Using `UPDATE` from the child to the parent to update delegation information is both allowed by the protocol and implemented since many years.

Problem Statement, cont'd

Therefore, our focus here is on how to design this convention for how to locate the target.

There are several alternatives for how this parent-side information should be presented. Each with its pros and cons.

- Obviously, static configuration (typically in the child primary nameserver) will always be an alternative
- But the discussion here and now is about the dynamic alternatives

What should the child lookup to locate the target?

The child needs to know the **mechanism** to use for notifications and where to send the message (the "**target**").

- The mechanism is **NOTIFY**, **UPDATE** or something else (perhaps "**API**").

Issue #1: What RR type should be looked up. Most likely either a new RR type (eg. `DSYNC` or `NOTIFY`) or an `SVCB` record.

Issue #2: What `qname` should be looked up?

What needs to be decided is a **convention**, i.e. a social contract. After this has been implemented in software and deployed in zones it would be painful to change.

- We should of course try hard to get it right.

What RR type to lookup to locate the notification target

Alternative #1: Define a new RR type.

- **Pro:** Possible to define exactly what is needed.
- **Pro:** A unique RR type will not collide with "other uses" in the same RRset (think `DNSKEY` vs. `KEY`).
- **Con:** Initially more difficult to debug, as tools will not know the new type.

Alternative #2: Use the existing type `SVCB` with an appropriate profile.

- **Pro:** `SVCB` is there and would work.
- **Con:** `SVCB` is still an Internet-Draft, not an RFC.
- **Con:** Risk of ending up with other uses of `SVCB` in the same RRset.

Our view: Long term is more important than short term. Hence new RR type is a better choice.

What qname to lookup to locate the notification target

Alternative #1: do a direct lookup of qname=parent. to locate target.

```
parent.  IN SOA ...
...
parent.  IN DSYNC CDS   1 5301 notifications.parent.
parent.  IN DSYNC CSYNC 1 5302 notifications.parent.
```

"scheme=1" indicate mechanism=NOTIFY

Target

Port

- "scheme=1" is interpreted as "send a a NOTIFY for the right RRtype to the target and port specified.

```
parent.  IN SOA ...
...
parent.  IN DSYNC ANY   2 5399 ddns-receiver.parent.
```

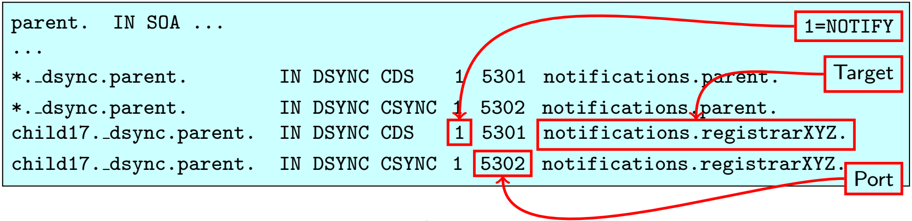
"scheme=2" indicate mechanism=UPDATE

- **Pro:** Simple to understand and implement.
- **Con:** Doesn't provide any escape for per-registrar targets.

Our view: Not flexible enough for all use cases.

What qname to lookup to locate the notification target

Alternative #2: do a direct lookup of `child.something.parent.` Will likely trigger a wild card expansion in most cases.



```
parent.  IN SOA ...
...
*._dsync.parent.        IN DSYNC CDS    1 5301 notifications.parent.
*._dsync.parent.        IN DSYNC CSYNC  1 5302 notifications.parent.
child17._dsync.parent.  IN DSYNC CDS    1 5301 notifications.registrarXYZ.
child17._dsync.parent.  IN DSYNC CSYNC  1 5302 notifications.registrarXYZ.
```

1=NOTIFY

Target

Port

- **Pro:** Allows separate targets for child zones that have a registrar that does scanning.
- **Con:** Potentially millions of additional records to publish (although they can be in a separate zone or generated dynamically).
- **Con:** Overly complex for the non-registry parent cases.
- **Con:** Name space pollution.

Our view: More complex than using parent apex (Alt. #1).

What `qname` to lookup to locate the notification target

child17 has a specific target

NOTIFY target for child17

Alternative #3: start with **Alt. #2**. Fall back to **Alt. #1** if needed.

1=NOTIFY

```
*._dsync.parent1.        IN DSYNC CDS    1  5301 notifications.parent1.
*._dsync.parent1.        IN DSYNC CSYNC  1  4553 notifications.parent1.
child17._dsync.parent1.  IN DSYNC CDS    1  5301 notifications.registrarXYZ.
child17._dsync.parent1.  IN DSYNC CSYNC  1  4553 notifications.registrarXYZ.
```

2=UPDATE

```
parent2.                 IN DSYNC CDS    2 5301 ddns-receiver.parent2.
parent2.                 IN DSYNC CSYNC  2 5302 ddns-receiver.parent2.
```

- If there is no answer (i.e. no RR at `child._dsync.parent2.`) then fall back to **Alternative #1** and do a lookup in the parent apex.
- **Pro:** Most flexible.
- **Con:** Will sometimes cause two DNS queries.

Our view: Best alternative so far.

Summary

In the end it is sometimes more important to make a choice than exactly which choice is made.

- However, it does become unnecessary unpleasant for implementors if the choice isn't flexible enough for the use cases. Revisiting a previously made choice can be painful.

Our view:

- The choice of new RR type vs `SVCB` is rather simple. Both will work (although with different pros and cons). Let's just pick one.
  - We suggest allocating a new RR type.
- The choice of what `qname` the child should query for is more delicate. We do not want to get that wrong.
  - We suggest to play it safe and go for #3 (try most general with fallback if needed) as the most flexible alternative.