Network Working Group   R. Elz
Request for Comments: 2182   University of Melbourne
BCP: 16   R. Bush
Category: Best Current Practice   RGnet, Inc.
                                                     S. Bradner
                                             Harvard University
                                                     M. Patton
                                                     Consultant
                                                     July 1997

              Selection and Operation of Secondary DNS Servers

Status of this Memo

Abstract

   The Domain Name System requires that multiple servers exist for every
   delegated domain (zone). This document discusses the selection of
   secondary servers for DNS zones. Both the physical and topological
   location of each server are material considerations when selecting
   secondary servers. The number of servers appropriate for a zone is
   also discussed, and some general secondary server maintenance issues
   considered.

Contents

1. Introduction

   A number of problems in DNS operations today are attributable to poor
   choices of secondary servers for DNS zones. The geographic placement
   as well as the diversity of network connectivity exhibited by the set
   of DNS servers for a zone can increase the reliability of that zone
   as well as improve overall network performance and access
   characteristics. Other considerations in server choice can
   unexpectedly lower reliability or impose extra demands on the
   network.

   This document discusses many of the issues that should be considered
   when selecting secondary servers for a zone. It offers guidance in
   how to best choose servers to serve a given zone.

2. Definitions

   For the purposes of this document, and only this document, the
   following definitions apply:

   DNS   The Domain Name System [RFC1034, RFC1035].

   Zone   A part of the DNS tree, that is treated as a
                        unit.

   Forward Zone   A zone containing data mapping names to host
                        addresses, mail exchange targets, etc.

Reverse Zone   A zone containing data used to map addresses
                       to names.

Server   An implementation of the DNS protocols able to
                       provide answers to queries. Answers may be
                       from information known by the server, or
                       information obtained from another server.

Authoritative Server   A server that knows the content of a DNS zone
                       from local knowledge, and thus can answer
                       queries about that zone without needing to
                       query other servers.

Listed Server   An Authoritative Server for which there is an
                       "NS" resource record (RR) in the zone.

Primary Server   An authoritative server for which the zone
                       information is locally configured. Sometimes
                       known as a Master server.

Secondary Server   An authoritative server that obtains
                       information about a zone from a Primary Server
                       via a zone transfer mechanism. Sometimes
                       known as a Slave Server.

Stealth Server   An authoritative server, usually secondary,
                       which is not a Listed Server.

Resolver   A client of the DNS which seeks information
                       contained in a zone using the DNS protocols.

3. Secondary Servers

   A major reason for having multiple servers for each zone is to allow
   information from the zone to be available widely and reliably to
   clients throughout the Internet, that is, throughout the world, even
   when one server is unavailable or unreachable.

   Multiple servers also spread the name resolution load, and improve
   the overall efficiency of the system by placing servers nearer to the
   resolvers. Those purposes are not treated further here.

   With multiple servers, usually one server will be the primary server,
   and others will be secondary servers. Note that while some unusual
   configurations use multiple primary servers, that can result in data
   inconsistencies, and is not advisable.

   The distinction between primary and secondary servers is relevant
   only to the servers for the zone concerned, to the rest of the DNS
   there are simply multiple servers. All are treated equally at first
   instance, even by the parent server that delegates the zone.
   Resolvers often measure the performance of the various servers,
   choose the "best", for some definition of best, and prefer that one
   for most queries. That is automatic, and not considered here.

   The primary server holds the master copy of the zone file. That is,
   the server where the data is entered into the DNS from some source
   outside the DNS. Secondary servers obtain data for the zone using
   DNS protocol mechanisms to obtain the zone from the primary server.

3.1. Selecting Secondary Servers

   When selecting secondary servers, attention should be given to the
   various likely failure modes. Servers should be placed so that it is
   likely that at least one server will be available to all significant
   parts of the Internet, for any likely failure.

   Consequently, placing all servers at the local site, while easy to
   arrange, and easy to manage, is not a good policy. Should a single
   link fail, or there be a site, or perhaps even building, or room,
   power failure, such a configuration can lead to all servers being
   disconnected from the Internet.

   Secondary servers must be placed at both topologically and
   geographically dispersed locations on the Internet, to minimise the
   likelihood of a single failure disabling all of them.

   That is, secondary servers should be at geographically distant
   locations, so it is unlikely that events like power loss, etc, will
   disrupt all of them simultaneously. They should also be connected to
   the net via quite diverse paths. This means that the failure of any
   one link, or of routing within some segment of the network (such as a
   service provider) will not make all of the servers unreachable.

3.2. Unsuitable Configurations

   While it is unfortunately quite common, servers for a zone should
   certainly not all be placed on the same LAN segment in the same room
   of the same building - or any of those. Such a configuration almost
   defeats the requirement, and utility, of having multiple servers.
   The only redundancy usually provided in that configuration is for the
   case when one server is down, whereas there are many other possible
   failure modes, such as power failures, including lengthy ones, to
   consider.

3.3. A Myth Exploded

   An argument is occasionally made that there is no need for the domain
   name servers for a domain to be accessible if the hosts in the domain
   are unreachable. This argument is fallacious.

     + Clients react differently to inability to resolve than inability
       to connect, and reactions to the former are not always as
       desirable.
     + If the zone is resolvable yet the particular name is not, then a
       client can discard the transaction rather than retrying and
       creating undesirable load on the network.
     + While positive DNS results are usually cached, the lack of a
       result is not cached. Thus, unnecessary inability to resolve
       creates an undesirable load on the net.
     + All names in the zone may not resolve to addresses within the
       detached network. This becomes more likely over time. Thus a
       basic assumption of the myth often becomes untrue.

   It is important that there be nameservers able to be queried,
   available always, for all forward zones.

4. Unreachable servers

   Another class of problems is caused by listing servers that cannot be
   reached from large parts of the network. This could be listing the
   name of a machine that is completely isolated behind a firewall, or
   just a secondary address on a dual homed machine which is not
   accessible from outside. The names of servers listed in NS records
   should resolve to addresses which are reachable from the region to
   which the NS records are being returned. Including addresses which
   most of the network cannot reach does not add any reliability, and
   causes several problems, which may, in the end, lower the reliability
   of the zone.

   First, the only way the resolvers can determine that these addresses
   are, in fact, unreachable, is to try them. They then need to wait on
   a lack of response timeout (or occasionally an ICMP error response)
   to know that the address cannot be used. Further, even that is
   generally indistinguishable from a simple packet loss, so the
   sequence must be repeated, several times, to give any real evidence
   of an unreachable server. All of this probing and timeout may take
   sufficiently long that the original client program or user will
   decide that no answer is available, leading to an apparent failure of
   the zone. Additionally, the whole thing needs to be repeated from
   time to time to distinguish a permanently unreachable server from a
   temporarily unreachable one.

And finally, all these steps will potentially need to be done by
resolvers all over the network. This will increase the traffic, and
probably the load on the filters at whatever firewall is blocking
this access. All of this additional load does no more than
effectively lower the reliability of the service.

4.1. Servers behind intermittent connections

A similar problem occurs with DNS servers located in parts of the net
that are often disconnected from the Internet as a whole. For
example, those which connect via an intermittent connection that is
often down. Such servers should usually be treated as if they were
behind a firewall, and unreachable to the network at any time.

4.2. Other problem cases

Similar problems occur when a Network Address Translator (NAT)
[RFC1631] exists between a resolver and server. Despite what
[RFC1631] suggests, NATs in practice do not translate addresses
embedded in packets, only those in the headers. As [RFC1631]
suggests, this is somewhat of a problem for the DNS. This can
sometimes be overcome if the NAT is accompanied by, or replaced with,
an Application Layer Gateway (ALG). Such a device would understand
the DNS protocol and translate all the addresses as appropriate as
packets pass through. Even with such a device, it is likely to be
better in any of these cases to adopt the solution described in the
following section.

4.3. A Solution

To avoid these problems, NS records for a zone returned in any
response should list only servers that the resolver requesting the
information, is likely to be able to reach. Some resolvers are
simultaneously servers performing lookups on behalf of other
resolvers. The NS records returned should be reachable not only by
the resolver that requested the information, but any other resolver
that may be forwarded the information. All the addresses of all the
servers returned must be reachable. As the addresses of each server
form a Resource Record Set [RFC2181], all must be returned (or none),
thus it is not acceptable to elide addresses of servers that are
unreachable, or to return them with a low TTL (while returning others
with a higher TTL).

In particular, when some servers are behind a firewall, intermittent
connection, or NAT, which disallows, or has problems with, DNS
queries or responses, their names, or addresses, should not be
returned to clients outside the firewall. Similarly, servers outside
the firewall should not be made known to clients inside it, if the

clients would be unable to query those servers. Implementing this
usually requires dual DNS setups, one for internal use, the other for
external use. Such a setup often solves other problems with
environments like this.

When a server is at a firewall boundary, reachable from both sides,
but using different addresses, that server should be given two names,
each name associated with appropriate A records, such that each
appears to be reachable only on the appropriate side of the firewall.
This should then be treated just like two servers, one on each side
of the firewall. A server implemented in an ALG will usually be such
a case. Special care will need to be taken to allow such a server to
return the correct responses to clients on each side. That is,
return only information about hosts reachable from that side and the
correct IP address(es) for the host when viewed from that side.

Servers in this environment often need special provision to give them
access to the root servers. Often this is accomplished via "fake
root" configurations. In such a case the servers should be kept well
isolated from the rest of the DNS, lest their unusual configuration
pollute others.

5. How many secondaries?

The DNS specification and domain name registration rules require at
least two servers for every zone. That is, usually, the primary and
one secondary. While two, carefully placed, are often sufficient,
occasions where two are insufficient are frequent enough that we
advise the use of more than two listed servers. Various problems can
cause a server to be unavailable for extended periods - during such a
period, a zone with only two listed servers is actually running with
just one. Since any server may occasionally be unavailable, for all
kinds of reasons, this zone is likely, at times, to have no
functional servers at all.

On the other hand, having large numbers of servers adds little
benefit, while adding costs. At the simplest, more servers cause
packets to be larger, so requiring more bandwidth. This may seem,
and realistically is, trivial. However there is a limit to the size
of a DNS packet, and causing that limit to be reached has more
serious performance implications. It is wise to stay well clear of
it. More servers also increase the likelihood that one server will
be misconfigured, or malfunction, without being detected.

It is recommended that three servers be provided for most
organisation level zones, with at least one which must be well
removed from the others. For zones where even higher reliability is
required, four, or even five, servers may be desirable. Two, or

occasionally three of five, would be at the local site, with the
others not geographically or topologically close to the site, or each
other.

Reverse zones, that is, sub-domains of .IN-ADDR.ARPA, tend to be less
crucial, and less servers, less distributed, will often suffice.
This is because address to name translations are typically needed
only when packets are being received from the address in question,
and only by resolvers at or near the destination of the packets.
This gives some assurances that servers located at or near the packet
source, for example, on the the same network, will be reachable from
the resolvers that need to perform the lookups. Thus some of the
failure modes that need to be considered when planning servers for
forward zones may be less relevant when reverse zones are being
planned.

5.1. Stealth Servers

   Servers which are authoritative for the zone, but not listed in NS
   records (also known as "stealth" servers) are not included in the
   count of servers.

   It can often be useful for all servers at a site to be authoritative
   (secondary), but only one or two be listed servers, the rest being
   unlisted servers for all local zones, that is, to be stealth servers.

   This allows those servers to provide answers to local queries
   directly, without needing to consult another server. If it were
   necessary to consult another server, it would usually be necessary
   for the root servers to be consulted, in order to follow the
   delegation tree - that the zone is local would not be known. This
   would mean that some local queries may not be able to be answered if
   external communications were disrupted.

   Listing all such servers in NS records, if more than one or two,
   would cause the rest of the Internet to spend unnecessary effort
   attempting to contact all servers at the site when the whole site is
   inaccessible due to link or routing failures.

6. Finding Suitable Secondary Servers

   Operating a secondary server is usually an almost automatic task.
   Once established, the server generally runs itself, based upon the
   actions of the primary server. Because of this, large numbers of
   organisations are willing to provide a secondary server, if
   requested. The best approach is usually to find an organisation of
   similar size, and agree to swap secondary zones - each organisation
   agrees to provide a server to act as a secondary server for the other

organisation's zones. Note that there is no loss of confidential
data here, the data set exchanged would be available publically
whatever the servers are.

7. Serial Number Maintenance

Secondary servers use the serial number in the SOA record of the zone
to determine when it is necessary to update their local copy of the
zone. Serial numbers are basically just 32 bit unsigned integers
that wrap around from the biggest possible value to zero again. See
[RFC1982] for a more rigorous definition of the serial number.

The serial number must be incremented every time a change, or group
of changes, is made to the zone on the primary server. This informs
secondary servers they need update their copies of the zone. Note
that it is not possible to decrement a serial number, increments are
the only defined modification.

Occasionally due to editing errors, or other factors, it may be
necessary to cause a serial number to become smaller. Never simply
decrease the serial number. Secondary servers will ignore that
change, and further, will ignore any later increments until the
earlier large value is exceeded.

Instead, given that serial numbers wrap from large to small, in
absolute terms, increment the serial number, several times, until it
has reached the value desired. At each step, wait until all
secondary servers have updated to the new value before proceeding.

For example, assume that the serial number of a zone was 10, but has
accidentally been set to 1000, and it is desired to set it back to
11. Do not simply change the value from 1000 to 11. A secondary
server that has seen the 1000 value (and in practice, there is always
at least one) will ignore this change, and continue to use the
version of the zone with serial number 1000, until the primary
server's serial number exceeds that value. This may be a long time -
in fact, the secondary often expires its copy of the zone before the
zone is ever updated again.

Instead, for this example, set the primary's serial number to
2000000000, and wait for the secondary servers to update to that
zone. The value 2000000000 is chosen as a value a lot bigger than
the current value, but less that 2^{31} bigger (2^{31} is 2147483648).
This is then an increment of the serial number [RFC1982].

Next, after all servers needing updating have the zone with that
serial number, the serial number can be set to 4000000000.
4000000000 is 2000000000 more than 2000000000 (fairly clearly), and

   is thus another increment (the value added is less than 2^31).

   Once this copy of the zone file exists at all servers, the serial
   number can simply be set to 11. In serial number arithmetic, a
   change from 4000000000 to 11 is an increment. Serial numbers wrap at
   2^32 (4294967296), so 11 is identical to 4294967307 (4294967296 +
    11). 4294967307 is just 294967307 greater than 4000000000, and
   294967307 is well under 2^31, this is therefore an increment.

   When following this procedure, it is essential to verify that all
   relevant servers have been updated at each step, never assume
   anything. Failing to do this can result in a worse mess than existed
   before the attempted correction. Also beware that it is the
   relationship between the values of the various serial numbers that is
   important, not the absolute values. The values used above are
   correct for that one example only.

   It is possible in essentially all cases to correct the serial number
   in two steps by being more aggressive in the choices of the serial
   numbers. This however causes the numbers used to be less "nice", and
   requires considerably more care.

   Also, note that not all nameserver implementations correctly
   implement serial number operations. With such servers as secondaries
   there is typically no way to cause the serial number to become
   smaller, other than contacting the administrator of the server and
   requesting that all existing data for the zone be purged. Then that
   the secondary be loaded again from the primary, as if for the first
   time.

   It remains safe to carry out the above procedure, as the
   malfunctioning servers will need manual attention in any case. After
   the sequence of serial number changes described above, conforming
   secondary servers will have been reset. Then when the primary server
   has the correct (desired) serial number, contact the remaining
   secondary servers and request their understanding of the correct
   serial number be manually corrected. Perhaps also suggest that they
   upgrade their software to a standards conforming implementation.

   A server which does not implement this algorithm is defective, and
   may be detected as follows. At some stage, usually when the absolute
   integral value of the serial number becomes smaller, a server with
   this particular defect will ignore the change. Servers with this
   type of defect can be detected by waiting for at least the time
   specified in the SOA refresh field and then sending a query for the
   SOA. Servers with this defect will still have the old serial number.
   We are not aware of other means to detect this defect.

Security Considerations

   It is not believed that anything in this document adds to any
   security issues that may exist with the DNS, nor does it do anything
   to lessen them.

   Administrators should be aware, however, that compromise of a server
   for a domain can, in some situations, compromise the security of
   hosts in the domain. Care should be taken in choosing secondary
   servers so that this threat is minimised.

References

   [RFC1034]   Mockapetris, P., "Domain Names - Concepts and Facilities",
               STD 13, RFC 1034, November 1987.

   [RFC1035]   Mockapetris, P., "Domain Names - Implementation and
               Specification", STD 13, RFC 1035, November 1987

   [RFC1631]   Egevang, K., Francis, P., "The IP Network Address Translator
               (NAT)", RFC 1631, May 1994

   [RFC1982]   Elz, R., Bush, R., "Serial Number Arithmetic",
               RFC 1982, August 1996.

   [RFC2181]   Elz, R., Bush, R., "Clarifications to the DNS specification",
               RFC 2181, July 1997.

Acknowledgements

Authors' Addresses

   Robert Elz
   Computer Science
   University of Melbourne
   Parkville, Vic, 3052
   Australia.

   EMail: kre@munnari.OZ.AU

Randy Bush
RGnet, Inc.
5147 Crystal Springs Drive NE
Bainbridge Island, Washington, 98110
United States.

EMail: randy@psg.com

Scott Bradner
Harvard University
1350 Mass Ave
Cambridge, MA, 02138
United States.

EMail: sob@harvard.edu

Michael A. Patton
33 Blanchard Road
Cambridge, MA, 02138
United States.

EMail: MAP@POBOX.COM