Classless IN-ADDR.ARPA delegation

1. Status of this Memo

This document is an Internet-Draft. Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

To learn the current status of any Internet-Draft, please check the "id-abstracts.txt" listing contained in the Internet-Drafts Shadow Directories on ftp.is.co.za (Africa), nic.nordu.net (Europe), munnari.oz.au (Pacific Rim), ds.internic.net (US East Coast), or ftp.isi.edu (US West Coast).

2. Introduction

This document describes a way to do IN-ADDR.ARPA delegation on non-octet boundaries. The proposed method should thus remove one of the objections to subnet on non-octet boundaries but perhaps more significantly, make it possible to assign IP address space in smaller chunks than 24-bit prefixes, without losing the ability to delegate authority for the corresponding IN-ADDR.ARPA mappings. The proposed method is fully compatible with the original DNS lookup mechanisms specified in [1], i.e. there is no need to modify the lookup algorithm used, and there should be no need to modify any software which does DNS lookups either.

The document also discusses some operational considerations to provide some guidance in implementing this method.
3. Motivation

With the proliferation of classless routing technology, it has become feasible to assign address space on non-octet boundaries. In case of a Very Small Organization with only a few hosts, assigning a full 24-bit prefix (what has traditionally been referred to as a "class C network number") often leads to inefficient address space utilization.

One of the problems encountered when assigning a longer prefix (less address space) is that it seems impossible for such an organization to maintain its own reverse ("IN-ADDR.ARPA") zone autonomously. By use of the reverse delegation method described below, the most important objection to assignment of longer prefixes to unrelated organizations can be removed.

Let us assume we have assigned the address spaces to three different parties as follows:

```
192.0.2.0/25  to organization A
192.0.2.128/26 to organization B
192.0.2.192/26 to organization C
```

In the classical approach, this would lead to a single zone like this:

```
$ORIGIN 2.0.192.in-addr.arpa.
;
1         PTR  host1.A.domain.
2         PTR  host2.A.domain.
3         PTR  host3.A.domain.
;
129       PTR  host1.B.domain.
130       PTR  host2.B.domain.
131       PTR  host3.B.domain.
;
193       PTR  host1.C.domain.
194       PTR  host2.C.domain.
195       PTR  host3.C.domain.
```

The administration of this zone is problematic. Authority for this zone can only be delegated once, and this usually translates into "this zone can only be administered by one organization." The other organizations with address space that corresponds to entries in this zone would thus have to depend on another organization for their address to name translation. With the proposed method, this potential problem can be avoided.
4. Classless IN-ADDR.ARPA delegation

Since a single zone can only be delegated once we need more points to do delegation on to solve the problem above. These extra points of delegation can be introduced by extending the IN-ADDR.ARPA tree downwards, e.g. by using the first address or the first address and the network mask length (as shown below) in the corresponding address space to form the the first component in the name for the zones. For the problem described in the motivation section, the corresponding four zone files would look something like this (here shown also with network masks and network names in the form specified in [2] as well):

$ORIGIN 2.0.192.in-addr.arpa.
o IN SOA my-ns.my.domain. hostmaster.my.domain. ( ... )
;... 
; <<0-127>> /25
0/25  NS ns.A.domain.
0/25  NS some.other.name.server.
; 1  CNAME 1.0/25.2.0.192.in-addr.arpa.
2  CNAME 2.0/25.2.0.192.in-addr.arpa.
3  CNAME 3.0/25.2.0.192.in-addr.arpa.
;  <<128-191>> /26
128/26  NS ns.B.domain.
128/26  NS some.other.name.server.too.
129  CNAME 129.128/26.2.0.192.in-addr.arpa.
130  CNAME 130.128/26.2.0.192.in-addr.arpa.
131  CNAME 131.128/26.2.0.192.in-addr.arpa.
;  <<192-255>> /26
192/26  NS ns.C.domain.
192/26  NS some.other.third.name.server.
193  CNAME 193.192/26.2.0.192.in-addr.arpa.
194  CNAME 194.192/26.2.0.192.in-addr.arpa.
195  CNAME 195.192/26.2.0.192.in-addr.arpa.
Note that the use of network masks and network names as specified in [2] is optional, but is strongly recommended.

This approach to splitting up the responsibility for maintaining the IN-ADDR.ARPA mappings makes it necessary to install approximately 256 CNAME records in the parent zone more or less permanently for each size-256 chunk split up this way. Some people might view this as ugly; we will not argue that particular point. It is however quite easy to automatically generate the CNAME resource records in the parent zone once and for all, if the way the address space is partitioned is known.
The advantage of this approach over the other proposed approaches for dealing with this problem is that there should be no need to modify any already-deployed software. In particular, the lookup mechanism in the DNS does not have to be modified to accommodate this splitting of the responsibility for the IPv4 address to name translation on 'non-dot' boundaries. Furthermore, this technique has been in use for several years in at least one installation, apparently with no ill effects.

As usual, a resource record like

```plaintext
$ORIGIN   2.0.192.in-addr.arpa.
129       CNAME     129.128/26.2.0.192.in-addr.arpa.
```

can be conveniently abbreviated to

```plaintext
$ORIGIN   2.0.192.in-addr.arpa.
129       CNAME     129.128/26.2
```

Note also that it is legal to use slash ('/') in the name of the resource record (1.0/25.2.0.192.IN-ADDR.ARPA) because these are not host names; hence the restriction of [3] does not apply here.

5. Operational considerations

5.1 Recommended secondary name service

Some older versions of name server software will make no effort to find and return the pointed-to name in CNAME records if the pointed-to name is not already known locally as cached or as authoritative data. This can cause some confusion in resolvers, as only the CNAME record will be returned in the response. To avoid this problem it is recommended that the authoritative name servers for the delegating zone (the zone containing all the CNAME records) all run as slave (secondary) name servers for the 'child' zones delegated and pointed into via the CNAME records.

5.2 Alternative naming conventions

As a result of this method, the location of the zone containing the actual PTR records is no longer predefined. This gives flexibility and some examples will be presented here.

An obvious alternative to using the first address or the first address and the network mask length in the corresponding address space to name the new zones is simply to use some other (non-numeric) name. It is of course also possible to point to an entirely different part of the DNS tree (e.g. outside of the IN-ADDR.ARPA...
tree). It would be necessary to use one of these alternate methods if two organizations somehow shared the same physical subnet (and corresponding IP address space) with no "neat" alignment of the addresses, but still wanted to administrate their own IN-ADDR.ARPA mappings.

The following short example shows how you can point out of the IN-ADDR.ARPA tree:

```
$ORIGIN 2.0.192.in-addr.arpa.
@    IN   SOA  my-ns.my.domain. hostmaster.my.domain. ( ... )
; ...
1     CNAME  1.A.domain.
2     CNAME  2.A.domain.
; ...
129   CNAME  129.B.domain.
130   CNAME  130.B.domain.
;
$ORIGIN A.domain.
@    IN   SOA  my-ns.A.domain. hostmaster.A.domain. ( ... )
; ...
;
host1  A     192.0.2.1
1     PTR  host1
;
host2  A     192.0.2.2
2     PTR  host2
;
;
```

Done this way you can actually end up with the name->address and the (pointed-to) address->name mapping data in the same zone file -- some may view this as an added bonus as no separate set of secondaries for the reverse zone is required. Do however note that the traversal via the IN-ADDR.ARPA tree will still be done, so the CNAME records inserted there need to point in the right direction for this to work.

An approach as sketched below is an alternative approach using the same solution:
It is clear that many possibilities exist which can be adapted to the specific requirements of the situation at hand.

5.3 Other operational issues

Note that one cannot provide CNAME referrals twice for the same address space, i.e. you cannot allocate a /25 prefix to one organisation, and run IN-ADDR.ARPA this way, and then have the organisation subnet the /25 into longer prefixes, and attempt to employ the same technique to give each subnet control of its own number space. This would result in a CNAME record pointing to a CNAME record, which may be less robust overall.

Unfortunately, some old beta releases of the popular DNS name server implementation BIND 4.9.3 had a bug which caused problems if a CNAME record was encountered when a reverse lookup was made. The beta releases involved have since been obsoleted, and this issue is resolved in the released code. Some software manufacturers have included the defective beta code in their product. In the few cases we know of, patches from the manufacturers are available or planned to replace the obsolete beta code involved.

6. Security Considerations

Security considerations are not discussed in this memo.

7. Conclusion

The suggested scheme gives more flexibility in delegating authority in the IN-ADDR.ARPA domain, thus making it possible to assign address space more efficiently without losing the ability to delegate the DNS authority over the corresponding address to name mappings.
8. Acknowledgments

Glen A. Herrmannsfeldt described this trick on comp.protocols.tcp-ip.domains some time ago. Alan Barrett and Sam Wilson provided valuable comments on the newsgroup.

We would like to thank Rob Austein, Randy Bush, Matt Crawford, Glen A. Herrmannsfeldt, Daniel Karrenberg, David Kessens, Tony Li, Paul Mockapetris, Eric Wassenaar, Michael Patton, and Peter Koch for their review and constructive comments.

9. References


10. Author’s Addresses

Havard Eidnes
SINTEF RUNIT
N-7034 Trondheim
Norway

Phone: +47 73 59 44 68
Fax: +47 73 59 17 00

Email: Havard.Eidnes@runit.sintef.no

Geert Jan de Groot
RIPE Network Coordination Centre
Kruislaan 409
1098 SJ Amsterdam
the Netherlands

Phone: +31 20 592 5065
Fax: +31 20 592 5090

Email: GeertJan.deGroot@ripe.net

Paul Vixie
Internet Software Consortium
Star Route Box 159A
Woodside, CA 94062
USA

Phone: +1 415 747 0204

Email: paul@vix.com