dnsop
Internet-Draft
Intended status: Informational
Expires: April 6, 2015

C. Contavalli
W. van der Gaast
Google
S. Leach
VeriSign
D. Lawrence
Akamai
W. Kumari
Google

October 03, 2014

Client Subnet in DNS Requests
draft-vandergaast-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. It was originally going to be an Experimental document, but is now detailing a deployed system.

Status of This Memo

Table of Contents

1. Introduction 3
2. Requirements Notation 4
3. Terminology 4
4. Overview 5
5. Option Format 6
6. Protocol Description 7
6.1. Originating the Option 7
6.2. Generating a Response 7
6.3. Handling edns-client-subnet Replies and Caching 9
6.4. Transitivity 11
7. IANA Considerations 11
8. DNSSEC Considerations 11
9. NAT Considerations 12
10. Security Considerations 12
10.1. Privacy 12
10.2. Birthday Attacks 13
10.3. Cache Pollution 13
11. Sending the Option 14
11.1. Probing 15
11.2. Whitelist 15
12. Example 16
13. Contributing Authors 17
14. Acknowledgements 18
15. References 18
15.1. Normative References 18
15.2. Informative References 19

15.3. URIs 19
Appendix A. Document History 19
A.1. -00 19
A.2. -01 20
A.3. -02 20

Authors' Addresses 20

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 topological sense 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 serves query sources without regard to topology. The motivation for a query source to use such a Third-party 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.) When using a Third-party Resolver, there can no longer be any assumption of close proximity between the originator and the recursive resolver, leading to less than optimal replies from the authority servers.

A similar situation exists within some ISPs where the Recursive Resolvers are topologically distant from some edges of the ISP network, resulting in less than optimal replies from the authority servers.

This draft defines an EDNS0 option to convey network information that

is relevant to the message but not otherwise included in the datagram. This will provide the mechanism to carry sufficient network information about the originator for the authority server to tailor responses. It also provides for the authority server 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, starting at the root. 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.

Third-party Resolvers: Recursive Resolvers provided by parties that are not Internet Service Providers (ISPs). These services are often offered as substitutes for ISP-run nameservers.

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.

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 that a query is coming from when talking to other 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 contain an EDNS0 client-subnet option, clearly indicating that (1) the server made use of this information and (2) 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 enabled (and used) by Recursive Resolvers and Authoritative Nameserver that incur geolocation issues.

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

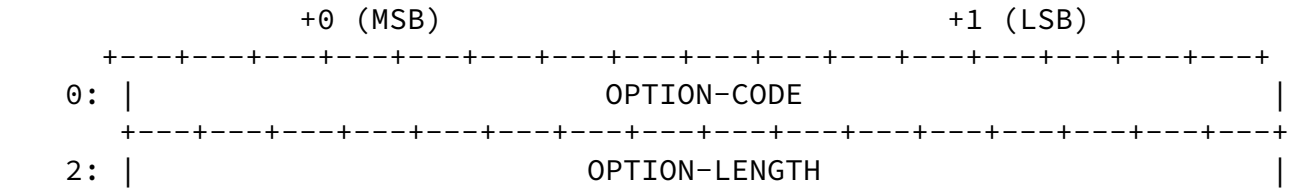
Failure to support this option or its improper handling will, at worst, cause sub-optimal geolocation, which is a pretty common occurrence in current CDN setups and not a cause of concern.

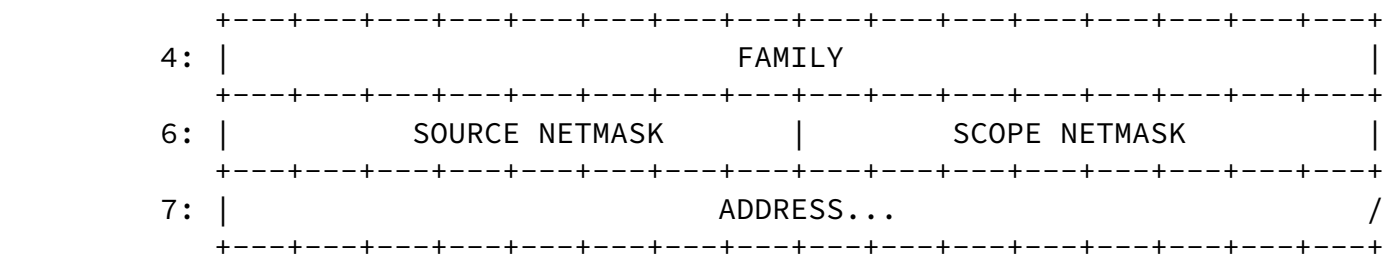
Section 6.1 also provides a mechanism for Stub Resolvers to signal Recursive Resolvers that they do not want an edns-client treatment for specific requests.

Additionally, owners of resolvers 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 IP information in DNS messages. The option is structured as follows:





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

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

- FAMILY, 2 octets, indicates the family of the address contained in the option, using address family codes as assigned by IANA in IANA-AFI [1].

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 byte representing the length of the netmask pertaining to the query. In replies, it mirrors the same value as in the requests.

- SCOPE NETMASK, unsigned byte representing the length of the netmask pertaining to the reply. In requests, it MUST be set to 0. In responses, this may or may not match SOURCE NETMASK.

- ADDRESS, variable number of octets, contains either an IPv4 or IPv6 address (depending on FAMILY), truncated 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.

All fields are in network byte order. Throughout the document, we will often refer to "longer" or "shorter" netmasks, corresponding to netmasks that have a "higher" or "lower" value when represented as integers.

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 of the client that caused the query to be generated, truncated to a number of bits specified in the SOURCE NETMASK field.

The IP of the client can generally be determined by looking at the source IP indicated in the IP header of the request.

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 Stub Resolver may also add non-empty edns-client-subnet options to its queries, but Recursive Resolvers are not required to accept/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 server, as described in Section 10.

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 servers that have not implemented or enabled support for the edns-client-subnet may safely ignore the option within incoming queries. 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 (i.e. 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 has to be cached accordingly). If the option was not included in a query, it MUST NOT be included in the response.

The FAMILY, ADDRESS and SOURCE NETMASK in the response MUST match those in the request. 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 larger than the SOURCE NETMASK indicates that the address and netmask provided in the query was not specific enough to select a single, best response, and that an optimal reply would require at least SCOPE NETMASK bits of address information.

Conversely, a shorter SCOPE NETMASK indicates that more bits than necessary were provided.

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 (for example, if an optimized reply was temporarily unavailable or not supported for the requested domain name), 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).

If no optimized answer could be found at all for the FAMILY, ADDRESS and SOURCE NETMASK indicated in the query, the Authoritative Nameserver SHOULD still return the best result it knows of (i.e. by using the query source IP address instead, or a sensible default), and indicate that this result should only be cached for the FAMILY, ADDRESS and SOURCE NETMASK indicated in the request. The server will indicate this by copying the SOURCE NETMASK into the SCOPE NETMASK field.

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 may cache the result.

If the FAMILY, ADDRESS and SOURCE NETMASK fields 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.

If another query is received matching the entry in the cache, the resolver will verify that the FAMILY and ADDRESS that represent the client match 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 highest 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 sub-optimal 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 sub-optimal 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 low SOURCE NETMASK in the original request could cause a sub-optimal reply to be returned by the Authoritative Nameserver.

To avoid this sub-optimal reply from being served from cache for 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 sub-optimal. 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.

- o If the SCOPE NETMASK in the reply is shorter 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.

When another request is performed, the existing entries SHOULD be kept in the cache until their TTL expires, as per standard behavior.

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.

Although omitting this behaviour 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 implements transitive behaviour (i.e. forward 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. 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 an IP range not authorized to use the server, private/unroutable address space, ...), 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 OPT resource record [TODO: Reference OPT] containing an edns-client-subnet option in a DNS query does not change the usage of those resource records and mechanisms used to provide data origin authentication and data integrity to the DNS, as described in [RFC4033], [RFC4034] and [RFC4035].

9. NAT Considerations

Special awareness of edns-client-subnet in devices that perform 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 centralized DNS infrastructure), an internal Intermediate Nameserver may have detailed network layout information, and may 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 NAT SHOULD NOT originate edns-client-subnet options with their external IP address, and instead rely on downstream Intermediate Nameservers doing so.

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.

ISPs will often have more detailed knowledge of their own networks. I.e. 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 edns-client-subnet options, 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 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 still get it cached for many hosts).

To counter this, every edns-client-subnet option in a response packet MUST contain the full FAMILY, ADDRESS and SOURCE NETMASK fields from the corresponding request. 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 caclulate the optimized answer.
- mount a DoS attack against an intermediate resolver, 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, Third-party Resolvers providing answers to clients in multiple networks will need to cache different replies for different networks, putting more 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 must 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 SOURCE NETMASKs providing more bits in the ADDRESS than they are willing to cache responses for.

- 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 known to be wrong or known to be forged edns-client-subnet options. They must at least ignore unroutable addresses, such as some of the address blocks defined in [RFC6890] and [RFC4193], and should ignore and never forward edns-client-subnet options specifying networks or addresses that are known not to be served by those servers when feasible.

- 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 and TLD servers, as these are unlikely to generate different replies based on the IP 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 (i.e. zone, qtype).

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

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

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

   * OPTION-CODE, set to 8.

   * OPTION-LENGTH, set to 0x00 0x07.

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

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

   * SCOPE NETMASK, set to 0x00, as specified by this document for all requests.

   * 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 the Authoritative Nameserver ANS, ANS finds a reply that is optimal for the whole /16 of the client that performed the request.

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

   * OPTION-CODE, set to 8.

   * OPTION-LENGTH, set to 0x00 0x07.

   * FAMILY, set to 0x00 0x01.

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

   * SCOPE NETMASK, set to 0x10, indicating a /16 network.

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

9. The Recursive Resolver R receives the reply containing an edns-client-subnet option. The resolver verifies that FAMILY, SOURCE NETMASK, and ADDRESS match the request. If not, the option 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. R sends a response to stub resolver SR, without including an edns-client-subnet option.

12. R 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

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 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.

[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.

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

Sean Leach
VeriSign
21355 Ridgetop Circle
Dulles, VA 20166
US

Email: sleach@verisign.com

David
Akamai
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