dnsop
Internet-Draft
Intended status: Standards Track
Expires: March 21, 2018

J. Yao
P. Vixie
CNNIC-Farsight Joint Laboratory
N. Kong
X. Li
CNNIC

September 17, 2017

A DNS Query including A Main Question with Accompanying Questions
draft-yao-dnsop-accompanying-questions-04

Abstract

This document enables DNS initiators to send a main question accompanying with several related questions in a single DNS query, and enables DNS responders to put the answers into a single DNS response. This extension enables a range of initiators to look up "X, or failing that, Y" in a better way than both current alternatives. This mechanism can reduce the number of DNS round-trips per application work-unit.

Table of Contents

1. Introduction

Sometimes, when DNS lookup of X, an application will lookup Y if X fails. For examples, the initiator may fall back to A record if the lookup of MX record fails.

Some initiators do it in sequence, X and after a few seconds, then Y. Although it is simple, this leads to unpleasant waiting whenever X times out or answers negatively.

Some initiators use concurrent X/Y lookups and a state machine to decide whether to use X or Y. If an answer to Y arrives but none to X, the initiator needs to wait a little or else fall back to Y inappropriately. Concurrent lookup is faster if the X lookup takes time and falling back to Y is appropriate, but rather complex, with four states to test, and the initiator needs to wait for an answer to X or a timeout before it can use Y.

This document enables a quicker, more easily tested failover. There is no need to test different answer sequences, there's no need for a state machine, there's no need for timeouts beyond receiving the reply. This document describes a method by which DNS initiators can send a main question accompanying with several related questions in a single DNS query, and enables DNS responders place all related answers into a single DNS response. This mechanism can reduce the number of DNS round-trips per application work-unit, by carrying several related queries in a single query transaction. It has the following advantages compared to other solutions.

- Compared to sequential lookups: It's roughly as simple, but much faster in case a fallback to Y is necessary.

- Compared to the concurrent mechanism: It is slightly faster (if the initiator needs to wait for an X timeout) and/or prevents inappropriate fallback (if the answer to X arrives too late), and it has a simpler state machine.

This mechanism can also be used in the scenarios when the application needs more records of the same domain name or its sub-domain name. For examples, when asking about a QTYPE=A RRset, a QTYPE=AAAA RRset may also be of use [RFC 5321]; When asking for some RRset of www.example.com about A and AAAA, records of a sub-domain name such as _443._tcp.www.example.com for TLSA may be of interest[RFC 6698].

2. Terminology

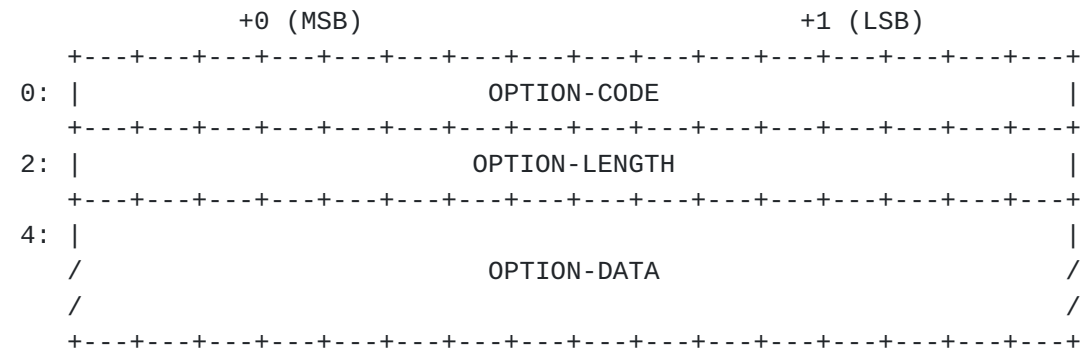
The basic key words such as "MUST", "MUST NOT", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "MAY", and "MAYNOT" are to be interpreted as described in [RFC2119].

The basic DNS terms used in this specification are defined in the documents [RFC1034] and [RFC1035].

3. Mechanism for a main question with accompanying questions

The initiator still puts a main question into the question section of the DNS query packet, as described in [RFC1035]. Accompanying questions will be put into the variable part of an OPT RR [RFC6891].

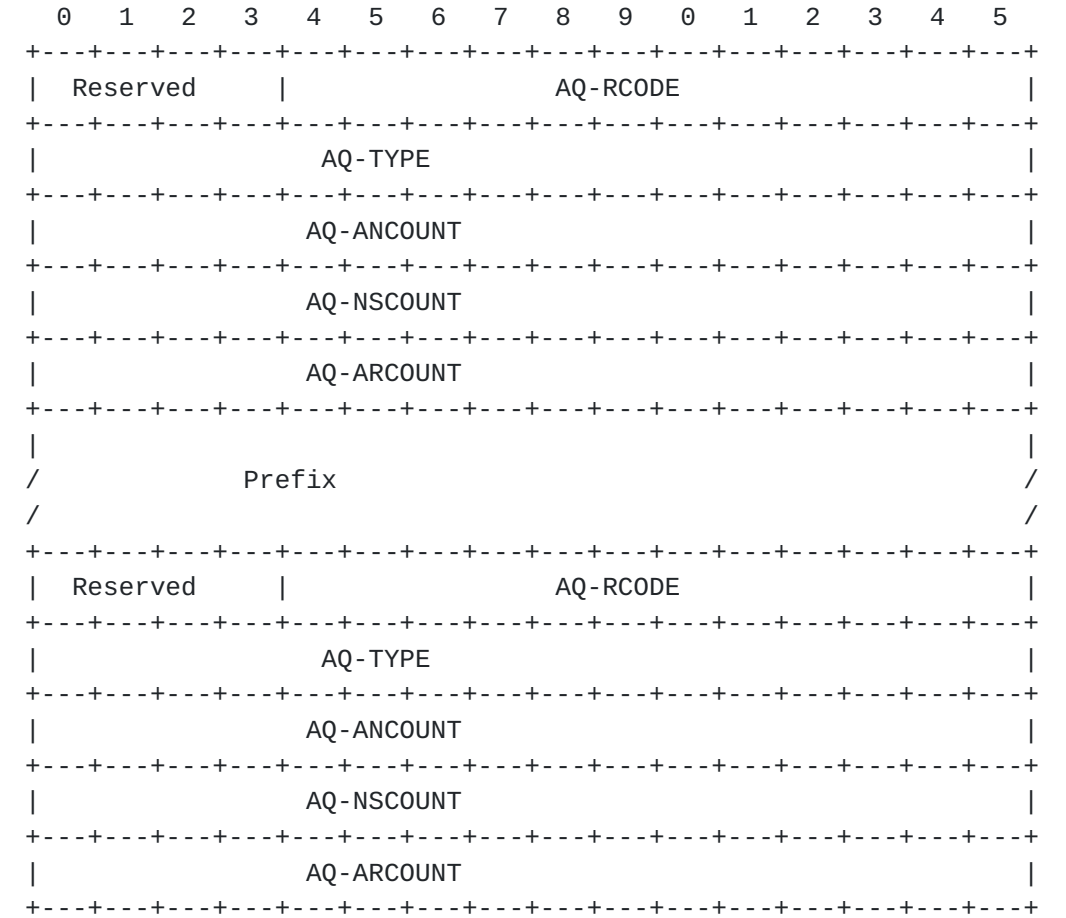
The variable part of an OPT RR is encoded in its RDATA and is structured as the following:

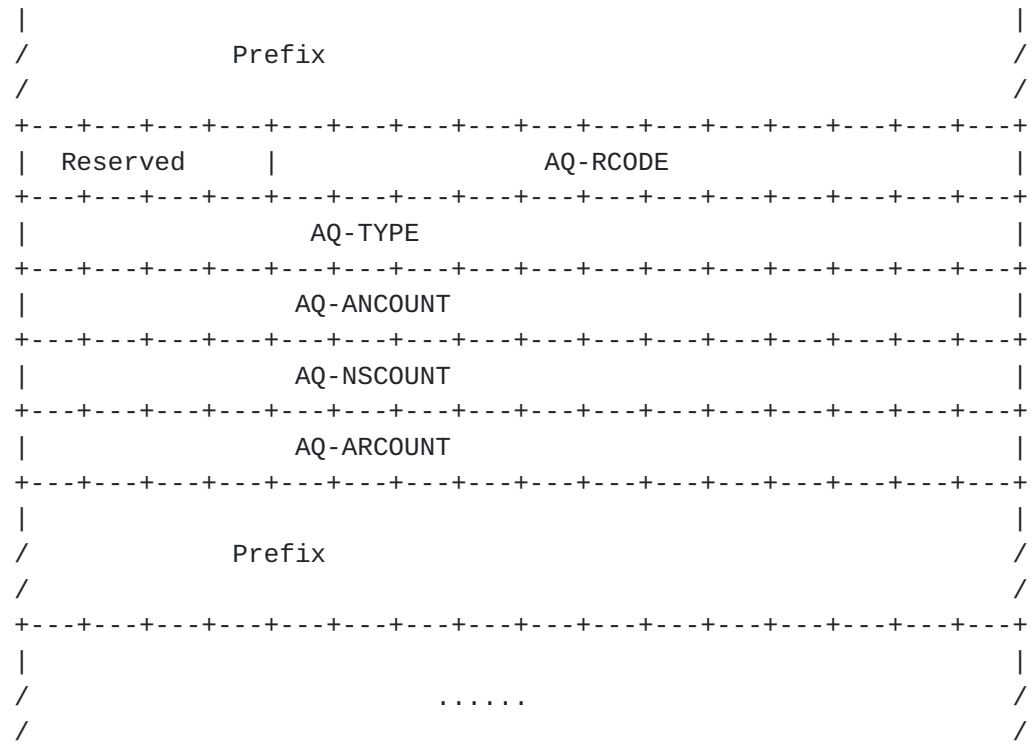


OPTION-CODE (Assigned by IANA.)

OPTION-LENGTH Size (in octets) of OPTION-DATA.

OPTION-DATA including at most 6 accompanying questions with AQ-RCODE.





- Reserved field is kept for the future use.

- AQ-RCODE field will be set to 111111110100 bits when being initialized. The AQ-RCODE with the value of 111111110100 bits means that the mechanism for accompanying has not been implemented, where "0100" in the RCODE value means "not been implemented". The AQ aware responders will put the RCODE value for the query of this question into AQ-RCODE fields.

- AQ-ANCOUNT field will indicate the number of resource records in the answer section for this accompanying question. The AQ aware responders will put the ANCOUNT value for the query of this question into AQ-ANCOUNT field.

- AQ-NSCOUNT field will indicate the number of name server resource records in the authority records section for this accompanying question. The AQ aware responders will put the NSCOUNT value for the query of this question into AQ-NSCOUNT field.

- AQ-ARCOUNT field will indicate the number of resource records in the additional records section for this accompanying question. The AQ aware responders will put the ARCOUNT value for the query of this question into AQ-ARCOUNT field.

- Prefix field indicates a domain name with the form of a dot or a sequence of labels ending with a pointer using the message compression defined in section 4.1.4. of RFC 1035. The domain name for accompanying questions MUST be same with the domain name for a main question or be children name of it. For an example, if the main domain name is example.com and the accompanying domain name is mail.example.com., the prefix is "mail." ending with a pointer pointing to "example.com.".

4. Responder Processing

The AQ aware responder will check the main question first, and put the results into the DNS response packet following RFC 1034. If the AQ OPT is present, the responder assembles the prefix with the main domain name and makes it to be an accompanying question, checks the accompanying questions in order, and put the results into the DNS answer section, authority section or additional records section of the response following RFC 1034; but the response code is placed in the respective AQ-RCODE field in AQ OPT of the response. The RCODE field in the DNS response header refers to the main question only. The AQ aware responders will put the ANCOUNT, NSCOUNT and ARCOUNT value for the query of this accompanying question into the respective AQ-ANCOUNT, AQ-NSCOUNT and AQ-ARCOUNT fields. The ANCOUNT, NSCOUNT and ARCOUNT fields in the DNS response header refer to the main question and its accompanying questions. Since the value for the accompanying questions' ANCOUNT, NSCOUNT and ARCOUNT can be known from the respective value of AQ-ANCOUNT, AQ-NSCOUNT and AQ-ARCOUNT, the actual value of the main question's ANCOUNT, NSCOUNT and ARCOUNT can be calculated from the ANCOUNT, NSCOUNT and ARCOUNT in the DNS response header. When the answer is negative for the accompanying question, the SOA resource record will be put in the authority section.

The mechanism proposed in this document is intended for both between stub resolvers and recursive resolvers, and between recursive resolvers and authoritative servers. If some DNS resource records are needed to be processed at the same time, the DNS administrator may configure it together. In case of that some children domain names are delegated and not in the main domain name's zone, the delegation information will be returned to the recursive resolvers. The recursive resolvers then check the children domain based on the delegation information, and get the answer for the respective children domain names.

When a stub resolver sends an AQ query to the recursive resolver, the recursive resolver may have some answers for one or more questions in the cache, but not for all questions. Under that case, the recursive resolver SHOULD forward this AQ query to some relative authoritative

servers for full answers instead of using the existing insufficient cache information.

An AQ unaware responder is expected to ignore the AQ OPT of the query, and may echo the received OPT back into additional section of the response message.

5. Initiator Processing

An AQ aware initiator will put the main question into the question section of the DNS query packet, and put each accompanying question into the related accompanying question fields of OPTION-DATA of OPT RR. AQ-RCODE value will be sent as 111111110100 bits. The AQ-TYPE value should be set as the query type related to accompanying questions. The Prefix value should be set as a dot or a sequence of labels ending with a pointer pointing to the the main domain name of the main question for the respective accompanying domain name of the accompanying question.

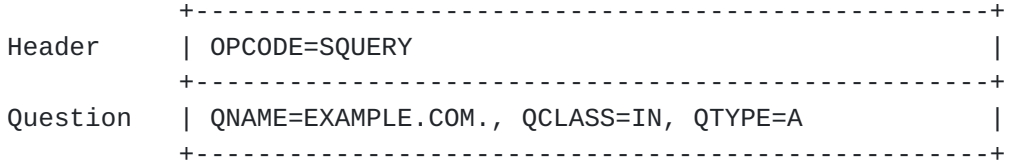
An AQ aware initiator SHOULD set the limitation of what is the maximum number of accompanying questions a AQ query can bring. This document suggests that the maximum number is six since most DNS resource records which need parallel query will not larger than six. The implementers may set six as the defaul value in the implementation. The responder can refuse to answer the AQ query if the maximum number of the accompanying questions is larger than the default maximum value, and return "not been implemented, too many accompanying-questions." information to the initiator.

If the initial value of the AQ-RCODE is unchanged in the response or the AQ OPT is not echo back, it indicates that the responder is AQ unaware. In that case, the responder will deal with the main question only. The initiator should sent the accompanying questions one by one via the normal DNS query. In such followup related queries, AQ processing should probably not be attempted, to reduce waste of network resources.

6. Query and Response Example

Example: one main question with 2 accompanying questions

The query would look like:



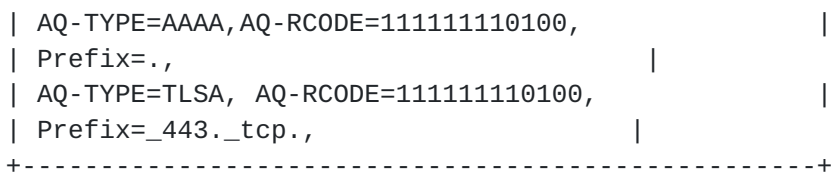
```
Answer     |                                                  |
           +--------------------------------------------------+
Authority  | <empty>                                          |
           +--------------------------------------------------+
Additional |                                                  |
           | AQ-TYPE=AAAA,AQ-RCODE=111111110100,              |
           | Prefix=.,                                  |
           | AQ-TYPE=TLSA,,AQ-RCODE=111111110100,             |
           | Prefix=_443._tcp.,                       |
           +--------------------------------------------------+
```

The response from AQ aware responders would be:

```
           +--------------------------------------------------+
Header     | OPCODE=SQUERY,  RESPONSE, AA, RCODE=NOERROR      |
           |      ANCOUNT=3, ARCOUNT=1, NSCOUNT=0             |
           +--------------------------------------------------+
Question   | QNAME=EXAMPLE.COM., QCLASS=IN, QTYPE=A           |
           +--------------------------------------------------+
Answer     |        example.com  IN A 192.168.0.1             |
           |        example.com. IN AAAA 2001:cc8::1          |
           |   _443._tcp.example.com. IN TLSA                 |
           |        ( 3 0 0 30820307308201efa003020102020... ) |
           +--------------------------------------------------+
Authority  | <empty>                                          |
           +--------------------------------------------------+
Additional |                                                  |
           | AQ-TYPE=AAAA, AQ-RCODE=NOERROR, AQ-ANCOUNT=1,    |
           |                   AQ-ARCOUNT=0, AQ-NSCOUNT=0,    |
           | Prefix=.,                                  |
           | AQ-TYPE=TLSA, AQ-RCODE=NOERROR, AQ-ANCOUNT=1,    |
           |                   AQ-ARCOUNT=0, AQ-NSCOUNT=0,    |
           | Prefix=_443._tcp.,                       |
           +--------------------------------------------------+
```

The response from AQ unaware responders would be:

```
           +--------------------------------------------------+
Header     | OPCODE=SQUERY,  RESPONSE, AA, RCODE=NOERROR      |
           +--------------------------------------------------+
Question   | QNAME=EXAMPLE.COM., QCLASS=IN, QTYPE=A           |
           +--------------------------------------------------+
Answer     |        example.com.  IN A 192.168.0.1            |
           +--------------------------------------------------+
Authority  | <empty>                                          |
           +--------------------------------------------------+
Additional |                                                  |
```



7. IANA Considerations

IANA should allocate DNS EDNS0 Option Codes (OPT) following this document. IANA should reserve RCODE with the value of 111111110100 bits for this document.

8. Security Considerations

TBD

9. Acknowledgements

The authors thank the members in DNSOP mailing list for helpful discussions, and especially thank Kazunori Fujiwara, JINMEI Tatuya, Bob Harold, Arnt Gulbrandsen, Olafur Gudmundsson and Stephane Bortzmeyer for kind comments, suggestions and improvements for the document. The authors also thanks Likun Zhang for helpful discussion about some topics related to implementation.

10. Change History

RFC Editor: Please remove this section.

10.1. draft-yao-dnsop-accompanying-questions: Version 00

- A Mechanism for DNS query including one main question with several accompanying questions

10.2. draft-yao-dnsop-accompanying-questions: Version 01

- Simpilfy the mechanism.

10.3. draft-yao-dnsop-accompanying-questions: Version 02

- Remove the AQ and Count bits, and add AQ-ANCOUNT AQ-ARCOUNT AQ-NSCOUNT

10.4. draft-yao-dnsop-accompanying-questions: Version 03

- Improve the introduction and explains the motivation of this draft

10.5. draft-yao-dnsop-accompanying-questions: Version 04

- Improve the document

11. Normative References

[RFC1034] Mockapetris, P., "Domain names - concepts and facilities", STD 13, RFC 1034, DOI 10.17487/RFC1034, November 1987, <https://www.rfc-editor.org/info/rfc1034>.

[RFC1035] Mockapetris, P., "Domain names - implementation and specification", STD 13, RFC 1035, DOI 10.17487/RFC1035, November 1987, <https://www.rfc-editor.org/info/rfc1035>.

[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>.

[RFC5321] Klensin, J., "Simple Mail Transfer Protocol", RFC 5321, DOI 10.17487/RFC5321, October 2008, <https://www.rfc-editor.org/info/rfc5321>.

[RFC6698] Hoffman, P. and J. Schlyter, "The DNS-Based Authentication of Named Entities (DANE) Transport Layer Security (TLS) Protocol: TLSA", RFC 6698, DOI 10.17487/RFC6698, August 2012, <https://www.rfc-editor.org/info/rfc6698>.

[RFC6891] Damas, J., Graff, M., and P. Vixie, "Extension Mechanisms for DNS (EDNS(0))", STD 75, RFC 6891, DOI 10.17487/RFC6891, April 2013, <https://www.rfc-editor.org/info/rfc6891>.

Authors' Addresses

Jiankang Yao
CNNIC-Farsight Joint Laboratory
4 South 4th Street,Zhongguancun,Haidian District
Beijing, Beijing 100190
China

Phone: +86 10 5881 3007
Email: yaojk@cnnic.cn

Paul Vixie
CNNIC-Farsight Joint Laboratory
4 South 4th Street,Zhongguancun,Haidian District
Beijing, Beijing 100190
China

Phone: +1 650 489 7919
Email: vixie@fsi.io

Ning Kong
CNNIC
4 South 4th Street,Zhongguancun,Haidian District
Beijing, Beijing 100190
China

Phone: +86 10 5881 3147
Email: nkong@cnnic.cn

Xiaodong Li
CNNIC
4 South 4th Street,Zhongguancun,Haidian District
Beijing, Beijing 100190
China

Phone: +86 10 5881 3020
Email: xl@cnnic.cn