dnsop
Internet-Draft
Intended status: Informational
Expires: May 19, 2015

C. Contavalli
W. van der Gaast
Google
D. Lawrence
Akamai Technologies
W. Kumari
Google

November 15, 2014

Client Subnet in DNS Requests
draft-ietf-dnsop-edns-client-subnet-00

Abstract

This draft defines an EDNS0 extension to carry information about the network that originated a DNS query, and the network for which the subsequent reply can be cached.

IESG Note

[RFC Editor: Please remove this note prior to publication]

This informational document describes an existing, implemented and deployed system. A subset of the operators using this is at http://www.afasterinternet.com/participants.htm . The authors believe that it is better to document this system (even if not everyone agrees with the concept) than leave it undocumented and proprietary.

Table of Contents

1. Introduction

Many Authoritative Nameservers today return different replies based on the perceived topological location of the user. These servers use the IP address of the incoming query to identify that location. Since most queries come from intermediate Recursive Resolvers, the source address is that of the Recursive Resolver rather than of the query originator.

Traditionally, and probably still in the majority of instances, Recursive Resolvers are reasonably close in the network topology to the Stub Resolvers or Forwarders that are the source of queries. For these resolvers, using their own IP address is sufficient for authority servers that tailor responses based upon location of the querier.

Increasingly, though, a class of Recursive Resolvers has arisen that handle query sources that are often not topologically close. The motivation for a user to configure such a Centralized Resolver varies but is usually because of some enhanced experience, such as greater cache security or applying policies regarding where users may connect. (Although political censorship usually comes to mind here, the same actions may be used by a parent when setting controls on where a minor may connect.) Similarly, many ISPs and other organizations use a Centralized Resolver infrastructure that can be distant from the clients the resolvers serve. The cases all lead to less than optimal replies from topology-sensitive Authoritative Nameservers.

This draft defines an EDNS0 [RFC6891] option to convey network information that is relevant to the DNS message. It will carry sufficient network information about the originator for the Authoritative Nameserver to tailor responses. It will also provide for the Authoritative Nameserver to indicate the scope of network addresses for which the tailored answer is intended. This EDNS0 option is intended for those recursive and authority servers that would benefit from the extension and not for general purpose deployment. It is completely optional and can safely be ignored by servers that choose not to implement it or enable it.

This draft also includes guidelines on how to best cache those results and provides recommendations on when this protocol extension should be used.

2. Requirements Notation

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].

3. Terminology

Stub Resolver: A simple DNS protocol implementation on the client side as described in [RFC1034] section 5.3.1.

Authoritative Nameserver: A nameserver that has authority over one or more DNS zones. These are normally not contacted by clients directly but by Recursive Resolvers. Described in [RFC1035] chapter 6.

Recursive Resolver: A nameserver that is responsible for resolving domain names for clients by following the domain's delegation chain. Recursive Resolvers frequently use caches to be able to respond to client queries quickly. Described in [RFC1035] chapter 7.

Intermediate Nameserver: Any nameserver (possibly a Recursive Resolver) in between the Stub Resolver and the Authoritative Nameserver.

Centralized Resolvers: Recursive Resolvers that serve a topologically diverse network address space.

Optimized Reply: A reply from a nameserver that is optimized for the node that sent the request, normally based on performance (i.e. lowest latency, least number of hops, topological distance, ...).

Topologically Close: Refers to two hosts being close in terms of number of hops or time it takes for a packet to travel from one host to the other. The concept of topological distance is only loosely related to the concept of geographical distance: two geographically close hosts can still be very distant from a topological perspective, and two geographically distant hosts can be quite close on the network.

4. Overview

The general idea of this document is to provide an EDNS0 option to allow Recursive Resolvers, if they are willing, to forward details about the origin network from which a query is coming when talking to Authoritative Nameservers.

The format of this option is described in Section 5, and is meant to be added in queries sent by Intermediate Nameservers in a way transparent to Stub Resolvers and end users, as described in Section 6.1.

As described in Section 6.2, an Authoritative Nameserver could use this EDNS0 option as a hint to better locate the network of the end user and provide a better answer.

Its reply would also contain an edns-client-subnet option, clearly indicating that the server made use of this information, and that the answer is tied to the network of the client.

As described in Section 6.3, Intermediate Nameservers would use this information to cache the reply.

Some Intermediate Nameservers may also have to be able to forward edns-client-subnet queries they receive. This is described in Section 6.4.

The mechanisms provided by edns-client-subnet raise various security related concerns, related to cache growth, the ability to spoof EDNS0 options, and privacy. Section 10 explores various mitigation techniques.

The expectation, however, is that this option will only be used by Recursive Resolvers and Authoritative Nameservers that incur geolocation issues.

Most Recursive Resolvers, Authoritative Nameservers and Stub Resolvers will never know about this option, and will continue working as they had been.

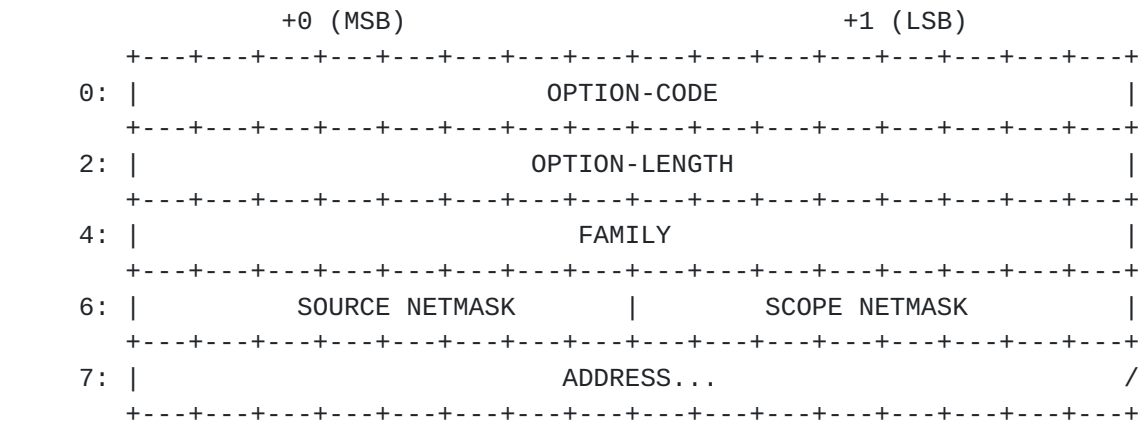
Failure to support this option or its improper handling will, at worst, cause suboptimal identification of client location, which is a common occurrence in current content delivery network (CDN) setups and not a cause of concern.

Section 6.1 also provides a mechanism for Stub Resolvers to signal Recursive Resolvers that they do not want edns-client-subnet treatment for specific requests.

Additionally, operators of Intermediate Nameservers with edns-client-subnet enabled are allowed to choose how many bits of the address of received queries to forward, or to reduce the number of bits forwarded for queries already including an edns-client-subnet option.

5. Option Format

This draft uses an EDNS0 [RFC6891]) option to include client address information in DNS messages. The option is structured as follows:



- (Defined in [RFC6891]) OPTION-CODE, 2 octets, for edns-client-subnet is 8 (0x00 0x08).

- (Defined in [RFC6891]) OPTION-LENGTH, 2 octets, contains the length of the payload (everything after OPTION-LENGTH) in octets.

- FAMILY, 2 octets, indicates the family of the address contained in the option, using address family codes as assigned by IANA in IANA-AFI [2].

The format of the address part depends on the value of FAMILY. This document only defines the format for FAMILY 1 (IP version 4) and 2 (IP version 6), which are as follows:

- SOURCE NETMASK, unsigned octet representing the length of the netmask pertaining to the query. In replies, it mirrors the same value as in the requests. It can be set to 0 to disable client-based lookups, in which case the ADDRESS field MUST be absent.

- SCOPE NETMASK, unsigned octet representing the length of the netmask pertaining to the reply. In requests, it SHOULD be set to the longest cacheable length supported by the Intermediate Nameserver. For backwards compatibiilty with draft versions of this specification, in requests it MAY be set to 0 to have the Authoritative Nameserver treat the longest cacheable length as the SOURCE NETMASK length. In responses, this field is set by the Authoritative Nameserver to indicate the coverage of the response. It might or might not match SOURCE NETMASK; it could be shorter or longer.

- ADDRESS, variable number of octets, contains either an IPv4 or IPv6 address, depending on FAMILY, truncated in the request to the number of bits indicated by the SOURCE NETMASK field, with bits set to 0 to pad up to the end of the last octet used. (This need not be as many octets as a complete address would take.) In the reply, if the SCOPE NETMASK of the request was 0 then ADDRESS must contain the same octets as in the request. Otherwise, the bits for ADDRESS will be significant through the maximum of the SOURCE NETMASK or SCOPE NETMASK, and 0 filled to the end of an octet.

All fields are in network byte order ("big-endian", per [RFC1700], Data Notation).

6. Protocol Description

6.1. Originating the Option

The edns-client-subnet option should generally be added by Recursive Resolvers when querying other servers, as described in Section 11.

In this option, the server should include the IP address of the client that caused the query to be generated, truncated to the number of bits specified in the SOURCE NETMASK field.

A Stub Resolver MAY generate DNS queries with an edns-client-subnet option with SOURCE NETMASK set to 0 (i.e. 0.0.0.0/0) to indicate that the Recursive Resolver MUST NOT add address information of the client to its queries. The subsequent Recursive Resolver query to the Authoritative Nameserver will then either not include an edns-client-subnet option or MAY optionally include its own address information, which is what the Authoritative Nameserver will use anyway to generate the reply in lieu of no option. This allows the answer to be handled by the same caching mechanism as other requests, with an explicit indicator of the applicable scope. Subsequent Stub Resolver requests for /0 can then be answered from this cached response.

The Stub Resolver may also add non-empty edns-client-subnet options to its queries, but Recursive Resolvers are not required to use this information.

For privacy reasons, and because the whole IP address is rarely required to determine an optimized reply, the ADDRESS field in the option SHOULD be truncated to a certain number of bits, chosen by the administrators of the Intermediate Nameserver, as described in Section 10.

If the Stub Resolver requests additional privacy via a SOURCE NETMASK that is shorter than the maximum cacheable SCOPE NETMASK that the

Recursive Resolver allows, the Recursive Resolver SHOULD issue the query with its longer SCOPE NETMASK.

6.2. Generating a Response

When a query containing an edns-client-subnet option is received, an Authoritative Nameserver supporting edns-client-subnet MAY use the address information specified in the option in order to generate an optimized reply.

Authoritative Nameservers that have not implemented or enabled support for the edns-client-subnet option may safely ignore it within incoming queries. Per [RFC6891] section 6.1.2, such a server MUST NOT include an edns-client-subnet option within replies, to indicate lack of support for the option.

Requests with wrongly formatted options (e.g., wrong size) MUST be rejected and a FORMERR response MUST be returned to the sender, as described by [RFC6891], Transport Considerations.

If the Authoritative Nameserver decides to use information from the edns-client-subnet option to calculate a response, it MUST include the option in the response to indicate that the information was used and SHOULD be cached accordingly. If the option was not included in a query, it MUST NOT be included in the response.

The FAMILY and SOURCE NETMASK in the response MUST match those in the request. The first SOURCE NETMASK bits of the ADDRESS in the response MUST match those in the request, even if fewer bits were used to form the response. Echoing back the address and netmask helps to mitigate certain attack vectors, as described in Section 10.

The SCOPE NETMASK in the reply indicates the netmask of the network for which the answer is intended.

A SCOPE NETMASK value longer than the SOURCE NETMASK indicates that the address and netmask provided in the query was not specific enough to select a single, best response. The ADDRESS MUST be extended to SCOPE NETMASK significant bits to indicate the network for which it is optimal, but the Recursive Resolver SHOULD still provide the result as the answer to the triggering client request even if the client is in a different address range.

Conversely, a shorter SCOPE NETMASK indicates that more bits than necessary were provided, and the answer is suitable for a broader range of addresses.

If a non-zero SCOPE NETMASK was supplied in the request, the SCOPE NETMASK of the response MUST be no longer than the SCOPE NETMASK of the request.

As not all netblocks are the same size, an Authoritative Nameserver may return different values of SCOPE NETMASK for different networks.

In both cases, the value of the SCOPE NETMASK in the reply has strong implications with regard to how the reply will be cached by Intermediate Nameservers, as described in Section 6.3.

If the edns-client-subnet option in the request is not used at all, a server supporting edns-client-subnet MUST indicate that no bits of the ADDRESS in the request have been used by specifying a SCOPE NETMASK of 0, equivalent to the networks 0.0.0.0/0 or ::/0. This could happen, for example, because the reply is invariant across the network space. The answer is suitable for all clients for the duration of its TTL.

The specific logic that an Authoritative Nameserver uses to choose an optimized reply is not in the scope of this document. Implementers are encouraged, however, to consider carefully their selection of SCOPE NETMASK for the reply in the event that an optimal reply cannot be determined.

6.3. Handling edns-client-subnet Replies and Caching

When an Intermediate Nameserver receives a reply containing an edns-client-subnet option, it will return a reply to its client and SHOULD cache the result.

If the FAMILY, SOURCE NETMASK, and SOURCE NETMASK bits of ADDRESS in the reply don't match the fields in the corresponding request, the full reply MUST be dropped, as described in Section 10.

In the cache, any resource record in the answer section will be tied to the network specified by the FAMILY, ADDRESS and SCOPE NETMASK fields, as detailed below. Note that the additional and authority sections from a DNS response message are specifically excluded here. Any information cached from these sections MUST NOT be tied to a network.

If another query is received matching the name and type of an entry in the cache, the resolver will check whether the FAMILY and ADDRESS of the client matches one of the networks in the cache for that entry.

If the address of the client is within any of the networks in the cache, then the cached response MUST be returned as usual. If the address of the client matches multiple networks in the cache, the entry with the longest SCOPE NETMASK value MUST be returned, as with most route-matching algorithms.

If the address of the client does not match any network in the cache, then the Recursive Resolver MUST behave as if no match was found and perform resolution as usual. This is necessary to avoid suboptimal replies in the cache from being returned to the wrong clients, and to avoid a single request coming from a client on a different network from polluting the cache with a suboptimal reply for all the users of that resolver.

Note that every time a Recursive Resolver queries an Authoritative Nameserver by forwarding the edns-client-subnet option that it received from another client, a short SOURCE NETMASK in the original request could cause a suboptimal reply to be returned by the Authoritative Nameserver.

When the request includes a longer SCOPE NETMASK, the reply returned may still be cached as optimal for the ADDRESS and SCOPE NETMASK of the reply. This might still be suboptimal for the original client.

To avoid this suboptimal reply from being served from cache for other clients for which a better reply would be available, the Recursive Resolver MUST check the SCOPE NETMASK that was returned by the Authoritative Nameserver:

- If the SCOPE NETMASK in the reply is longer than the SOURCE NETMASK, it means that the reply might be suboptimal. A Recursive Resolver MUST return this entry from cache only to queries that do not contain or allow a longer SOURCE NETMASK to be forwarded, or to queries originating from the network covered by the ADDRESS and SCOPE NETMASK..

- If the SCOPE NETMASK in the reply is shorter than or equal to the SOURCE NETMASK, the reply is optimal, and SHOULD be returned from cache to any client within the network indicated by ADDRESS and SCOPE NETMASK.

As another reply is received, the reply will be tied to a different network. The server SHOULD keep in cache both replies, and return the most appropriate one depending on the address of the client. Per standard DNS caching behavior, all records SHOULD be retained until their TTL expires. Subsequent queries to refresh the data should always specify the longest SCOPE NETMASK that the Recursive Resolver

is willing to cache, even if previous responses indicated that a shorter netmask was the optimal response.

Although omitting network-specific caching will significantly simplify an implementation, the resulting drop in cache hits is very likely to defeat most latency benefits provided by edns-client-subnet. Therefore, when implementing this option for latency purposes, implementing full caching support as described in this section is STRONGLY RECOMMENDED.

Any reply containing an edns-client-subnet option considered invalid should be treated as if no edns-client-subnet option was specified at all.

Replies coming from servers not supporting edns-client-subnet or otherwise not containing an edns-client-subnet option SHOULD be considered as containing a SCOPE NETMASK of 0 (e.g., cache the result for 0.0.0.0/0 or ::/0) for all the supported families.

In any case, the response from the resolver to the client MUST NOT contain the edns-client-subnet option if none was present in the client's original request. If the original client request contained a valid edns-client-subnet option that was used during recursion, the Recursive Resolver MUST include the edns-client-subnet option from the Authoritative Nameserver response in the response to the client.

Enabling support for edns-client-subnet in a recursive resolver will significantly increase the size of the cache, reduce the number of results that can be served from cache, and increase the load on the server. Implementing the mitigation techniques described in Section 10 is strongly recommended.

6.4. Transitivity

Generally, edns-client-subnet options will only be present in DNS messages between a Recursive Resolver and an Authoritative Nameserver, i.e., one hop. In certain configurations however, for example multi-tier nameserver setups, it may be necessary to implement transitive behaviour on Intermediate Nameservers.

It is important that any Intermediate Nameserver that forwards edns-client-subnet options received from their clients MUST fully implement the caching behaviour described in Section 6.3.

Intermediate Nameservers, including Recursive Resolvers, supporting edns-client-subnet MUST forward options with SOURCE NETMASK set to 0 (i.e., completely anonymized), such an option MUST NOT be replaced with an option with more accurate address information.

An Intermediate Nameserver MAY also forward edns-client-subnet options with actual address information. This information MAY match the source IP address of the incoming query, and MAY have more or less address bits than the Nameserver would normally include in a locally originated edns-client-subnet option.

If for any reason the Intermediate Nameserver does not want to use the information in an edns-client-subnet option it receives (too little address information, network address from a range not authorized to use the server, private/unroutable address space, etc), it SHOULD drop the query and return a REFUSED response. Note again that an edns-client-subnet option with 0 address bits MUST NOT be refused.

7. IANA Considerations

IANA has already assigned option code 8 in the "DNS EDNS0 Option Codes (OPT)" registry to edns-client-subnet.

The IANA is requested to update the reference ("draft-vandergaast-edns-client-subnet") to refer to this RFC when published.

8. DNSSEC Considerations

The presence or absence of an [RFC6891] EDNS0 OPT resource record containing an edns-client-subnet option in a DNS query does not change the usage of the resource records and mechanisms used to provide data origin authentication and data integrity to the DNS, as described in [RFC4033], [RFC4034] and [RFC4035]. OPT records are not signed.

9. NAT Considerations

Special awareness of edns-client-subnet in devices that perform Network Address Translation (NAT) as described in [RFC2663] is not required; queries can be passed through as-is. The client's network address SHOULD NOT be added, and existing edns-client-subnet options, if present, SHOULD NOT be modified by NAT devices.

In large-scale global networks behind NAT (but, for example, with a Centralized Resolver infrastructure), an internal Intermediate Nameserver might have detailed network layout information, and might know which external subnets are used for egress traffic by each internal network. In such cases, the Intermediate Nameserver MAY use that information when originating edns-client-subnet options.

In other cases, Recursive Resolvers sited behind a NAT device SHOULD NOT originate edns-client-subnet options with their IP address, and

instead rely on downstream Intermediate Nameservers doing so. They MAY, however, choose to include the option with their internal address for the purposes of signaling a shorter, more anonymous SOURCE NETMASK.

If an Authoritative Nameserver on the publicly routed Internet receives a request that specifies an ADDRESS in [RFC1918] or [RFC4193] private address space, it SHOULD ignore ADDRESS and look up its answer based on the address of the Recursive Resolver. In the reply it SHOULD set SCOPE NETMASK to cover all of the relevant private space. For example, a request for ADDRESS 10.1.2.0 with a SOURCE NETMASK of 24 would get a returned SCOPE NETMASK of 8. The Intermediate Nameserver MAY elect to cache the answer under one entry for special-purpose addresses [RFC6890]; see Section 10.3.

10. Security Considerations

10.1. Privacy

With the edns-client-subnet option, the network address of the client that initiated the resolution becomes visible to all servers involved in the resolution process. Additionally, it will be visible from any network traversed by the DNS packets.

To protect users' privacy, Recursive Resolvers are strongly encouraged to conceal part of the IP address of the user by truncating IPv4 addresses to 24 bits. No recommendation is provided for IPv6 at this time, but IPv6 addresses should be similarly truncated in order to not allow unique identification of the client.

When a non-zero SCOPE NETMASK is provided by the Recursive Resolver that is longer than SOURCE NETMASK, users can often obtain more optimal mapping if the resolver is well-used. Replies will have answers optimized up to SCOPE NETMASK bits for a subset of the SOURCE NETMASK. Subsequent requests within the TTL from clients within the cached range will be served the optimal answer, while still preserving privacy of the user.

ISPs will often have more detailed knowledge of their own networks. That is, they will know if all 24-bit prefixes in a /20 are in the same area. In those cases, for optimal cache utilization and improved privacy, the ISP's Recursive Resolver SHOULD truncate IP addresses in this /20 to just 20 bits, instead of 24 as recommended above.

Users who wish their full IP address to be hidden can include an edns-client-subnet option specifying the wildcard address 0.0.0.0/0 (i.e. FAMILY set to 1 (IPv4), SOURCE NETMASK to 0 and no ADDRESS).

As described in previous sections, this option will be forwarded across all the Recursive Resolvers supporting edns-client-subnet, which MUST NOT modify it to include the network address of the client.

Note that even without an edns-client-subnet option, any server queried directly by the user will be able to see the full client IP address. Recursive Resolvers or Authoritative Nameservers MAY use the source IP address of requests to return a cached entry or to generate an optimized reply that best matches the request.

10.2. Birthday Attacks

edns-client-subnet adds information to the DNS question tuple (q-tuple). This allows an attacker to send a caching Intermediate Nameserver multiple queries with spoofed IP addresses either in the edns-client-subnet option or as the source IP. These queries will trigger multiple outgoing queries with the same name, type and class, just different address information in the edns-client-subnet option.

With multiple queries for the same name in flight, the attacker has a higher chance of success in sending a matching response (with the address 0.0.0.0/0 to get it cached for many hosts).

To counter this, every edns-client-subnet option in a response packet MUST contain the FAMILY and SOURCE NETMASK fields from the corresponding request, along with identical ADDRESS bits for SOURCE NETMASK length. Intermediate Nameservers processing a response MUST verify that these match, and MUST discard the entire reply if they do not.

10.3. Cache Pollution

It is simple for an arbitrary resolver or client to provide false information in the edns-client-subnet option, or to send UDP packets with forged source IP addresses.

This could be used to:

- pollute the cache of intermediate resolvers, by filling it with results that will rarely (if ever) be used.

- reverse engineer the algorithms (or data) used by the Authoritative Nameserver to calculate the optimized answer.

- mount a denial-of-service attack against an Intermediate Nameserver, by forcing it to perform many more recursive queries

than it would normally do, due to how caching is handled for queries containing the edns-client-subnet option.

Even without malicious intent, Centralized Resolvers providing answers to clients in multiple networks will need to cache different replies for different networks, putting more memory pressure on the cache.

To mitigate those problems:

- Recursive Resolvers implementing edns-client-subnet should only enable it in deployments where it is expected to bring clear advantages to the end users. For example, when expecting clients from a variety of networks or from a wide geographical area. Due to the high cache pressure introduced by edns-client-subnet, the feature SHOULD be disabled in all default configurations.

- Recursive Resolvers SHOULD limit the number of networks and answers they keep in the cache for a given query.

- Recursive Resolvers SHOULD limit the number of total different networks that they keep in cache.

- Recursive Resolvers should never send edns-client-subnet options with a SCOPE NETMASK that is longer than they are willing to cache. Similarly, if using the backwards-compatible SCOPE NETMASK of 0, the request should not set a SOURCE NETMASK of more bits than they are willing to cache.

- Recursive Resolvers should implement algorithms to improve the cache hit rate, given the size constraints indicated above. Recursive Resolvers MAY, for example, decide to discard more specific cache entries first.

- Authoritative Nameservers and Recursive Resolvers should discard edns-client-subnet options that are either obviously forged or otherwise known to be wrong. They SHOULD at least treat unroutable addresses, such as some of the address blocks defined in [RFC6890], as equivalent to the Recursive Resolver's own identity. They SHOULD ignore and never forward edns-client-subnet options specifying other routable addresses that are known not to be served by the query source.

- Authoritative Nameservers consider the edns-client-subnet option just as a hint to provide better results. They can decide to ignore the content of the edns-client-subnet option based on black or white lists, rate limiting mechanisms, or any other logic implemented in the software.

11. Sending the Option

When implementing a Recursive Resolver, there are two strategies on deciding when to include an edns-client-subnet option in a query. At this stage, it's not clear which strategy is best.

11.1. Probing

A Recursive Resolver can send the edns-client-subnet option with every outgoing query. However, it is RECOMMENDED that Resolvers remember which Authoritative Nameservers did not return the option with their response, and omit client address information from subsequent queries to those Nameservers.

Additionally, Recursive Resolvers MAY be configured to never send the option when querying root, top-level, and effective top-level domain servers. These domains are delegation-centric and are very unlikely to generate different replies based on the address of the client.

When probing, it is important that several things are probed: support for edns-client-subnet, support for EDNS0, support for EDNS0 options, or possibly an unreachable Nameserver. Various implementations are known to drop DNS packets with OPT RRs (with or without options), thus several probes are required to discover what is supported.

Probing, if implemented, MUST be repeated periodically (i.e. daily). If an Authoritative Nameserver indicates edns-client-subnet support for one zone, it is to be expected that the Nameserver supports edns-client-subnet for all its zones. Likewise, an Authoritative Nameserver that uses edns-client-subnet information for one of its zones, MUST indicate support for the option in all its responses. If the option is supported but not actually used for generating a response, its SCOPE NETMASK value SHOULD be set to 0.

11.2. Whitelist

As described previously, it is expected that only a few Recursive Resolvers will need to use edns-client-subnet, and that it will generally be enabled only if it offers a clear benefit to the users.

To avoid the complexity of implementing a probing and detection mechanism (and the possible query loss/delay that may come with it), an implementation could decide to use a statically configured whitelist of Authoritative Namesevers to send the option to. Implementations MAY also allow additionally configuring this based on other criteria, such as zone or query type.

An additional advantage of using a whitelist is that partial client address information is only disclosed to Nameservers that are known to use the information, improving privacy.

A major drawback is scalability. The operator needs to track which Authoritative Nameservers support edns-client-subnet, making it harder for new Authoritative Nameservers to start using the option.

12. Example

1. A stub resolver SR with IP address 192.0.2.37 tries to resolve www.example.com, by forwarding the query to the Recursive Resolver R from IP address IP, asking for recursion.

2. RNS, supporting edns-client-subnet, looks up www.example.com in its cache. An entry is found neither for www.example.com, nor for example.com.

3. RNS builds a query to send to the root and .com servers. The implementation of R provides facilities so an administrator can configure RNS not to forward edns-client-subnet in certain cases. In particular, RNS is configured to not include an edns-client-subnet option when talking to delegation-centric nameservers, as described in Section 6.1. Thus, no edns-client-subnet option is added, and resolution is performed as usual.

4. RNS now knows the next server to query, Authoritative Nameserver ANS, responsible for example.com.

5. RNS prepares a new query for www.example.com, including an edns-client-subnet option with:

   * OPTION-CODE, set to 0x00 0x08.

   * OPTION-LENGTH, set to 0x00 0x07 for the following fixed 4 octets plus the 3 octets that will be used for ADDRESS.

   * FAMILY, set to 0x00 0x01 as IP is an IPv4 address.

   * SOURCE NETMASK, set to 0x18, as RNS is configured to conceal the last 8 bits of every IPv4 address.

   * SCOPE NETMASK, set to 0x1B, as RNS is willing to cache answers up to a /27.

   * ADDRESS, set to 0xC0 0x00 0x02, providing only the first 24 bits of the IPv4 address.

6. The query is sent. Server ANS understands and uses edns-client-subnet. It parses the edns-client-subnet option, and generates an optimized reply.

7. Due to the internal implementation of ANS, it finds an answer that is optimal for several /27 ranges within the ADDRESS/SOURCE NETMASK of the request. It chooses one randomly. (Note well, this is just one example of how ANS could pick a suitable answer. Other selection methods are possible.)

8. The Authoritative Nameserver ANS adds an edns-client-subnet option in the reply, containing:

   * OPTION-CODE, set to 0x00 0x08.

   * OPTION-LENGTH, set to 0x00 0x08 for the following fixed 4 octets plus the 4 octets that will be used for ADDRESS .

   * FAMILY, set to 0x00 0x01, the same as the request.

   * SOURCE NETMASK, set to 0x18, copied from the request.

   * SCOPE NETMASK, set to 0x1B, indicating a /27 network.

   * ADDRESS, set to 0xC0 0x00 0x02 0xE0, copied from the request.

9. RNS receives the reply containing an edns-client-subnet option. The resolver verifies that FAMILY, SOURCE NETMASK, and the SOURCE NETMASK bits of ADDRESS match the request. If not, the message is discarded.

10. The reply is interpreted as usual. Since the reply contains an edns-client-subnet option, the ADDRESS, SCOPE NETMASK, and FAMILY in the response are used to cache the entry.

11. RNS sends a response to stub resolver SR, without including an edns-client-subnet option.

12. RNS receives another request to resolve www.example.com. This time, a reply is cached. The reply, however, is tied to a particular network. If the address of the client matches any network in the cache, then the reply is returned from the cache. Otherwise, another query is performed. If multiple results match, the one with the longest SCOPE NETMASK is chosen, as per common best-network match algorithms.

13. Contributing Authors

The below individuals contributed significantly to the draft. The RFC Editor prefers a maximum of 5 names on the front page, and so we have listed additional authors in this section

Edward Lewis
ICANN
12025 Waterfront Drive, Suite 300 Los Angeles, CA 90094-2536 USA
Email: edward.lewis@icann.org

Sean Leach
Fastly
POBox 78266
San Francisco, CA 94107

14. Acknowledgements

The authors wish to thank Darryl Rodden for his work as a co-author on previous versions, and the following people for reviewing early drafts of this document and for providing useful feedback: Paul S. R. Chisholm, B. Narendran, Leonidas Kontothanassis, David Presotto, Philip Rowlands, Chris Morrow, Kara Moscoe, Alex Nizhner, Warren Kumari, Richard Rabbat from Google, Terry Farmer, Mark Teodoro, Edward Lewis, Eric Burger from Neustar, David Ulevitch, Matthew Dempsky from OpenDNS, Patrick W. Gilmore and Jason Moreau from Akamai, Colm MacCarthaigh, Richard Sheehan and all the other people that replied to our emails on various mailing lists.

15. References

15.1. Normative References

[RFC1034] Mockapetris, P., "Domain names - concepts and facilities", STD 13, RFC 1034, November 1987.

[RFC1035] Mockapetris, P., "Domain names - implementation and specification", STD 13, RFC 1035, November 1987.

[RFC1700] Reynolds, J. and J. Postel, "Assigned Numbers", RFC 1700, October 1994.

[RFC1918] Rekhter, Y., Moskowitz, R., Karrenberg, D., Groot, G., and E. Lear, "Address Allocation for Private Internets", BCP 5, RFC 1918, February 1996.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC4033] Arends, R., Austein, R., Larson, M., Massey, D., and S. Rose, "DNS Security Introduction and Requirements", RFC 4033, March 2005.

[RFC4034] Arends, R., Austein, R., Larson, M., Massey, D., and S. Rose, "Resource Records for the DNS Security Extensions", RFC 4034, March 2005.

[RFC4035] Arends, R., Austein, R., Larson, M., Massey, D., and S. Rose, "Protocol Modifications for the DNS Security Extensions", RFC 4035, March 2005.

[RFC4193] Hinden, R. and B. Haberman, "Unique Local IPv6 Unicast Addresses", RFC 4193, October 2005.

[RFC6890] Cotton, M., Vegoda, L., Bonica, R., and B. Haberman, "Special-Purpose IP Address Registries", BCP 153, RFC 6890, April 2013.

[RFC6891] Damas, J., Graff, M., and P. Vixie, "Extension Mechanisms for DNS (EDNS(0))", STD 75, RFC 6891, April 2013.

15.2. Informative References

[RFC2663] Srisuresh, P. and M. Holdrege, "IP Network Address Translator (NAT) Terminology and Considerations", RFC 2663, August 1999.

15.3. URIs

[1] http://www.iana.org/assignments/address-family-numbers/

Appendix A. Document History

[RFC Editor: Please delete this section before publication.]

A.1. -00

- Document moved to experimental track, added experiment description in header with details in a new section.
- Specifically note that edns-client-subnet applies to the answer section only.
- Warn that caching based on edns-client-subnet is optional but very important for performance reasons.
- Updated NAT section.

- Added recommendation to not use the default /24 recommendation for the source netmask field if more detailed information about the network is available.

- Rewritten problem statement to be more clear about the goal of edns-client-subnet and the fact that it's entirely optional.

- Wire format changed to include the original address and netmask in responses in defence against birthday attacks.

- Security considerations now includes a section about birthday attacks.

- Renamed edns-client-ip in edns-client-subnet, following suggestions on the mailing list.

- Clarified behavior of resolvers when presented with an invalid edns-client-subnet option.

- Fully take multi-tier DNS setups in mind and be more clear about where the option should be originated.

- Added a few definitions in the Terminology section, and a few more aesthetic changes in the rest of the document.

A.2. -01

- Document version number reset from -02 to -00 due to the rename to edns-client-subnet.

- Clarified example (dealing with TLDs, and various minor errors).

- Referencing RFC5035 instead of RFC1918.

- Added a section on probing (and how it should be done) vs. whitelisting.

- Moved description on how to forward edns-client-subnet option in dedicated section.

- Queries with wrongly formatted edns-client-subnet options should now be rejected with FORMERR.

- Added an "Overview" section, providing an introduction to the document.

- Intermediate Nameservers can now remove an edns-client-subnet option, or reduce the SOURCE NETMASK to increase privacy.

- Added a reference to DoS attacks in the Security section.
- Don't use "network range", as it seems to have different meaning in other contexts, and turned out to be confusing.
- Use shorter and longer netmasks, rather than higher or lower. Add a better explanation in the format section.
- Minor corrections in various other sections.

A.3. -02

- Added IANA-assigned option code.

A.4. -03*

- [*] There was no -03 version of the draft; these changes, however, were made after -02.
- Allow non-zero SCOPE NETMASK for Recursive Resolvers to indicate their maximum cacheable mask length, and updated the example accordingly.
- A note on Authoritative Nameservers receiving requests that specify private address space.
- A note to always ask for the longest acceptable SCOPE NETMASK, even if a prior answer indicated that a shorter netmask was optimal.
- Marked up a couple of references.
- Minor grammatical consistency edits.

Authors' Addresses

Carlo Contavalli
Google
1600 Amphitheater Parkway
Mountain View, CA 94043
US

Email: ccontavalli@google.com

Wilmer van der Gaast
Google
Belgrave House, 76 Buckingham Palace Road
London SW1W 9TQ
UK

Email: wilmer@google.com

David C Lawrence
Akamai Technologies
8 Cambridge Center
Cambridge, MA 02142
US

Email: tale@akamai.com

Warren Kumari
Google
1600 Amphitheatre Parkway
Mountain View, CA 94043
US

Email: warren@kumari.net