DNSIND Working Group
INTERNET-DRAFT
<draft-ietf-dnsind-edns0-00.txt>

Paul Vixie
ISC
September, 1998

Extension mechanisms for DNS (EDNS0)

Status of this Memo

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 clients to advertise their capabilities to servers. This document describes backward compatible mechanisms for allowing the protocol to grow.

1 - Rationale and Scope

1.1. DNS (see [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 is fixed. Many of DNS's protocol limits are too small for uses which are or which are desired to become common. There is no way for clients to advertise their capabilities to servers.

1.2. Existing clients will not know how to interpret the protocol extensions detailed here. In practice, these clients will be upgraded when they have need of a new feature, and only new features will make use of the extensions. We must however take account of client behaviour in the face of extra fields, and design a fallback scheme for interoperability with these clients.

2 - Affected Protocol Elements

2.1. The DNS Message Header's (see [RFC1035 4.1.1]) second full 16-bit word is divided into a 4-bit OPCODE, a 4-bit RCODE, and a number of 1-bit flags. The original reserved Z bits have been allocated to various purposes, and most of the RCODE values are now in use. More types and more possible RCODEs are needed.

2.2. The first two bits of a wire format domain label are used to denote the type of the label. [RFC1035 4.1.4] allocates two of the four possible types and reserves the other two. Proposals for use of the remaining types far outnumber those available. More label types are needed.

2.3. DNS Messages are limited to 512 octets in size when sent over UDP. While the minimum maximum reassembly buffer size is still 512 bytes, most of the hosts now connected to the Internet are able to reassemble larger datagrams. Some mechanism must be created to allow requestors to advertise larger buffer sizes to responders.

3 - Extended Label Types

3.1. The ``0 1'' label type will now indicate an extended label type, whose value is encoded in the lower six bits of the first octet of a label. All subsequently developed label types should be encoded using an extended label type.

3.2. The ``1 1 1 1 1 1'' extended label type will be reserved for future expansion of the extended label type code space.

4 - OPT pseudo-RR

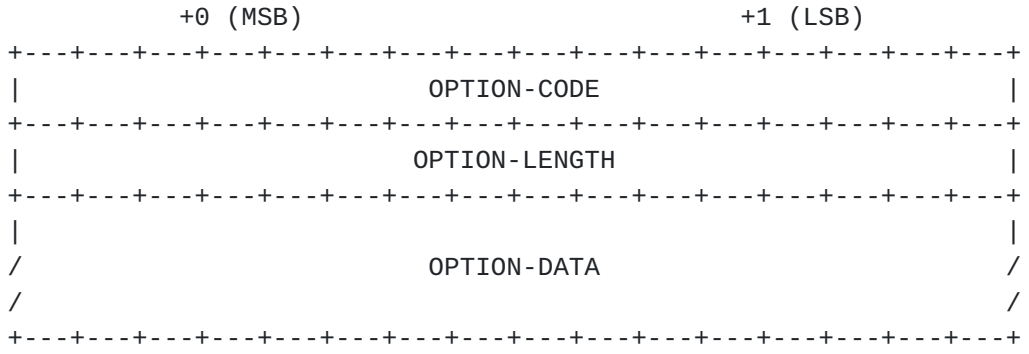
4.1. The OPT pseudo-RR can be added to the additional data section of either a request or a response. An OPT is called a pseudo-RR because it pertains to a particular transport level message and not to any actual DNS data. OPT RRs shall never be cached, forwarded, or stored in or loaded from master files.

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

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

Field Name	Field Type	Description
NAME	domain name	empty (root domain)
TYPE	u_int16_t	OPT (XXX IANA)
CLASS	u_int16_t	sender's UDP buffer size
TTL	u_int32_t	extended RCODE and flags
RDLEN	u_int16_t	describes RDATA
RDATA	octet stream	{attribute,value} pairs

4.4. The variable part of an OPT RR is encoded in its RDATA and is structured as zero or more of the following:



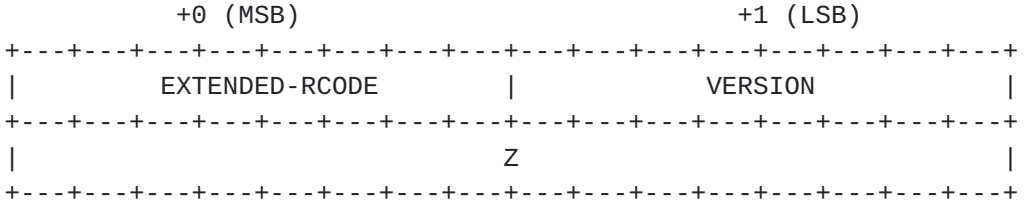
OPTION-CODE	Assigned by the IANA. Value 65535 is reserved for future expansion.
OPTION-LENGTH	Size (in octets) of OPTION-DATA.
OPTION-DATA	Varies per OPTION-CODE.

4.5. The sender's UDP buffer size is the number of octets of the largest UDP payload that can be reassembled and delivered in the sender's network stack. Note that path MTU, with or without fragmentation, may be smaller than this.

4.5.1. Note that a 512-octet UDP payload requires a 576-octet IP reassembly buffer. Choosing 1280 on an Ethernet connected requestor would be reasonable. The consequence of choosing too large a value may be an ICMP message from an intermediate gateway, or even a silent drop of the response message. Requestors are advised to retry in such cases.

4.5.2. Both requestors and responders are advised to take account of the path's already discovered MTU (if known) when considering message sizes.

4.6. The extended RCODE and flags are structured as follows:



EXTENDED-RCODE Forms upper 8 bits of extended 12-bit RCODE.

VERSION Indicates the implementation level of whoever sets it. Full conformance with the draft standard version of this specification is version ``0.'' Note that both requestors and responders should set this to the highest level they implement, that responders should send back RCODE=BADVERS (XXX IANA) and that requestors should be prepared to probe using lower version numbers if they receive an RCODE=BADVERS.

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

5 - Transport Considerations

5.1. 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.

5.2. Lack of use of these features in a request must be taken as an indication that the requestor does not implement any part of this specification and that the responder may make no use of any protocol extension described here in its response.

5.3. Responders who do not understand these protocol extensions are expected to send a respose with RCODE NOTIMPL, FORMERR, or SERVFAIL. Therefore use of extensions should be ``probed'' such that a responder who isn't known to support them be allowed a retry with no extensions if it responds with such an RCODE. If a responder's capability level is cached by a requestor, a new probe should be sent periodically to test for changes to responder capability.

6 - Security Considerations

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

7 - Acknowledgements

Paul Mockapetris, Mark Andrews, Robert Elz, Don Lewis, Bob Halley, Donald Eastlake, Rob Austein, Matt Crawford, and Randy Bush were each instrumental in creating this specification.

8 - References

[RFC1035] P. Mockapetris, ``Domain Names - Implementation and Specification,'' RFC 1035, USC/Information Sciences Institute, November 1987.

9 - Author's Address

Paul Vixie
Internet Software Consortium
950 Charter Street
Redwood City, CA 94063
+1 650 779 7001
<paul@vix.com>