Internet Engineering Task Force (IETF)
Request for Comments: 6891
STD: 75
Obsoletes: 2671, 2673
Category: Standards Track
ISSN: 2070-1721
April 2013

J. Damas
Bond Internet Systems
M. Graff

P. Vixie
Internet Systems Consortium

Extension Mechanisms for DNS (EDNS(0))

Abstract

The Domain Name System's wire protocol includes a number of fixed fields whose range has been or soon will be exhausted and does not allow requestors to advertise their capabilities to responders. This document describes backward-compatible mechanisms for allowing the protocol to grow.

This document updates the Extension Mechanisms for DNS (EDNS(0)) specification (and obsoletes RFC 2671) based on feedback from deployment experience in several implementations. It also obsoletes RFC 2673 ("Binary Labels in the Domain Name System") and adds considerations on the use of extended labels in the DNS.

Status of This Memo

This is an Internet Standards Track document.

This document is a product of the Internet Engineering Task Force (IETF). It represents the consensus of the IETF community. It has received public review and has been approved for publication by the Internet Engineering Steering Group (IESG). Further information on Internet Standards is available in Section 2 of RFC 5741.

Information about the current status of this document, any errata, and how to provide feedback on it may be obtained at http://www.rfc-editor.org/info/rfc6891.

Copyright Notice

Table of Contents

1. Introduction

DNS [RFC1035] specifies a message format, and within such messages there are standard formats for encoding options, errors, and name compression. The maximum allowable size of a DNS message over UDP not using the extensions described in this document is 512 bytes. Many of DNS’s protocol limits, such as the maximum message size over UDP, are too small to efficiently support the additional information that can be conveyed in the DNS (e.g., several IPv6 addresses or DNS Security (DNSSEC) signatures). Finally, RFC 1035 does not define any way for implementations to advertise their capabilities to any of the other actors they interact with.

[RFC2671] added extension mechanisms to DNS. These mechanisms are widely supported, and a number of new DNS uses and protocol extensions depend on the presence of these extensions. This memo refines and obsoletes [RFC2671].

Unextended agents will not know how to interpret the protocol extensions defined in [RFC2671] and restated here. Extended agents need to be prepared for handling the interactions with unextended clients in the face of new protocol elements and fall back gracefully to unextended DNS.

EDNS is a hop-by-hop extension to DNS. This means the use of EDNS is negotiated between each pair of hosts in a DNS resolution process, for instance, the stub resolver communicating with the recursive resolver or the recursive resolver communicating with an authoritative server.

[RFC2671] specified extended label types. The only such label proposed was in [RFC2673] for a label type called "Bit-String Label" or "Binary Labels", with this latest term being the one in common use. For various reasons, introducing a new label type was found to be extremely difficult, and [RFC2673] was moved to Experimental. This document obsoletes [RFC2673], deprecating Binary Labels. Extended labels remain defined, but their use is discouraged due to practical difficulties with deployment; their use in the future SHOULD only be considered after careful evaluation of the deployment hindrances.

2. Terminology

"Requestor" refers to the side that sends a request. "Responder" refers to an authoritative, recursive resolver or other DNS component that responds to questions. Other terminology is used here as defined in the RFCs cited by this document.

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 RFC 2119 [RFC2119].

3. EDNS Support Requirement

EDNS provides a mechanism to improve the scalability of DNS as its uses get more diverse on the Internet. It does this by enabling the use of UDP transport for DNS messages with sizes beyond the limits specified in RFC 1035 as well as providing extra data space for additional flags and return codes (RCODEs). However, implementation experience indicates that adding new RCODEs should be avoided due to the difficulty in upgrading the installed base. Flags SHOULD be used only when necessary for DNS resolution to function.

For many uses, an EDNS Option Code may be preferred.

Over time, some applications of DNS have made EDNS a requirement for their deployment. For instance, DNSSEC uses the additional flag space introduced in EDNS to signal the request to include DNSSEC data in a DNS response.

Given the increase in DNS response sizes when including larger data items such as AAAA records, DNSSEC information (e.g., RRSIG or DNSKEY), or large TXT records, the additional UDP payload capabilities provided by EDNS can help improve the scalability of the DNS by avoiding widespread use of TCP for DNS transport.

4. DNS Message Changes

4.1. Message Header

The DNS message header’s second full 16-bit word is divided into a 4-bit OPCODE, a 4-bit RCODE, and a number of 1-bit flags (see Section 4.1.1 of [RFC1035]). Some of these flag values were marked for future use, and most of these have since been allocated. Also, most of the RCODE values are now in use. The OPT pseudo-RR specified below contains extensions to the RCODE bit field as well as additional flag bits.

4.2. Label Types

The first 2 bits of a wire format domain label are used to denote the type of the label. [RFC1035] allocates 2 of the 4 possible types and reserves the other 2. More label types were defined in [RFC2671]. The use of the 2-bit combination defined by [RFC2671] to identify extended label types remains valid. However, it has been found that deployment of new label types is noticeably difficult and so is only recommended after careful evaluation of alternatives and the need for deployment.

4.3. UDP Message Size

Traditional DNS messages are limited to 512 octets in size when sent over UDP [RFC1035]. Fitting the increasing amounts of data that can be transported in DNS in this 512-byte limit is becoming more difficult. For instance, inclusion of DNSSEC records frequently requires a much larger response than a 512-byte message can hold.

EDNS(0) specifies a way to advertise additional features such as larger response size capability, which is intended to help avoid truncated UDP responses, which in turn cause retry over TCP. It therefore provides support for transporting these larger packet sizes without needing to resort to TCP for transport.

5. Extended Label Types

The first octet in the on-the-wire representation of a DNS label specifies the label type; the basic DNS specification [RFC1035] dedicates the 2 most significant bits of that octet for this purpose.

[RFC2671] defined DNS label type 0b01 for use as an indication for extended label types. A specific extended label type was selected by the 6 least significant bits of the first octet. Thus, extended label types were indicated by the values 64-127 (0b01xxxxxx) in the first octet of the label.

Extended label types are extremely difficult to deploy due to lack of support in clients and intermediate gateways, as described in [RFC3363], which moved [RFC2673] to Experimental status; and [RFC3364], which describes the pros and cons. As such, proposals that contemplate extended labels SHOULD weigh this deployment cost against the possibility of implementing functionality in other ways.

Finally, implementations MUST NOT generate or pass Binary Labels in their communications, as they are now deprecated.

6. The OPT Pseudo-RR

6.1. OPT Record Definition

6.1.1. Basic Elements

An OPT pseudo-RR (sometimes called a meta-RR) MAY be added to the additional data section of a request.

The OPT RR has RR type 41.

If an OPT record is present in a received request, compliant responders MUST include an OPT record in their respective responses.

An OPT record does not carry any DNS data. It is used only to contain control information pertaining to the question-and-answer sequence of a specific transaction. OPT RRs MUST NOT be cached, forwarded, or stored in or loaded from master files.

The OPT RR MAY be placed anywhere within the additional data section. When an OPT RR is included within any DNS message, it MUST be the only OPT RR in that message. If a query message with more than one OPT RR is received, a FORMERR (RCODE=1) MUST be returned. The placement flexibility for the OPT RR does not override the need for the TSIG or SIG(0) RRs to be the last in the additional section whenever they are present.

6.1.2. Wire Format

An OPT RR has a fixed part and a variable set of options expressed as {attribute, value} pairs. The fixed part holds some DNS metadata, and also a small collection of basic extension elements that we expect to be so popular that it would be a waste of wire space to encode them as {attribute, value} pairs.

The fixed part of an OPT RR is structured as follows:

Field Name	Field Type	Description
NAME	domain name	MUST be 0 (root domain)
TYPE	u_int16_t	OPT (41)
CLASS	u_int16_t	requestor’s UDP payload size
TTL	u_int32_t	extended RCODE and flags
RDLEN	u_int16_t	length of all RDATA
RDATA	octet stream	{attribute,value} pairs

OPT RR Format

The variable part of an OPT RR may contain zero or more options in the RDATA. Each option MUST be treated as a bit field. Each option is encoded as:

```
                +0 (MSB)                            +1 (LSB)
     +---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
  0: |                          OPTION-CODE                          |
     +---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
  2: |                         OPTION-LENGTH                         |
     +---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
  4: |                                                               |
     /                          OPTION-DATA                          /
     /                                                               /
     +---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
```

OPTION-CODE
   Assigned by the Expert Review process as defined by the DNSEXT working group and the IESG.

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

OPTION-DATA
   Varies per OPTION-CODE. MUST be treated as a bit field.

The order of appearance of option tuples is not defined. If one option modifies the behaviour of another or multiple options are related to one another in some way, they have the same effect regardless of ordering in the RDATA wire encoding.

Any OPTION-CODE values not understood by a responder or requestor MUST be ignored. Specifications of such options might wish to include some kind of signaled acknowledgement. For example, an option specification might say that if a responder sees and supports option XYZ, it MUST include option XYZ in its response.

6.1.3. OPT Record TTL Field Use

The extended RCODE and flags, which OPT stores in the RR Time to Live (TTL) field, are structured as follows:

```
                 +0 (MSB)                            +1 (LSB)
      +---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
   0: |         EXTENDED-RCODE        |            VERSION            |
      +---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
   2: | DO|                           Z                               |
      +---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
```

EXTENDED-RCODE

Forms the upper 8 bits of extended 12-bit RCODE (together with the 4 bits defined in [RFC1035]. Note that EXTENDED-RCODE value 0 indicates that an unextended RCODE is in use (values 0 through 15).

VERSION

Indicates the implementation level of the setter. Full conformance with this specification is indicated by version ’0’. Requestors are encouraged to set this to the lowest implemented level capable of expressing a transaction, to minimise the responder and network load of discovering the greatest common implementation level between requestor and responder. A requestor’s version numbering strategy MAY ideally be a run-time configuration option.

If a responder does not implement the VERSION level of the request, then it MUST respond with RCODE=BADVERS. All responses MUST be limited in format to the VERSION level of the request, but the VERSION of each response SHOULD be the highest implementation level of the responder. In this way, a requestor will learn the implementation level of a responder as a side effect of every response, including error responses and including RCODE=BADVERS.

6.1.4. Flags

DO

DNSSEC OK bit as defined by [RFC3225].

Z

Set to zero by senders and ignored by receivers, unless modified in a subsequent specification.

6.2. Behaviour

6.2.1. Cache Behaviour

The OPT record MUST NOT be cached.

6.2.2. Fallback

If a requestor detects that the remote end does not support EDNS(0), it MAY issue queries without an OPT record. It MAY cache this knowledge for a brief time in order to avoid fallback delays in the future. However, if DNSSEC or any future option using EDNS is required, no fallback should be performed, as these options are only signaled through EDNS. If an implementation detects that some servers for the zone support EDNS(0) while others would force the use of TCP to fetch all data, preference MAY be given to servers that support EDNS(0). Implementers SHOULD analyse this choice and the impact on both endpoints.

6.2.3. Requestor’s Payload Size

The requestor’s UDP payload size (encoded in the RR CLASS field) is the number of octets of the largest UDP payload that can be reassembled and delivered in the requestor’s network stack. Note that path MTU, with or without fragmentation, could be smaller than this.

Values lower than 512 MUST be treated as equal to 512.

The requestor SHOULD place a value in this field that it can actually receive. For example, if a requestor sits behind a firewall that will block fragmented IP packets, a requestor SHOULD NOT choose a value that will cause fragmentation. Doing so will prevent large responses from being received and can cause fallback to occur. This knowledge may be auto-detected by the implementation or provided by a human administrator.

Note that a 512-octet UDP payload requires a 576-octet IP reassembly buffer. Choosing between 1280 and 1410 bytes for IP (v4 or v6) over Ethernet would be reasonable.

Where fragmentation is not a concern, use of bigger values SHOULD be considered by implementers. Implementations SHOULD use their largest configured or implemented values as a starting point in an EDNS transaction in the absence of previous knowledge about the destination server.

Choosing a very large value will guarantee fragmentation at the IP layer, and may prevent answers from being received due to loss of a single fragment or to misconfigured firewalls.

The requestor’s maximum payload size can change over time. It MUST NOT be cached for use beyond the transaction in which it is advertised.

6.2.4. Responder’s Payload Size

The responder’s maximum payload size can change over time but can reasonably be expected to remain constant between two closely spaced sequential transactions, for example, an arbitrary QUERY used as a probe to discover a responder’s maximum UDP payload size, followed immediately by an UPDATE that takes advantage of this size. This is considered preferable to the outright use of TCP for oversized requests, if there is any reason to suspect that the responder implements EDNS, and if a request will not fit in the default 512-byte payload size limit.

6.2.5. Payload Size Selection

Due to transaction overhead, it is not recommended to advertise an architectural limit as a maximum UDP payload size. Even on system stacks capable of reassembling 64 KB datagrams, memory usage at low levels in the system will be a concern. A good compromise may be the use of an EDNS maximum payload size of 4096 octets as a starting point.

A requestor MAY choose to implement a fallback to smaller advertised sizes to work around firewall or other network limitations. A requestor SHOULD choose to use a fallback mechanism that begins with a large size, such as 4096. If that fails, a fallback around the range of 1280-1410 bytes SHOULD be tried, as it has a reasonable chance to fit within a single Ethernet frame. Failing that, a requestor MAY choose a 512-byte packet, which with large answers may cause a TCP retry.

Values of less than 512 bytes MUST be treated as equal to 512 bytes.

6.2.6. Support in Middleboxes

In a network that carries DNS traffic, there could be active equipment other than that participating directly in the DNS resolution process (stub and caching resolvers, authoritative servers) that affects the transmission of DNS messages (e.g., firewalls, load balancers, proxies, etc.), referred to here as "middleboxes".

Conformant middleboxes MUST NOT limit DNS messages over UDP to 512 bytes.

Middleboxes that simply forward requests to a recursive resolver MUST NOT modify and MUST NOT delete the OPT record contents in either direction.

Middleboxes that have additional functionality, such as answering queries or acting as intelligent forwarders, SHOULD be able to process the OPT record and act based on its contents. These middleboxes MUST consider the incoming request and any outgoing requests as separate transactions if the characteristics of the messages are different.

A more in-depth discussion of this type of equipment and other considerations regarding their interaction with DNS traffic is found in [RFC5625].

7. Transport Considerations

The presence of an OPT pseudo-RR in a request should be taken as an indication that the requestor fully implements the given version of EDNS and can correctly understand any response that conforms to that feature’s specification.

Lack of presence of an OPT record in a request MUST be taken as an indication that the requestor does not implement any part of this specification and that the responder MUST NOT include an OPT record in its response.

Extended agents MUST be prepared for handling interactions with unextended clients in the face of new protocol elements and fall back gracefully to unextended DNS when needed.

Responders that choose not to implement the protocol extensions defined in this document MUST respond with a return code (RCODE) of FORMERR to messages containing an OPT record in the additional section and MUST NOT include an OPT record in the response.

If there is a problem with processing the OPT record itself, such as an option value that is badly formatted or that includes out-of-range values, a FORMERR MUST be returned. If this occurs, the response MUST include an OPT record. This is intended to allow the requestor to distinguish between servers that do not implement EDNS and format errors within EDNS.

The minimal response MUST be the DNS header, question section, and an OPT record. This MUST also occur when a truncated response (using the DNS header's TC bit) is returned.

8. Security Considerations

Requestor-side specification of the maximum buffer size may open a DNS denial-of-service attack if responders can be made to send messages that are too large for intermediate gateways to forward, thus leading to potential ICMP storms between gateways and responders.

Announcing very large UDP buffer sizes may result in dropping of DNS messages by middleboxes (see Section 6.2.6). This could cause retransmissions with no hope of success. Some devices have been found to reject fragmented UDP packets.

Announcing UDP buffer sizes that are too small may result in fallback to TCP with a corresponding load impact on DNS servers. This is especially important with DNSSEC, where answers are much larger.

9. IANA Considerations

The IANA has assigned RR type code 41 for OPT.

[RFC2671] specified a number of IANA subregistries within "DOMAIN NAME SYSTEM PARAMETERS":

- DNS EDNS(0) Options
- EDNS Version Number
- EDNS Header Flags

Additionally, two entries were generated in existing registries:

- EDNS Extended Label Type in the DNS Label Types registry
- Bad OPT Version in the DNS RCODES registry

IANA has updated references to [RFC2671] in these entries and subregistries to this document.

[RFC2671] created the DNS Label Types registry. This registry is to remain open.

The registration procedure for the DNS Label Types registry is Standards Action.

This document assigns option code 65535 in the DNS EDNS0 Options registry to "Reserved for future expansion".

The current status of the IANA registry for EDNS Option Codes at the time of publication of this document is

- 0-4 assigned, per references in the registry

- 5-65000 Available for assignment, unassigned

- 65001-65534 Local/Experimental use

- 65535 Reserved for future expansion

[RFC2671] expands the RCODE space from 4 bits to 12 bits. This allows more than the 16 distinct RCODE values allowed in [RFC1035]. IETF Review is required to add a new RCODE.

This document assigns EDNS Extended RCODE 16 to "BADVERS" in the DNS RCODES registry.

[RFC2671] called for the recording of assignment of extended label types 0bxx111111 as "Reserved for future extended label types"; the IANA registry currently contains "Reserved for future expansion". This request implied, at that time, a request to open a new registry for extended label types, but due to the possibility of ambiguity, new text registrations were instead made within the general DNS Label Types registry, which also registers entries originally defined by [RFC1035]. There is therefore no Extended Label Types registry, with all label types registered in the DNS Label Types registry.

This document deprecates Binary Labels. Therefore, the status for the DNS Label Types registration "Binary Labels" is now "Historic".

IETF Standards Action is required for assignments of new EDNS(0) flags. Flags SHOULD be used only when necessary for DNS resolution to function. For many uses, an EDNS Option Code may be preferred.

IETF Standards Action is required to create new entries in the EDNS Version Number registry. Within the EDNS Option Code space, Expert Review is required for allocation of an EDNS Option Code. Per this document, IANA maintains a registry for the EDNS Option Code space.

9.1. OPT Option Code Allocation Procedure

OPT Option Codes are assigned by Expert Review.

Assignment of Option Codes should be liberal, but duplicate functionality is to be avoided.

10. References

10.1. Normative References

[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.

[RFC2671] Vixie, P., "Extension Mechanisms for DNS (EDNS0)", RFC 2671, August 1999.

[RFC3225] Conrad, D., "Indicating Resolver Support of DNSSEC", RFC 3225, December 2001.

10.2. Informative References

[RFC2673] Crawford, M., "Binary Labels in the Domain Name System", RFC 2673, August 1999.

[RFC3363] Bush, R., Durand, A., Fink, B., Gudmundsson, O., and T. Hain, "Representing Internet Protocol version 6 (IPv6) Addresses in the Domain Name System (DNS)", RFC 3363, August 2002.

[RFC3364] Austein, R., "Tradeoffs in Domain Name System (DNS) Support for Internet Protocol version 6 (IPv6)", RFC 3364, August 2002.

[RFC5625] Bellis, R., "DNS Proxy Implementation Guidelines", BCP 152, RFC 5625, August 2009.

Appendix A. Changes since RFCs 2671 and 2673

Following is a list of high-level changes to RFCs 2671 and 2673.

- Support for the OPT record is now mandatory.
- Extended label types remain available, but their use is discouraged as a general solution due to observed difficulties in their deployment on the Internet, as illustrated by the work with the "Binary Labels" type.
- RFC 2673, which defined the "Binary Labels" type and is currently Experimental, is requested to be moved to Historic.
- Made changes in how EDNS buffer sizes are selected, and provided recommendations on how to select them.

Authors' Addresses

Joao Damas
Bond Internet Systems
Av Albufera 14
S.S. Reyes, Madrid 28701
ES

Phone: +1 650.423.1312
EMail: joao@bondis.org

Michael Graff

EMail: explorer@flame.org

Paul Vixie
Internet Systems Consortium
950 Charter Street
Redwood City, California 94063
US

Phone: +1 650.423.1301
EMail: vixie@isc.org