Amends: RFC 1035, [UPDATE]

Deferred Dynamic Updates in the Domain Name System (DNS DEFUPD)

Status of this Memo

Abstract

Not all applications that perform dynamic updates using the protocol specified in [UPDATE] want their updates applied immediately. A case in point is [DHCP], wherein the DHCP lease time should control the lifetime of associated DNS data even if the DHCP client or server is not available at the time the DHCP lease expires.

The essence of this proposal is that DNS Dynamic Updates should be deferrable for some time delay period, after which they will be executed normally. Furthermore, RRs added or updated by a deferred update can have expiration times specified for them, as enforced by the automatic Dynamic Updates. Automatic serial number changes (as in [UPDATE]), dynamic zone slave notification (see [NOTIFY]) and incremental zone transfer (see [IXFR]) will jointly see to it that the zone changes are propagated with expedience.

1 - New Assigned Numbers

```
Opcode:   DEFUPD (6?)
RRtype:   TUU (34?)
RRtype:   TUE (35?)
```

2 - Message Format

The format and encoding of a DEFUPD is identical to that of UPDATE as defined in [UPDATE 2], except that the Opcode is DEFUPD rather than UPDATE, and there are two new RR types that can be used in the Additional Data section.

2.1 - Additional Data Section: TUU RR

In addition to the optional uses described in [UPDATE 2.6], a DEFUPD request's Additional Data section can include a TUU (Time Until Update) RR as follows:

```
Owner:      same as ZNAME (see [UPDATE 2.3])
Class:      same as ZCLASS (see [UPDATE 2.3])
Type:       TUU (new RRtype for this protocol)
TTL:        deferral time, relative, in seconds
RDLENGTH:   0
RDATA:      empty
```

Of particular note is the TTL, which contains the relative time delay, in seconds, starting from the time this DEFUPD is received by the primary master, before operations contained in the Update Section (see [UPDATE 2.5]) will actually be performed.

2.2 - Additional Data Section: TUE RR

In addition to the optional uses described in [UPDATE 2.6], a DEFUPD request's Additional Data section can include a TUU (Time Until Expiry) RR as follows:

```
Owner:      same as ZNAME (see [UPDATE 2.3])
Class:      same as ZCLASS (see [UPDATE 2.3])
Type:       TUE (new RRtype for this protocol)
TTL:        expiry time, relative, in seconds
RDLENGTH:   0
RDATA:      empty
```

Of particular note is the TTL, which contains the expiration time delay, in seconds, starting from the time this DEFUPD is received by the primary master, of all RRs added or updated by operations in the Update Section (see [UPDATE 2.5]).

3 - Server Behavior

A server, upon receiving a DEFUPD request, will first scan the request's Additional Data section in search of TUU or TUE RRs. If no RRs of either type TUU or TUE are found, then this request will be processed as a normal UPDATE with no special behaviour. If any TUU or TUE RRs are found, then processing continues as follows.

3.1 - Verify TUU RR

If any TUU RRs are found in the Additional Data section, this update will be processed with Deferral as explained below. If more than one TUU RR is found, signal FORMERR to requestor. The TUU RR's owner name and class are compared to ZNAME and ZCLASS; if a mismatch occurs, signal FORMERR to requestor. The TUU RR's RDLENGTH/RDATA is ignored by implementations of this specification, but future specifications may make use of this field.

3.2 - Verify TUE RR

If any TUE RRs are found in the Additional Data section, this update will be processed with Expiry as explained below. If more than one TUE RR is found, signal FORMERR to requestor. The TUE RR's owner name and class are compared to ZNAME and ZCLASS; if a mismatch occurs, signal FORMERR to requestor. The TUE RR's RDLENGTH/RDATA is ignored by implementations of this specification, but future specifications may make use of this field.

3.3 - Deferral

If a TUU RR was specified in the Additional Data section, this update will be processed with Deferral. Deferral means that the update will not be applied synchronously to the requestor's transaction cycle, but instead will be applied asynchronously at some potentially later time. The delay period is measured in seconds and expressed in the TUU's TTL.

3.3.1 - Store Deferred Update

Subject to per-server, per-zone, and per-RRset quotas, this UPDATE message is stored, persistently, on the name server. If per-RRset quotas are implemented, it is recommended that a DEFUPD ``count against'' all RRsets mentioned in the Update Section. If an implementation defined quota is exceeded by this deferred update, or if persistent storage is unavailable, signal SERVFAIL to requestor (leaving the zone in its former state). Note that even a deferred update whose TUU's TTL is zero (0), specifying immediate application, should be subject to quotas if the name server implements quotas.

3.3.2 - Send Early Response

Signal NOERROR to requestor.

3.3.3 - Apply Deferred Update

When a period of time equal to or greater than the TUU's TTL (measured in seconds) has elapsed since a DEFUPD was first received at the primary master, the DEFUPD message is applied to the zone as an UPDATE would be, except that no error can be signalled to the requestor. Thus, all errors not found and reported at the time the DEFUPD request was received are silent, affecting only the continued processing of the deferred update. Note that all sections are processed, including those processed before the deferred update were stored. Thus, prerequisites must hold before and after the deferral period.

3.4 - Expiry Processing

When a DEFUPD is applied, either during the requestor's transaction cycle or following the optional Deferral period, the inclusion of a TUE RR in the Additional Data section will cause this update to be processed with Expiry.

Expiry as expressed in the TUE's TTL is the time, in seconds, before all RRs added or modified by the Update Section will be automatically deleted by the primary master server. This time is relative to the time the DEFUPD message is processed, which might be after the delay period specified by a TUU RR.

3.4.1 - Initial TTL Limits

The TTL of all added or updated RRs in the Update Section will be maximized silently to one half of the Expiry time. This will cause downstream caching name servers to purge RRsets containing this RR at least once before expiry.

3.4.2 - TTL Half Life

Each time an RR's expiry reaches half of its previous value, that RR's TTL will be reduced to half of its previous value, until the TTL reaches a value less than or equal to sixty (60), i.e., one minute of real time, at which time the TTL will not be automatically reduced further by the primary master. It should be noted that RRs held in a server's memory need not be swept for half life processing, as long as the TTL changes appear when the RR next becomes externally visible, and as long as the ``zone has changed'' processing (see below) is done at the time of the half life expiration.

Whenever the TTL is automatically reduced by this process, the zone will be considered ``changed'' for the purpose of automatic SOA SERIAL increment (see [UPDATE 3.6]) and real time zone slave notification (see [NOTIFY]).

3.4.3 - Automatic Deletion

When the time since an RR was added or updated by a DEFUPD with Expiry exceeds the TUE's TTL as specified in that update, all RRs added or updated by that DEFUPD's Update Section will be automatically deleted by the primary master. This deletion can be deferred until the deleted RRs would next become visible, so long as the ``zone has changed'' processing (see below) is done at the time of expiration (i.e., when the automatic deletion is first deferred.)

Whenever automatic deletion occurs, the zone will be considered ``changed'' for the purpose of automatic SOA SERIAL increment (see [UPDATE 3.6]) and real time zone slave notification (see [NOTIFY]).

3.5 - Requirements for Persistence

Stored deferred updates should persist across name server restarts.

3.5.1 - Restarts and Deferral

In the event of a name server restart, all deferred updates whose TUU has expired must take effect before any network requests are processed using data from the affected zone, and before any Expiry processing takes place on RRs in the affected zone.

3.5.2 - Restarts and Expiry

In the event of a name server restart, all expiries must be checked as of the current time before any network responses are generated using data from the affected zone.

If an RR's expiry time has been reached while the name server was not running, that RR will be deleted. Otherwise, the RR's TTL will be set to one half of the time remaining before expiration, and half life processing as specified in Section 3.4.2 will be restarted.

If any RR is deleted or if an RR's TTL is changed during startup, then the zone will be considered ``changed'' for the purpose of automatic SOA SERIAL increment (see [UPDATE 3.6]) and real time zone slave notification (see [NOTIFY]).

4 - Requestor Behaviour

A requestor will generate and transmit a DEFUPD request as specified in [UPDATE 4], except that TUU and TUE RRs can be included in the Additional Data section.

4.1. The TUU RR, if specified, must have an owner name and class equal to the ZNAME and ZCLASS (see [UPDATE 2.3]). The TTL should be set to the delay, measured in seconds, before this update should be processed by the primary master. RDLENGTH should be set to 0, and RDATA should therefore be empty.

4.2. The TUE RR, if specified, must have an owner name and class equal to the ZNAME and ZCLASS (see [UPDATE 2.3]). The TTL should be set to the delay, measured in seconds, before all RRs added or changed by the Update Section will be automatically deleted by the primary master. This delay is measured starting from the time the update is applied, which could follow a deferral delay if a TUU RR was also included in this update.

5 - Notes on Resource Consumption

A TUE RR will require the primary master will initiate an automatic update approximately O(log2(TTL)) times over the life of an expiring RR. Even for massively sized zones, this is not considered an inappropriate load on the primary master server itself.

The network bandwidth consumed due to the use of TUE RRs is more noticeable, although for massively sized zones, the bandwidth requirements should flatten somewhat as many RRs will be automatically updated during any given cycle of NOTIFY and AXFR/IXFR.

6 - Security Considerations

This protocol suffers the same abject and intentional lack of security as [UPDATE], from which it inherits its basic semantics. In the absence of DNS Secure Updates, this protocol should not be used outside of an enterprise network, and only with great care within an enterprise network.

7 - Discussion Items for DNSIND and Namedroppers

Should the server's response to a DEFUPD include an opaque cookie called a ``deferred update ID'' which could be used in future DEFUPD requests to cancel or replace a previous deferred update?

Should automatic updates caused by a TUE RR be required to be batched up and processed at some minimum interval? For example, if we only checked for half life and expiration once per minute, this might drastically reduce the number of NOTIFY/AXFR/IXFR cycles caused by any given zone. We would have to recommend that all zones in a given server not be synchronized to the same timer, lest a server with many zones cause all of its zones to change and require NOTIFY/AXFR/IXFR in the same second.

Astute readers will have noticed that this proposal is a precise superset of [UPDATE], and by adding the optional behaviour and definitions of TUU and TUE to [UPDATE], we could do away with the separate Opcode for DEFUPD. This was only possible up until the time [UPDATE] went to proposed standard, but hopefully the intent was clear.

8 - References

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

[IXFR]
M. Ohta, ``Incremental Zone Transfer,'' Internet Draft, February 1996, <draft-ietf-dnsind-ixfr-06.txt>.

[NOTIFY]
P. Vixie, ``A Mechanism for Prompt Notification of Zone Changes,'' Internet Draft, March 1996, <draft-ietf-dnsind-notify-07.txt>.

[UPDATE]
P. Vixie (Ed), et al, ``Dynamic Updates in the Domain Name System,'' Internet Draft, March 1996, <draft-ietf-dnsind-dynDNS-09.txt>.

[DHCP]
Y. Rechter, ``Interaction between DHCP and DNS,'' Internet Draft, February 1996, <draft-ietf-dhc-dhcp-dns-00.txt>.

9 - Acknowledgements

Yakov Rechter assisted in the design of this protocol. Robert Elz assisted in the requirements definition.

10 - Author's Addresses

Paul Vixie
Internet Software Consortium
Star Route Box 159A
Woodside, CA 94062
+1 415 747 0204
<paul@vix.com>