Updates: RFC 1034

Negative Caching of DNS Queries (DNS NCACHE)

Status of This Memo

Abstract

When [RFC1034] was written there were no DNS servers that implemented negative caching [RFC1034 Section 4.3.4]. This document replaces [RFC1034 Section 4.3.4] in the light of experience.

Negative caching was a optional part of the DNS specification and deals with the caching of the non-existence of a RRset or domain name.

Negative caching is useful as it reduces the response time for negative answers. It also reduces the number of messages that have to be sent between servers hence overall network traffic. A large proportion of DNS traffic on the Internet could be eliminated if all servers implemented negative caching. With this in mind negative caching should no longer be seen as a optional

part of a DNS server.

0 - History

[Anyone want to say why negative caching wasn't implemented in the early servers or why not in BIND? If I don't get anything this section will go. MPA]

1 - Terminology

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 [RFCXXXX].

Negative caching is the storage of knowledge that something does not exist. For example we can storage the knowledge that a record has a particular value. We can also do the reverse, that is, to store the knowledge that a record does not exist. It is the storage of knowledge that something does not exist, cannot or does not give an answer that we call negative caching.

"QNAME" this refers to the name in the query section of an answer or where this resolves to a CNAME, or CNAME chain, the data field of the last CNAME. CNAMEs are expected to appear in order so that a single pass of the message will expose the chain before any requested records. Note SIG records may be intermixed with CNAME records.

"NODATA" is a pseudo RCODE which indicated that the name is valid, for the given class, but there was records of the given type. A NODATA response has to be inferred from the answer.

An understanding of [RFC1034], [RFC1035] and [RFC2065] is expected when reading this document.

2 - Negative Responses

Negative responses usually refer to the existence or lack there of a particular RRset in the DNS. The first sections of this document deal with these responses. There are other negative responses that indicate failures of servers. They are dealt with from section ``7 - Other negative responses'' onwards.

A negative response is indicated by one of the following conditions:

2.1 - Name Error

Name errors (NXDOMAIN) are indicated by the presence of "Name Error" in the RCODE field. In this case the domain referred to in the QNAME does not exist. Note answer section may have SIG and CNAME RRs and authority section may have SOA, NXT and SIG RRsets.

Some nameservers do not look at the RCODE and continue processing in the presence of NS records in the authority section. NS records MUST NOT be sent in the authority section where the RCODE is NXDOMAIN.

Some nameservers incorrectly continue processing if the authority flag is not set. The authority flag SHOULD be set even in non-authoritative answers. It is hoped that this restriction can be relaxed at some future time. With this in mind all servers MUST accept and correctly process messages with the authority flag not set.

It is possible to distinguish between a referral and a NXDOMAIN response by the presense of NXDOMAIN in the RCODE regardless of the presence of NS records in the authority section.

2.1 - No Data

NODATA responses have to be algorithmically determined from the responses contents as there is no RCODE field to indicate NODATA. In some cases it is necessary to query again to get a definitive answer.

No data is indicated by an answer with a RCODE of NOERROR, no relevant answers in the answer section and no NS records in the authority section. The authority section may contain SOA, NXT and SIG RRsets. CNAMEs and SIG records may exist in the answer section.

It is possible to distiguish between a referral and a NODATA response by the presence of a SOA record in the authority section or the absence of NS records in the authority section.

Some nameservers fail to stop processing if there is a SOA record in the authority section along with NS record. To prevent interoperability problems NS records SHOULD NOT be added to the authority section of a NODATA response. Nameervers MUST ensure that they stop processing in the presence of the SOA record in the authority section so that this restriction can be lifted at a further date.

Some nameservers fail to set the RCODE to NXDOMAIN in the presence of CNAMEs in the answer section. If a definitive NXDOMAIN / NODATA answer is required the server must query again with QNAME.

3 - Negative Answers from Authoritative Servers

Authoritative servers SHOULD add the SOA record of the containing zone to the authority section of answers containing negative responses to enable the response to be cached. The TTL of this record is set from the MINIMUM field of the SOA record and is not the TTL of the SOA itself and indicates how long a server may cache this negative answer.

If the containing zone is signed [RFC2065] the SOA and appropriate NXT and SIG records MUST be added.

4 - Caching Negative Answers

Like normal answers negative answers have a time to live (TTL). As there is no record in the answer section to which this TTL can be applied, the TTL must be carried by another method. This is done by using the SOA record from the containing zone and putting it in the authority section with an initial TTL set from the SOA minimum field. This TTL decrements in a similar manner to a normal cached answer [RFC1034 Section ...]. When the TTL reaches zero (0) the record MUST be discarded.

Often SOA minimums are set with no regard to the TTL of negative answers so that negative responses have TTL measured in days. Early advertise-ment of a service before all the secondaries have a copy of the relevant zone can lead to prolonged denials of service. With this in mind a server SHOULD set an upper bound on the TTL of the negative answer it is willing to cache. If it does this it the TTL MUST be set to the minimum of the server threshold and the received TTL. A server threshold of one (1) hour is often appropriate.

As this TTL is different to the TTL value of the SOA record itself, this SOA record MUST NOT be used to answer SOA queries.

A negative answer that resulted from a name error (NXDOMAIN) should be cached such that it can be retrieved and returned in response to another query for the same <QNAME, QCLASS> that resulted in the cached negative response.

A negative answer that resulted from a no data error (NODATA) should be cached such that it can be retrieved and returned in response to another query for the same <QNAME, QTYPE, QCLASS> that resulted in the cached negative response.

The NXT record, if it exists, MUST be stored such that it can be be located as should a SIG record. For NXDOMAIN answers there is no "necessary" obvious relationship between the NXT records and the query name. The NXT record MUST have the same owner name as the query name for NODATA responses.

Negative responses without SOA records SHOULD NOT be cached as there is no way to prevent the negative responses looping forever between a pair of servers even with a short TTL.

5 - Negative answers from the cache

Whan a server, in answering a query, encounters a cached negative response it MUST add the cached SOA to the authority section of the response.

If a NXT record was cached along with SOA record it MUST be added to the authority section. If a SIG record was cached along with a NXT record it SHOULD be added to the authority section.

NS records MUST NOT be added to the authority section as existing servers do not look for the SOA record that would indicate the difference between a NODATA response and a referal.

6 - Changes from RFC 1034

Negative caching in servers is no-longer optional.

Non-authoritative negative answers MAY be cached.

The SOA record from the authority section MUST be cached. Name error indications MUST be cached against the tuple <query name, QCLASS>. No data indications MUST be cached against <query name, QTYPE, QCLASS> tuple.

A cached SOA record MUST be added to the response. This was explicitly not allowed.

7 - Other Negative Responses

Caching of other negative responses is not covered by any existing RFC. There is no way to indicated a desired TTL of these responses. Care needs to be taken to ensure that there are not forwarding loops. [Do we need to have a hold down period where we cannot cache these, transport layer indications aside? MPA]

7.1 Server Failure (OPTIONAL)

Server failures fall into two major classes. The first is where a server can determine that it has been misconfigured for a zone. This may be where it has been listed as a server, but not configured to be a server for the zone, or where it has been configured to be a server server for the zone, but cannot obtain the zone data for some reason, either because the zone file does not exist or contains errors, or because another server from which the zone should have been available either did not respond or was unable or unwilling to supply the zone.

The second class is where the server needs to obtain an answer from elsewhere, but is unable to do so, due to network failures, other servers that don't reply, or return server failure errors, or similar.

A server MAY cache a server failure response. If it does so it MUST NOT cache it for longer that five (5) minutes, and it MUST be cached against the specific query tuple <query name, type, class, server IP address>.

7.2 Dead Server (OPTIONAL)

Dead servers are servers that fails to respond in any way to a query or the transport layer has provided an indication that the server does not exist. A server is deemed to be dead if it has not responded to an out-standing query within 120 seconds.

Examples of transport layer indications are:

```
        ICMP error messages
        TCP resets
        Kernel error messages indicating host or net unreachable.
```

A server MAY cache a dead server indication. If it does so it MUST NOT be deemed dead for longer than five (5) minutes. The indication MUST be stored against query tuple <query name, type, class, server IP address> unless there was a transport layer indication that the server does not exist, in which case it is stored against the specific IP address involved.

Security

It is believed that this document does not introduce and significant additional security threats.

With negative caching it might be possible to propagate a denial of service attack by spreading a NXDOMAIN message with a very high TTL. Without negative caching that would be much harder. A similar effect could be achieved previously by spreading a bad A record, so that the server could not be reached - which is almost the same but not quite. It has the same effect as far as what the end user is able to do, but with a different psychological effect. With the bad A, I feel "damn the network is broken again" and try again tomorrow. With the "NXDOMAIN" I feel "Oh, they've turned off the server and it doesn't exist any more" and probably never bother trying this server again.

For such an attack to be sucessful you need to get the NXDOMAIN indiction injected into a parent server (or a busy caching server). This can only be done by the use of a CNAME which results in the parent server quering an attackers server. Servers that are wish to prevent such attacks can query again the final QNAME ignoring any NS data in the query responses it has received for this query.

Impementing TTL sanity check will reduce the impact of such an attack and cause it to be come an active attack rather than a passive attack, i.e. it needs to be reprimed regularly.

DNS Security [RFC2065] provides a mechanism to verify whether a negative response is valid or not, through the use of NXT and SIG records. This document supports the use of that mechanism by promoting the transmission of the relevant security records even in a non security aware server.

References

[RFC1034]
P. Mockapetris, ``DOMAIN NAMES - CONCEPTS AND FACILITIES,'' RFC 1034, ISI, November 1987.

[RFC1035]P. Mockapetris, ``DOMAIN NAMES - IMPLEMENTATION AND SPECIFICATION,'' RFC 1035, ISI, November 1987.

[RCF2065]
D. Eastlake, 3rd, C. Kaufman, ``Domain Name System Security Extensions,'' RFC 2065, CyberCash, Iris, January 1997

[RFCXXXX]
S. Bradner, ``Key words for use in RFCs to Indicate Requirement Levels,'' RFC XXXX, Harvard University, January 1997

Authors' Addresses

Mark Andrews
CSIRO - Mathematical and Information Sciences
Locked Bag 17
North Ryde NSW 2113
AUSTRALIA
+61 2 9325 3148
<Mark.Andrews@cmis.csiro.au>