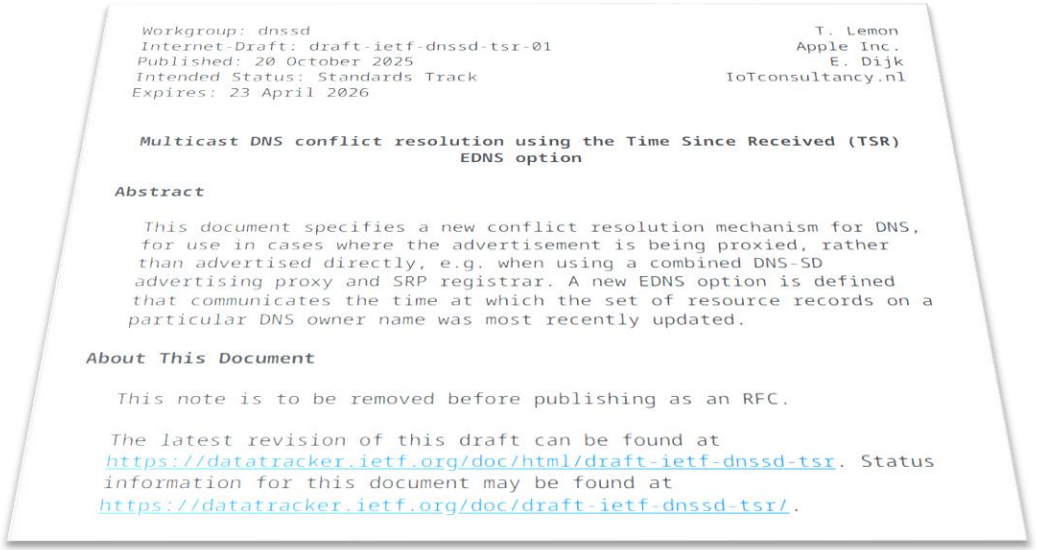
Multicast DNS conflict resolution using the Time Since Received (TSR) EDNS option

draft-ietf-dnssd-tsr-01



Workgroup: dnssd
Internet-Draft: draft-ietf-dnssd-tsr-01
Published: 20 October 2025
Intended Status: Standards Track
Expires: 23 April 2026

T. Lemon
Apple Inc.
E. Dijk
IoTconsultancy.nl

Multicast DNS conflict resolution using the Time Since Received (TSR) EDNS option

Abstract

This document specifies a new conflict resolution mechanism for DNS, for use in cases where the advertisement is being proxied, rather than advertised directly, e.g. when using a combined DNS-SD advertising proxy and SRP registrar. A new EDNS option is defined that communicates the time at which the set of resource records on a particular DNS owner name was most recently updated.

About This Document

This note is to be removed before publishing as an RFC.

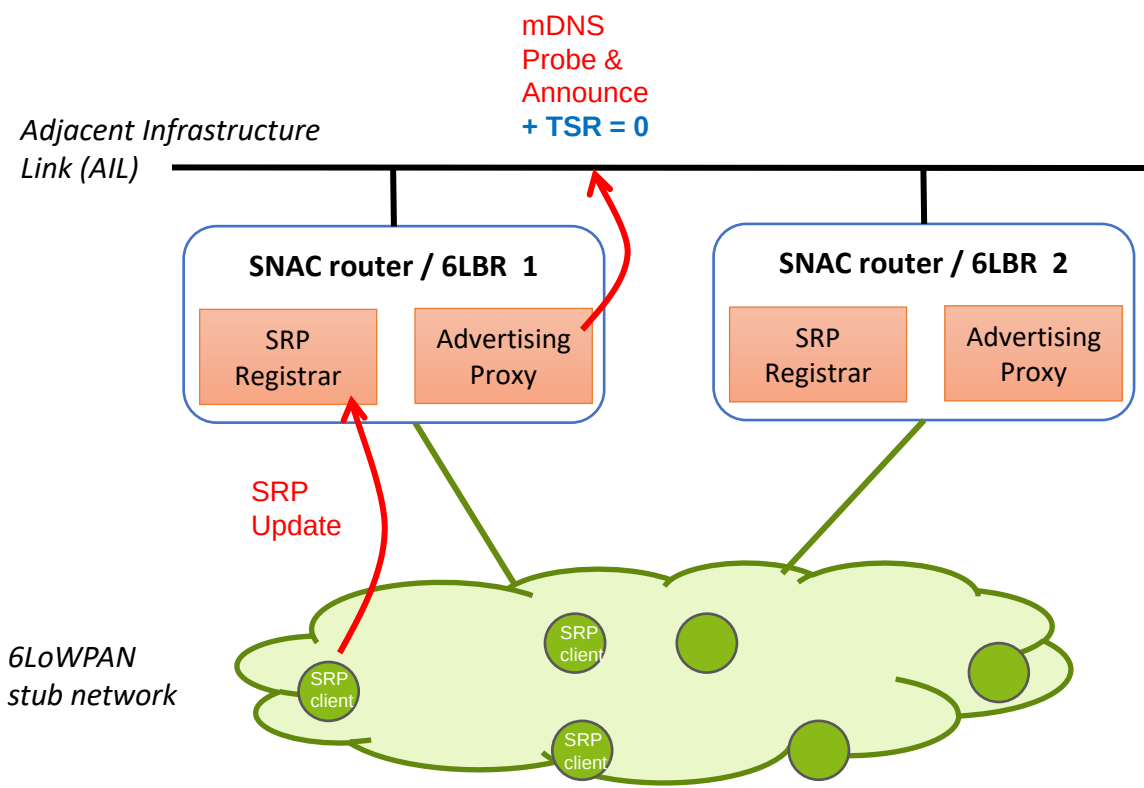
The latest revision of this draft can be found at https://datatracker.ietf.org/doc/html/draft-ietf-dnssd-tsr. Status information for this document may be found at https://datatracker.ietf.org/doc/draft-ietf-dnssd-tsr/.

DNSSD WG

IETF 124, Montreal, November 2025

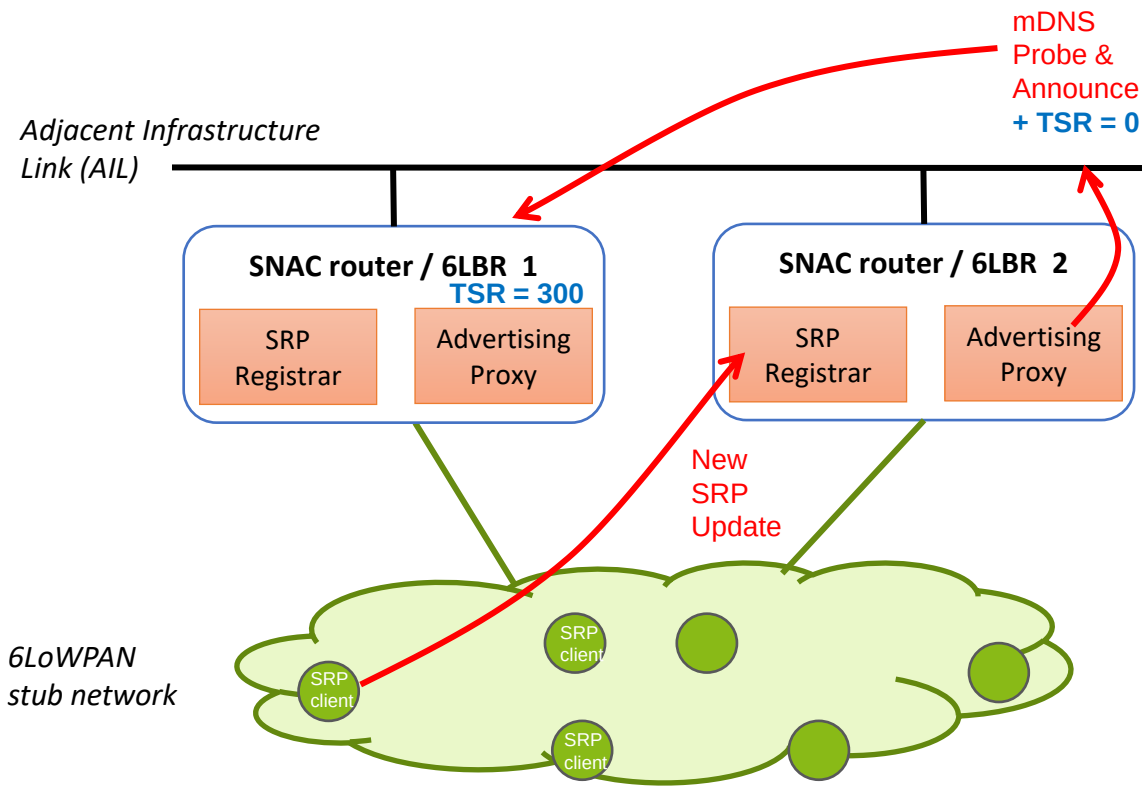
Ted Lemon – Apple Inc.
Esko Dijk – IoTconsultancy.nl **(presenting)**
esko.dijk@iotconsultancy.nl

Recap: Using Time Since Received (TSR) in Advertising Proxies



- Service registered via SRP Registrar
- Advertising Proxy ‘translates’ it to mDNS
- mDNS client can query for services of stub nodes

Recap: Using Time Since Received (TSR) in Advertising Proxies



- SRP client registers to **other** SRP Registrar
- SNAC router 2 can still advertise the service, since more recent
- No mDNS conflict thanks to TSR
- Stale registration at SNAC router 1 gets removed.

Version -01: Updates since IETF 123

- Added terms & definitions section
- Consistent use of same terms for same things
- References cleanup

Open Issues – To Discuss

- **Do we need to probe when we see existing data that has the same timestamp? #6**

> If such data exists and the key checksums match, there are three possibilities based on the known TSR time and the proposed TSR time:
>
> **Known time is more recent** In this case, the registrar **MUST** treat the new registration as stale, and return an indication to the registrant that its registration is stale. This indication must be different than the conflict indication.
>
> **Both times are the same** In this case, the new record is added to the local registration database and put in the probing state.
>
> **Proposed time is more recent** In this case, all cached data on the name is discarded. The registrant for any existing locally-registered data is notified that the data they have registered is stale, and the stale data is removed from the local registration database. The new data is added and put in the probing state, and the TSR data is updated with the proposed TSR data.

Open Issues – To Discuss

- **Do we need to probe when we see existing data that has the same timestamp?** #6

- If we can avoid mDNS traffic, it's always better.
- The registered data is already in the cache of the (local) mDNS registrar – same timestamp means same data.
- Do we need to announce even?

- **Not announcing implies**: an mDNS registrant cannot submit multiple updates with different data within the same 1-second period.

Open Issues – To Discuss

- **Do we need to probe when we see existing data that has the same timestamp? #6**

- The 1-second period (prev slide) seems to fit with mDNS RFC 6762:

> To protect the network against excessive packet flooding due to software bugs or malicious attack, a Multicast DNS responder MUST NOT (except in the one special case of answering probe queries) multicast a record on a given interface until at least one second has elapsed since the last time that record was multicast on that particular interface.

Open Issues – To Discuss

- **Multiple proxies - clarify handling of "older data transmitted first" case? #16**

- This assumes there could be N (>1) mDNS registrars with registered data for the same name.
  - For example, if a probe + announce was missed (msg drop? cable issue?)

- Should we clarify the outcome of the requirements in Section 3.3 / 3.4 ?
  - If an mDNS query is sent, a stale answer (higher TSR) may be sent,
  - Followed by a more fresh answer (lower TSR), etc.
  - The reverse is far less likely, because stale data is cleared by fresher answers.

Open Issues – To Discuss

- **Multiple proxies - clarify handling of "older data transmitted first" case?** #16
- What if the query sender already started to use the stale data?
  - Do we expect TSR-aware mDNS clients to wait a bit longer, to get fresh data?
- **Or**: do we expect the mDNS registrar with fresher data to quickly respond, if it sees stale data being advertised in response to a query that the registrar was planning to answer already?
- **Or**: should the mDNS registrar delay the answer more, the higher the TSR value for the data is?

Open Issues – To Discuss

- **Multiple proxies - clarify handling of "older data transmitted first" case? #16**
- Should an mDNS registrar suppress its answer if another registrar already answered with exact same data?
  - Even if the previous answer was more stale according to TSR value?
  - Is the mDNS registrar even able to compare received data to its local data always?

Open Issues – To Discuss

- **Multiple proxies - mixed use of TSR / non-TSR answers** #17
- mDNS registrar that uses TSR may see non-TSR mDNS advertisements, for a name it has registered.
- Should we add a section 3.x “Processing messages without TSR” ?
  - We have Section 3.1 but that’s only about probing a **new** registration.
- What should the mDNS registrar do?
  - E.g. mark as conflict and clear from cache, or
  - Keep in cache and try to trigger the other mDNS host to mark it as conflict?

Scope Question – Use of SRP Replication

- **Are specific requirements for interworking with SRP Replication in scope of the TSR document?**
- Proposal (my opinion): that should be in the SRP replication draft.
- Example topic: if multiple replicated proxies hold the same data, which one(s) respond to a query? #18