

dnsop  
Internet-Draft  
Intended status: Informational  
Expires: July 26, 2020

D. Migault  
Ericsson  
January 23, 2020

DNS over Foo Discovery Mechanism  
draft-mglt-dprive-add-dofoo-discovery-00

## Abstract

This document describes a mechanism that enables a DNS client to discover other DNS alternatives such as DoT or DoH for one specific domain. This document also extends this discovery mechanism to an Internet wide of open resolvers search - though this is expected to be described further in a separate document. While search are always initiated by the DNS client, this document also indicates how a resolver MAY indicate a DNS client its preference for using alternative flavors of transport layer.

## Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of [BCP 78](#) and [BCP 79](#).

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <https://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on July 26, 2020.

## Copyright Notice

Copyright (c) 2020 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to [BCP 78](#) and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must

Internet-Draft

DoFoo Discovery Mechanism

January 2020

include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

## Table of Contents

<a href="#">1.</a>	Requirements Notation . . . . .	<a href="#">2</a>
<a href="#">2.</a>	Introduction . . . . .	<a href="#">2</a>
<a href="#">3.</a>	Discovery mechanism associated to one domain . . . . .	<a href="#">2</a>
<a href="#">3.1.</a>	Inferring domain from resolvers IP address . . . . .	<a href="#">5</a>
<a href="#">4.</a>	Resolver advertising other service sub type . . . . .	<a href="#">5</a>
<a href="#">5.</a>	Migration to service sub types . . . . .	<a href="#">6</a>
<a href="#">6.</a>	Service discovery over the Internet . . . . .	<a href="#">6</a>
<a href="#">7.</a>	Security Considerations . . . . .	<a href="#">7</a>
<a href="#">7.1.</a>	Use of protected channel is RECOMMENDED . . . . .	<a href="#">7</a>
<a href="#">7.2.</a>	DNSSEC is RECOMMENDED . . . . .	<a href="#">8</a>
<a href="#">7.3.</a>	TLSA is RECOMMENDED . . . . .	<a href="#">8</a>
<a href="#">8.</a>	Privacy Considerations . . . . .	<a href="#">9</a>
<a href="#">9.</a>	IANA Considerations . . . . .	<a href="#">10</a>
<a href="#">10.</a>	Appendix . . . . .	<a href="#">10</a>
<a href="#">11.</a>	Normative References . . . . .	<a href="#">11</a>
	Author's Address . . . . .	<a href="#">11</a>

## [1.](#) Requirements Notation

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described [BCP 14](#) [[RFC2119](#)] [[RFC8174](#)] when, and only when, they appear in all capitals, as shown here.

## [2.](#) Introduction

## [3.](#) Discovery mechanism associated to one domain

The discovery mechanism is intended to enable a DNS client to discover what are the resolvers options available as well as how to further use these resolvers.

The procedure is based on service discovery [[RFC8145](#)] and the overall procedure consists in finding various instances of the service "rdns". The resolution service is designated as "rdns" and differs from the service "domain" defined by IANA. The motivation for

creating a new service is that "domain" is associated to port 53 as well as TCP and UDP and designates both the authoritative as well as the resolving service. On the other hand the service "rdns" is expected to be limited to the DNS resolution service that can have various transport flavors including using different ports.

In this document, the service "rdns" is associated to a domain such as example.com. This means that the discovery process is performed over a specific portion of the internet, and resolvers that have no relation to that domain are not expected to be found. It is expected that the domain may be provisioned as a configuration parameter in the DNS client. It is expected that the domain provides a good meaning of the administrative entity managing the resolver, as it reflects the trust/mistrust the end user puts in the resolution. This configuration parameters differs from the one that is currently provisioned and discussion on how to proceed to resolver discovery from a legacy provisioning is described in more details in [Section 3.1](#).

The DNS client then searches for the rdns service associated to the domain example.com by querying PTR RRsets associated to \_rdns\_udp.example.com. This query corresponds to the general case of DNS service discovery. While tcp is reserved for TCP only and DNS is not only running on top of TCP we use \_udp as a representation of \_srv.

The difference with service discovery is that the response is expected to return instances of the service type. These instances may offer completely different services, but the end user is expected to select them according to their human readable name. In our case, the rdns service type can be implemented into different sub services types that are in our cases (DOT, DOH DNS). DOT, DOH and DNS are only example and any other designation may have been provided. Possible ways to distinguish these services could have been to adopt a convention in the service instance names or to have standard value for the service names. We prefer not to take that path and remove any constraints on the service name as it usually appears to the end user and we want to leave it free to contain what is going to be meaningful for the end user. Typically, DOT, DOH or DNS are unlikely to be meaningful to the waste majority of the internet users. Instead we used the DNS-SD capabilities to specify sub services by prefixing with \_dot, \_doh and \_dns53 the dns.\_udp service type.

DNS client --> Resolver  
\_rdns.example.com PTR ?

<--  
\_rdns\_udp.example.com PTR DOT.\_dot.\_sub.\_rdns.\_udp.example.com  
\_rdns\_udp.example.com PTR DOH.\_doh\_sub.\_rdns.\_udp.example.com  
\_rdns\_udp.example.com PTR DNS.\_dns53\_sub.\_rdns.\_udp.example.com

Note that "DOT", "DOH" and "DNS" are the strings that may be shown to the end user. The main difference with DNS-SD is that the sub type

was initially designed so the end user can narrow down its search. More explicitly its purpose was to enable an end user to narrow down the search on services providing DNS resolution over HTTPS with \_doh.\_sub.\_rdns.\_udp.example.com. The purpose was not to split a generic service into multiple sub types of services.

Note that the user interface is expected to interpret and present to the end user the different services by interpreting the \_dot, \_doh or \_dns53 sub service types and easing the understanding of the end user. If the DNS client is implementing a specific configuration, it will also have to interpret the sub types according to the configuration of the end user.

Now that the end user has the various services available ("DOH", "DOT" and "DNS") with there associated types, the selection can occur, and the DNS client can request additional information about the service itself to set up a session with the chosen service. In our case this is mostly the host name, ports, the ip address, the certificates, .... If the DNS client choses to use DoH, for example, it will request the SRV RRsets associated to that service.

Note that in our case, the sub service type carries sufficient information and no additional information is needed. There is no need to request the TXT reccord. Note also that carrying the sub type into the TXT RRsets would not be appropriated as this is believe to be a sufficiently important information to prevent a DNS client to browse thought all the different service instances.

While the TXT RRset is not necessary now, it MAY contain additional

information that may be usefull to the DNS client as well.

It is expected these exchanges are protected with DNSSEC as these could be performed over an untrusted channel as well as through semi trusted resolver. The additional section SHOULD also carry the necessary information to set up the session between the DNS client and the resolver. This includes the IP addresses (A and AAAA) RRsets, for services implemented over TLS the necessary security credentials (TLSA RRsets).

```
DNS client --> Resolver
DOH._doh_sub._rdns._udp.example.com SRV ?
    <--
DOH._doh_sub._rdns.example.com SRV priority=0, weight=0,
                                port=443 host=resolver.example.com
DOH._doh_sub._rdns.example.com SRV priority=0, weight=1,
                                port=443 host=resolver.example.com
DOH._doh_sub._rdns.example.com RRSIG (SRV) <signature>
resolver.example.com AAAA <ip6_address>
resolver.example.com AAAA <ip6_address>
resolver.example.com RRSIG (A) <signature>
resolver.example.com TLSA <certificate>
resolver.example.com RRSIG (TLSA) <signature>
```

### [3.1.](#) Inferring domain from resolvers IP address

When an application such as an web browser has a DNS client as part of its components, the configuration of that DNS client can be part of the application configuration. In such case, the domain may be provisioned in the configuration either by the software vendor or manually by the end user. On the other hand, a non negligible part of the systems the resolver is automatically provided by the network

and designated by an IP address [[RFC3646](#)]. In such cases, there is a need to derive the domain associated to that domain name.

This section describes a procedure performed by the DNS client to derive the domain from the IP address. It is also expected that resolver adapt their naming convention so that the procedure works. More precisely, the domain will be derived from the IP address by:

1. performing a reverse resolution
2. assuming the resulting FQDN is composed of a hostname and the domain name. For example, if `resolver.example.com` is the resulting FQDN from the reverse resolution, then the domain will be `example.com`.

#### [4.](#) Resolver advertising other service sub type

A resolver receiving a DNS request over a service sub type MAY be willing to advertise the DNS client that other sub service type are available. This is especially useful, when, for example, a resolver wants that the DNS resolver switches to other service sub types that are more secure.

In order to do so the resolver MAY provide in the additional data field the appropriated SRV RRsets. As an example, if the resolver wants to advertise the existence of resolver using `dot` or `doh` sub

service type, the resolver would add the following RRsets. Additional RRsets such as A, AAAA or TLSA RRsets MAY also be added.

```
DOH._doh._sub_rdns.example.com SRV priority=0, weight=0,
                                port=443 host=resolver.example.com
DOH._doh._sub_rdns.example.com SRV priority=0, weight=1,
                                port=443 host=resolver.example.com
DOH._doh._sub_rdns.example.com RRSIG (SRV) <signature>
DOT._dot._sub_rdns.example.com SRV priority=0, weight=0,
                                port=443 host=resolver.example.com
DOT._dot._sub_rdns.example.com SRV priority=0, weight=1,
                                port=443 host=resolver.example.com
DOT._dot._sub_rdns.example.com RRSIG (SRV) <signature>
```

#### [5.](#) Migration to service sub types

The principle of the discovery mechanism is that the resolver indicates the available service sub types and let the DNS client chose which sub type it prefers. On the other hand, the resolver MAY also indicate a preference using the priority and weight fields however, there is no mechanisms that could permit an indirection from one service sub type to another service sub type. Redirection MAY especially be needed when a DNS client is using the dns53 sub type and the resolver would liek to upgrade the DNS client session to a more secure session. The MAY require a specific ERROR code that will request the DNS client to perform service discovery.

It is expected that dns53 sub type MUST always be provided to perform resolver discovery. In other words, resolver discovery MUST be available though the non confidential channels designated by the sub service type dns53. However, this does not mean that a resolver is expected to implement the dns53 sub type service for resolutions. The availability of the sub service types for resolution. If a resolver chose not to provide the dns53 sub service type, that service MUST NOT be pointed by the \_rdns.example.com search.

## [6.](#) Service discovery over the Internet

THIS SECTION NEEDS PROBABLY TO THE TOPIC OF A DEDICATED DRAFT.

The current document describes a search mechanism over a single domain. This is mostly useful for one resolver to anounce availability of other sub service types as well as for resolver to discover available alternatives. However, this requires the knowledge of a domain. The domain can be provisionned out-of band and results from a configuration setting. In the case of local scope resolver, it can also be derived from a provisioned IP address. This section aims at extending the ability for one DNS client to learn

about the different available domain associated to a resolver on the Internet. This section will list open resolvers that are available.

The mechanism involves the creation of a special domain name rdns.arpa that will list the various rdns domains. This mechanism remains valid as long as the list of rdns domain name remains relatively limited. The number of rdns domain that can fit into a payload will depend on the length of the rnds domain, so rdns domains

are expected to have limited length. However the compactness is not expected to match the one achieved for the root servers that are designated by a one character size identity. The reason for it is that the identity of the resolver is expected to carry some meaning to the DNS client as opposed to the root servers.

That said, a UDP packet of 4096 bytes is expected to contain a significant amount of resolvers. The number of open resolver is not expected to reach that limit and if so the list can be retrieved through TCP.

The traffic associated to that domain is expected to be limited as most applications are expected to be provisioned with that RRset.

```
b._rdns.rdns.arpa PTR <rdns domain0>
b._rdns.rdns.arpa PTR <rdns domain1>
[...]
```

TBD: \* IANA procedure to register \* criterias that needs to be met to appear in the list

QUESTION: \* Do we need to add some criteria for the search ?

## [7.](#) Security Considerations

### [7.1.](#) Use of protected channel is RECOMMENDED

When available, it is recommended to chose a protected version of the rdns service. More specifically, the use of end-to-end protection ensures that the DNS client is connected to the expected platform and that its traffic cannot be intercepted on path. Typically, the selection of resolver on the Internet (and not on your ISP network) and the use of a non protected channel enables an attacker to monitor your DNS traffic. The similar observation remains true if you are connected to the resolver of your ISP. It is commonly believed that trusting your ISP (that is your first hop) makes encryption unnecessary. Trusting your ISP is mandatory in any case, but the associated level of trust with an protected channel is restricted to the operation of the DNS platform. With non protected channel the trust is extended to any segment between the DNS client and the

resolver, which is consequently larger. The use of a protected



channel is recommended as it will prevent anyone on path to monitor your traffic.

## [7.2.](#) DNSSEC is RECOMMENDED

The exchanges SHOULD be protected with DNSSEC to ensure integrity of the information between the authoritative servers and the DNS client. Without DNSSEC protection, DNS messages may be tampered typically when they are transmitted over an unprotected channel either between the DNS client and the resolver or between the resolver and the authoritative servers. The messages may be tampered by an online attacker intercepting the messages or by the intermediary devices. It is important to realize that protection provided by TLS is limited to the channel between the DNS client and the resolver. There are a number of cases where the trust in the resolver is not sufficient which justify the generalization of the use of DNSSEC. The following examples are illustrative and are intended to be exhaustive.

First, the discovery exchanges may happen over an unprotected channel, in which case, the messages exchanged may be tampered by anyone on-path between the DNS client and the resolver as well as between the resolver and the authoritative servers – including the resolver. When TLS is used between the DNS client and the resolver, this does not necessarily mean the DNS client trusts the resolver. Typically, the TLS session may be established with a self-signed certificate in which case the session is basically protected by a proof-of-ownership. In other cases, the session may be established based on Certificate Authorities (CA) that have been configured into the TLS client, but that are not necessarily trusted by the DNS client. In such cases, the connected resolver may be used to discover resolvers from another domain. In this case, the resolver is probably interacting with authoritative servers using untrusted and unprotected channels. Integrity protection relies on DNSSEC.

## [7.3.](#) TLSA is RECOMMENDED

When TLS is used to protect the DNS exchanges, certificates or fingerprint SHOULD be provided to implement trust into the communication between the DNS client and the resolver. The TLS session and the association of the private key to a specific identity can be based on two different trust models. The Web PKI that will rely on CA provisioned in the TLS library or the TA provided to the DNS client. A DNS client SHOULD be able to validate the trust of a TLS session based on the DNSSEC trust model using DANE.

When the DNS client is protecting its session to the resolver via TLS, the DNS client may initiate a TLS session that is not validated

by a CA or a TLSA RRsets. The DNS client MUST proceed to the discovery process and validate the certificate match the TLSA RRset. In case of mismatch the DNS client MUST abort the session.

## 8. Privacy Considerations

When the discovery protocol is performed using a resolver that belongs to one domain for another domain, or over an unprotected channel, the DNS client must be conscious that its is revealing to the resolver its intention to use another resolver. More specifically, suppose an resolver is complying some legal requirements that DNS traffic must be unencrypted. Using this resolver to perform a resolver discovery reveals the intention of potentially using alternative resolvers. Alternatively, narrowing down the discovery over a specific sub type of resolver (DoT, or DoH) may reveal to that resolver the type of communication. As result, when performing a discovery over a domain that differs to the domain the resolver belongs to, it is RECOMMENDED to request the SRV RRsets associated to all different sub type of proposed services.

The absence of traffic that results from switching completely to a newly discovered resolver right after the discovery process provides an indication to the resolver the DNS client is switching to. It is hard to make that switch unnoticed to the initial resolver and the DNS resolver MUST assume this will be noticed. The information of switching may be limited by sharing the traffic between different resolvers, however, the traffic pattern associated to each resolver may also reveal the switch. In addition, when the initial resolver is provided by the ISP, the ISP is also able to monitor the IP traffic and infer the switch. As a result, the DNS client SHOULD assume the switch will be detected.

With DoT or DoH, the selection of port 443 will make the traffic indistinguishable from HTTPS traffic. This means that an observer will not be able to tell whether the traffic carries web traffic or DNS traffic. Note that it presents an interest if the server offers both a web service as well as a resolution service. Note that many resolvers have a dedicated IP address for the resolution service, in which case, the information will be inferred from the IP address. Note also that traffic analysis may infer this as well. Typically suppose an IP address hosts one or multiple web sites that are not popular as well as a resolving service. If this IP address is associated frequent short size exchanges, it is likely that these exchanges will be DNS exchanges rather than Web traffic. The size of the packet may also be used as well as many other patterns. As a result, the use port 443 to hide the DNS traffic over web traffic

should be considered as providing limited privacy.

## 9. IANA Considerations

This document requests the IANA the creation of a new service name in the Service Name and Transport Protocol Port Number Registry

<https://www.iana.org/assignments/service-names-port-numbers/service-names-port-numbers.xml>

Fields Port Number, Transport Protocol, Assignee, Contact, Modification Date, Service Unauthorized Use Report, Assignment Notes are void.

Service Name	Description	Registration Date	Reference
rdns	DNS resolution	TBD1	RFC-TBD

This document requests the IANA the creation of the following underscored node names in the Underscored and Globally Scoped DNS Node Names registry <https://www.iana.org/assignments/dns-parameters/dns-parameters.xhtml#dns-parameters-14>

RR Type	_NODE NAME	Reference
SRV	_rdn	RFC-TBD
SRV	_dot	RFC-TBD
SRV	_doh	RFC-TBD
SRV	_dns53	RFC-TBD

## 10. Appendix

Example of a file.

\_rdns\_udp.example.com PTR DOT.\_dot.\_sub.\_rdns.\_udp.example.com  
\_rdns\_udp.example.com PTR DOH.\_doh\_sub.\_rdns.\_udp.example.com  
\_rdns\_udp.example.com PTR DNS.\_dns53\_sub.\_rdns.\_udp.example.com

\_dot\_sub\_rdns.example.com PTR DOT.\_dot\_sub\_rdns.\_udp.example.com  
\_doh\_sub\_rdns.example.com PTR DOH.\_doh\_sub\_rdns.\_udp.example.com  
\_dns53\_sub\_rdns.example.com PTR DNS.\_dns53\_sub\_rdns.\_udp.example.com

DOT.\_dot\_sub\_rdns.example.com SRV port=443 host=dns.example.com  
DOT.\_dot\_sub\_rdns.example.com SRV port=53 host=dns.example.com  
DOH.\_dot\_sub\_rdns.example.com SRV port=443 host=dns-dot.example.com  
DNS.\_dns53\_sub\_rdns.example.com SRV port=53 host=dns53.example.com

dns.example.com AAAA  
dns.example.com TLSA  
dns.example.com RRSIG

dns53.example.com AAAA  
dns53.example.com RRSIG

## 11. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", [BCP 14](#), [RFC 2119](#), DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/info/rfc2119>>.
- [RFC3646] Droms, R., Ed., "DNS Configuration options for Dynamic Host Configuration Protocol for IPv6 (DHCPv6)", [RFC 3646](#), DOI 10.17487/RFC3646, December 2003, <<https://www.rfc-editor.org/info/rfc3646>>.

[RFC8145] Wessels, D., Kumari, W., and P. Hoffman, "Signaling Trust Anchor Knowledge in DNS Security Extensions (DNSSEC)", [RFC 8145](#), DOI 10.17487/RFC8145, April 2017, <<https://www.rfc-editor.org/info/rfc8145>>.

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in [RFC 2119](#) Key Words", [BCP 14](#), [RFC 8174](#), DOI 10.17487/RFC8174, May 2017, <<https://www.rfc-editor.org/info/rfc8174>>.

Author's Address

Migault

Expires July 26, 2020

[Page 11]

---

Internet-Draft

DoFoo Discovery Mechanism

January 2020

Daniel Migault  
Ericsson  
8275 Trans Canada Route  
Saint Laurent, QC H4S 0B6  
Canada

EMail: [daniel.migault@ericsson.com](mailto:daniel.migault@ericsson.com)

