IEICE TRANS. COMMUN., VOL.E89–B, NO.4 APRIL 2006

PAPER Special Section on Internet Technology VI

Federated Domain Name Service Using DNS Metazones

Paul VIXIE[†a)], ***Nonmember***

SUMMARY Authority zones in the Domain Name System must be declared to have one or more authoritative name servers, usually consisting of one primary name server and several secondary name servers. These name servers are expected to synchronize zone data using DNS's zone transfer protocols, but the configuration of these synchronization relationships depends upon out of band information and manual processes. This paper describes a way to create name service federations such that a varying set of zones offered by a primary name server can be automatically configured for synchronization by secondary name servers. A sample implementation based on ISC BIND and Perl is described.

key words: *DNS, zone, management*

1. Introduction

DNS is a coherent autonomous distributed hierarchical database, designed in 1987 (see RFC 1034 [1]) to map Internet host names to their Internet Protocol (IP) addresses and to allow for future expansion. While DNS has scaled well compared to the systems it replaced, some management elements have $O(n)$ cost with expectedly high values for N. Among these elements is *secondary name service* for large and changing sets of DNS zones.

This paper describes a method of synchronizing secondary name service information among cooperating parties. We call the resulting automaton a *name service federation*, which can represent complex collections of both bilateral and unidirectional relationships between secondary name servers and primary name servers.

2. Background

In order to understand name service federations, it is necessary to first possess some basic knowledge about DNS's zone management mechanisms.

2.1 DNS Data Model: Zones

The Domain Name System's data model describes *authority zones* starting from some *apex* name and continuing downward toward *delegation points* where other zones begin, and containing *terminal* and *non-terminal* nodes which themselves contain *resource records* that are the system's payload. Consider the diagram in Fig. 1.

In this example, one zone has apex `COM` and another zone has apex `VIX.COM`. We say that `COM` has a *delegation* for `VIX.COM`, and that `VIX.COM` is a *zone cut* and also a *start of authority*.

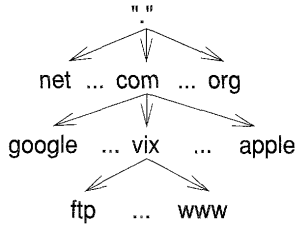

Fig. 1 DNS tree.

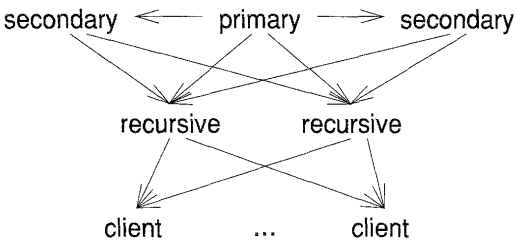

Fig. 2 Primary/secondary relationships.

2.2 DNS Zone Management

For a zone to be usable, it must be served by one or more *authority servers*, one of which is called the *primary* and the rest *secondary*. The primary name server for a zone is where changes to zone content can be made. Secondary name servers transfer the zone's content from the primary. The primary and secondary name servers, acting together, are responsible for answering queries about a zone from recursive name servers, who are in turn responsible for answering queries about all zones from their local client populations. Figure 2 shows these relationships.

The need for operational diversity among a zone's authority name servers usually leads to some form of outsourcing, whereby a zone operator selects multiple suppliers for authority name service. Outsourced domain name service is a common product available from Internet Service Providers and also from specialty providers who build redundant and high capacity infrastructure for this single purpose.

Most common, though, is the case where a zone operator operates an authority name server capable of serving many different zones, and offers to "trade" zone authority name services with other operators. For example,

Manuscript received July 20, 2005.
Manuscript revised October 4, 2005.
[†]The author is with Internet Systems Consortium, USA.
a) E-mail: vixie@isc.org
DOI: 10.1093/ietcom/e89–b.4.1144

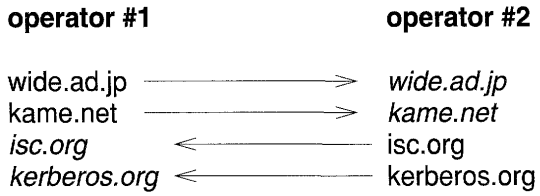

Fig. 3 Trade relationships.

operator "#1" might own the WIDE.AD.JP and KAME.NET zones, and operator "#2" might own the ISC.ORG and KERBEROS.ORG zones; in this case, operator "#1"'s authority name server might be the primary for WIDE.AD.JP and KAME.NET and secondary for ISC.ORG and KERBEROS.ORG, whereas operator "#2"'s authority name server would be the primary for ISC.ORG and KERBEROS.ORG and secondary for WIDE.AD.JP and KAME.NET. This arrangement conveniently diversifies the authority name service for all affected zones, at no direct additional cost to the zone's owners. The relationships are shown in Fig. 3.

2.3 DNS Dynamism

The principal advantage of DNS over prior systems is its dynamism. Prior systems relied on a central authority to distribute changes to all users. With DNS, payload data can be entered, modified, or removed at any time by any zone owner, and within the limits of positive and negative caching, these changes will be globally visible without significant distribution cost, delay, or special processing. But while this is true of payload data such as the IP address of WWW.VIX.COM in our earlier example, it is generally *untrue* of zone cut data and authority name server configurations, since those will require coordinated manual processing by cooperating human operators.

Often the process of adding a new zone cut, or of adding or removing service for a given zone by a given name server, begins with e-mail of the form "please add zone VIX.COM on your name server" or "zone DEC.COM no longer exists and can be removed." The processes for authenticating, acting upon, testing, and acknowledging these requests are usually not automated and are therefore labour intensive and error prone. Where automation exists, it is generally both proprietary and idiosyncratic—for example, using commercial tools that might not be generally available due to their cost, or using tools developed in-house that might be ideal for one business but incompatible with others, even assuming availability. Standardized automation in this area is needed.

3. Federated Name Service

The operator of a primary name server knows, by definition, what zones she offers primary name service for. For some or all of these zones, it is also necessary to either operate or discover secondary name servers. It is also possible that some zones on a primary name server will have their secondary name service chosen by the zone's owner rather than by the name server operator. The case which interests us here is when some set of zones on some primary name server is desired to be uniformly transferred to and served by a set of secondary name servers, and where additions and changes to the set of zones served in this way are both expected and automated. We call such a relationship a *name service federation.*

3.1 DNS Metazones

Federated name service can be formally automated using DNS *metazones* which are just zones that contain the names of other zones, and which are used to signal configuration changes necessary on secondary name servers to match the intent and desire of a primary name server's operator. A metazone has no special processing requirements for DNS itself; it is created, edited, served, secured, transferred, monitored, and maintained in exactly the same way as other zones. What makes a metazone special is how it is post-processed by secondary name servers.

A common alternative strategy for sharing peer-to-peer configuration data would be to use Web Services [2] and XML [3] rather than DNS Services [4] and in-band DNS. In-band DNS was chosen for carrying federated name service configuration information for three reasons:

1. by definition, DNS must already be working between prospective endpoints, meaning that all software, expertise, and firewall access is already present.
2. the features of DNS whereby zone information is transmitted only incrementally and only when necessary, and cached locally, are necessary for federated name service, and are not built into Web Services or XML.
3. DNS servers should have very few dependencies, because nearly all other services and protocols depend on DNS for proper operation. Web Services, for example, depends on DNS.

3.2 Metazone Syntax

A metazone's apex SOA and NS resource records are all as defined by the DNS protocol, with an important distinction that the NS resource records will never be used and are really just placeholders required by the protocol. It is not necessary for this zone to be properly delegated by NS resource records in the parent zone, other than as required by local housekeeping policy if any.

The rest of the zone consists of idiosyncratic PTR, RT and MG resource records that describes a name service federation. Note that these resource record types have standard meanings which are different from the meanings given to them in metazones, as shown in Table 1. While the overloading of standard resource record types is controversial [5], the authors believe that reuse of existing record types permits instant experimentation and deployment of new DNS-based services.

Table 1 Metazone RR types.

RR Type	Standard Meaning	Metazone Meaning
PTR	pointer	zone name
RT	route pointer	ordered list
MG	mail group	unordered list

```
$ORIGIN fh-sa.mz.vix.com.
$TTL 3600
@   SOA  ( ns.lah1.vix.com.
           hostmaster.vix.com.
           2005050704
           3600 1800
           604800 42 )
;
    NS   ns.lah1.vix.com.
    NS   ns.sql1.vix.com.
;
$ORIGIN masters.fh-sa.vix.com.
@   RT       10 ns-lah1.servers
$ORIGIN allow-transfer.fh-sa.vix.com.
@   MG       fh-sa.tsig.vix.com.
    MG       ns-ext.
$ORIGIN also-notify.fh-sa.vix.com.
@   RT       10 ns-ext.servers
;
$ORIGIN servers.fh-sa.vix.com.
ns-lah1 A        204.152.188.234
        AAAA     2001:4f8:2::9
ns-ext  A        204.152.184.64
        AAAA     2001:4f8:0:2::13
;
$ORIGIN zones.fh-sa.vix.com.
vix.com          PTR      vix.com.
anog.net         PTR      anog.net.
anog.org         PTR      anog.org.
```

Fig. 4 Example metazone.

For example, a metazone might be called `FH-SA.MZ.VIX.COM` in order to describe "the set of zones that `FH` wants `SA` to serve," and the content of the zone might be as shown in Fig. 4.

Note: This example demonstrates a subtle feature of DNS Master File syntax, whereby the absence of a trailing dot (.) on the owner names causes the current *origin* to be implied.

This metazone syntax design carefully balances three factors: (1) ease of expressing operator intent, (2) ease of expressing erroneous intent, and (3) likelihood of sending new and meaningless network traffic toward non-participating third parties. As specified, the most likely result of an operator induced syntax error will be: *inaction*.

Note that the SOA.MINIMUM is overloaded to mean *metazone syntax version*, and the current metazone syntax version number is 42. Implementations are expected to reject or ignore metazones having an unexpected syntax version number.

RT resource records are used to refer, by name, to the

1. read entire metazone, gathering:
   a. **A/AAAA**'s matching ***.servers**
   b. **MG**'s under **allow-transfer**
   c. **RT**'s under **masters**
   d. **RT**'s under **also-notify**
   e. **PTR**'s under **zones**
2. reject metazone upon any of:
   a. syntax version number mismatch
   b. inclusion of unrecognized RR
   c. use of unrecognized subdomain
3. for "masters" and "also-notify":
   a. reorder by **RT** priority field
   b. resolve to **A/AAAA** values given
4. for "allow-transfer":
   a. each value is a **TSIG**[9] key name
5. generate a zone for each PTR name:
   a. zone name is **PTR** value
   b. zone type is **secondary**
   c. master servers from **RT** values
   d. transfers allowed from **MG** values
   e. send **notify** to **RT** values

Fig. 5 Metazone algorithm.

authority name servers for all zones in this federation. `RT` was used because it provides a means of ordering within the resource record set; when describing lists of authority name servers, order is often significant. For example, some master servers might be expected to receive new zone data earlier, and should therefore be tried first. Also, since notifications of zone change can be transmitted lazily, it is sometimes desireable to ensure that some nameservers receive these notifications earlier than others. `RT` records must be under the `masters` or `also-notify` subdomains of the metazone. The `RT` resource records under the `also-notify` subdomain constitute the "Notify Set" described by RFC 1996 [6].

`MG` resource records are used to refer, by name, to access control lists which are to be locally defined on each secondary name server, and which are used to protect all zones in this federation from being copied in their entirety by outsiders. It is wise to fully qualify access control list names to prohibit name collisions, since a given secondary name server might be a member of more than one name service federation. `MG` records must be under the `allow-transfer` subdomain of the metazone.

`A` and `AAAA` resource records are used to describe the addresses of each name server referred to by an `RT` resource record described above. `A` and `AAAA` records must be under the `servers` subdomain of the metazone.

PTR resource records in metazones give the names of real zones who are members of the name service federation. The target of each PTR record must be the real zone name, whereas the PTR record's owner name is the real zone name

```
zone "vix.com" {
    type slave;
    file "sec/mz/fh-sa.vix.com/vix.com";
    masters { 2001:4f8:2::9;
              204.152.188.234; };
    also-notify { 2001:4f8:0:2::13;
                  204.152.184.64; };
    allow-transfer {
        key fh-sa.tsig.vix.com;
        key ns-ext;
    };
};

zone "anog.net" {
    type slave;
    file "sec/mz/fh-sa.vix.com/anog.net";
    masters { 2001:4f8:2::9;
              204.152.188.234; };
    also-notify { 2001:4f8:0:2::13;
                  204.152.184.64; };
    allow-transfer {
        key fh-sa.tsig.vix.com;
        key ns-ext;
    };
};

zone "anog.org" {
    type slave;
    file "sec/mz/fh-sa.vix.com/anog.org";
    masters { 2001:4f8:2::9;
              204.152.188.234; };
    also-notify { 2001:4f8:0:2::13;
                  204.152.184.64; };
    allow-transfer {
        key fh-sa.tsig.vix.com;
        key ns-ext;
    };
};
```

Fig. 6 Example named.conf.

given as a subdomain under the zones subdomain of the metazone.

3.3 Secondary Name Server Behaviour

As stated earlier, a metazone is "just a zone" from DNS's point of view, and it will be created, edited, transferred, etc., as a normal zone. However, whenever a secondary name server becomes aware of a new version of a metazone and pulls over the new content of such a zone, it takes the additional step of generating configuration directives based on the content of the zone. This can be done internally if the secondary name server is metazone-aware, or it can be done outside the name server using external tools. This paper describes a metazone implementation in Perl [7] suitable for ISC BIND [8].

One way to generate secondary name server configuration from the contents of a metazone would be to follow the algorithm given in Fig. 5.

In ISC BIND terms, the earlier metazone example can be represented by the configuration directives shown in Fig. 6.

Note that other metazone implementations are expected to be created, and each will have its own underlying configuration syntax. We hope that all such implementations will adhere to the metazone syntax given here, for interoperability reasons.

4. Usage

With the introduction of metazone technology into the DNS marketplace, we expect existing relationships to be optimized, new relationships to be formed, and some new services to be offered.

4.1 Unilateral Relationships

Where a primary name server operator currently uses e-mail or other methods to request changes in secondary name service, it is expected that a metazone will be created to formalize and automate the existing relationship, and that all subsequent changes in secondary name service will be signalled by updates to the content of the metazone.

4.2 Bilateral Relationships

Where two primary name server operators have agreed to trade secondary name service, two metazones will be created—one for each "direction"—to formalize and automate the existing relationship, as above.

4.3 Internal Operations

Where an organization operates their own primary and secondary name servers, it is expected that metazone technology will compare favourably to other methods of automating the usual "configure each zone on all of our name servers" property.

4.4 Stealth Primary

There is also an increasingly popular configuration called *stealth primary* whereby the primary name server is only used for zone transfers to secondary name servers, and is not advertised as one of the zone's name servers, and so never receives any normal updates [10] or queries for the zone. In this case it could be desirable to allow an owner of many zones to edit those zones locally and offer them en masse, using metazone technology, to the operators of their chosen public name servers.

4.5 Security Considerations

The primary purpose of DNS is publication, and so the default condition of all DNS data is that it is publically available. Metazone content is sensitive in that it could be used

by evildoers to plan *denial-of-service* attacks, or for data mining purposes. Therefore the suggested security profile of a DNS metazone is that it only be transferrable using TSIG or a similar mechanism, and that it only be available for queries from local clients.

4.6 Case Study: NS-EXT.ISC.ORG

NS-EXT.ISC.ORG is a general purpose name server that answers thousands of queries per second for about five hundred (500) zones, of which about fifty (50) are top-level domains (TLDs). The service is provided by a distributed cluster of five computers located worldwide, using a mixture [11] of local anycast and explicit name server naming. These name servers are part of several name service federations using metazones for configuration management, for both for internal and customer zones. While the costs and benefits are difficult to quantify, acceptance and appreciation has been universal among system administrators and customers who tested this technology and compared it to their previous manual processes.

5. Sample Implementation

The proof of concept for federated name service has been operating inside ISC since early 2004. It relies on UNIX-like name servers running ISC BIND, with a small entry in the system crontab, and three small Perl scripts. It is hoped that this functionality will be folded into a future ISC BIND release, to remove the dependencies on UNIX, cron, and Perl. It is also hoped that other name service implementors will add federated name service by means of DNS metazones, and that they will strictly adhere to the metazone syntax described here, in order to promote interoperability and overall market size.

5.1 Files

Table 2 shows the file structure of ISC's proof of concept implementation of federated name service. This implementation generates configuration files for ISC BIND.

5.2 Theory of Operation

The files described above are installed in a directory "/var/named/mz," and a crontab entry is added which will cause the mz.sh script to run every three minutes. This script checks the SOA SERIAL of each metazone by querying the local name server looking for changes since the last successful cron run. If there are any changes, then the modified metazone is given to the mzg.pl script which will generate a named.conf-format file corresponding to the metazone's content. The end result of running mz.sh from cron is that the file "/var/named/mz/var/all.inc" will be built or rebuilt, and if any changes occurred which require action by the name server, the command rndc reconfig will be given, which tells the local name server to reload its configuration. Note that rndc is an ISC BIND9 command, and that secondary name servers using ISC BIND8 would use ndc reconfig instead.

Note that the metazone is fetched by mzg.pl using the DNS zone transfer protocol (AXFR) from the local name server. This is because the metazone content could be in DNS Master File format, binary format, or SQL, or something else. It is necessary that the metazone be transferrable from the local name server host—and therefore necessary that the local security policies permit this access.

Table 2 File structure.

File	*Contains*
README	operating instructions
bin/getsoa.pl	retrieve a zone's serial number
bin/mz.sh	overall driver—runs from cron
bin/mzg.pl	processes one metazone
etc/config	options to govern local behaviour
etc/config.default	default options
etc/crontab.example	example crontab entry
etc/zones	list of local metazones
var/	directory for generated files

5.3 named.conf Changes

To bring a metazone to life, two directives must be added to named.conf on each secondary name server. The first is to establish a secondary name service relationship for the metazone. The second is to include by reference the configuration data generated by the bin/mz.sh script. For our example, those directives are shown in Fig. 7.

If a secondary name server subscribes to more than one metazone, then each one will have to be declared in a zone directive, but a single include directive for the one generated "mz/var/all.inc" is all that's required. (In ISC BIND, the include directive is replaced by the contents of the file thus identified.)

5.4 Clustering Features

In order to support software based DNS clustering [12], the sample implementation of federated name service allows each federated secondary name server to declare the other members of its cluster, whose addresses are then prepended to the generated also-notify and masters directives. Other implementors are advised to consider adding a sim-

```
zone "fh-sa.mz.vix.com" {
    type slave;
    masters { 2001:4f8:2::9;
              204.152.188.234; };
    file "sec/fh-sa.vix.com";
    notify no;
};

include "mz/var/all.inc";
```

Fig. 7 New directives.

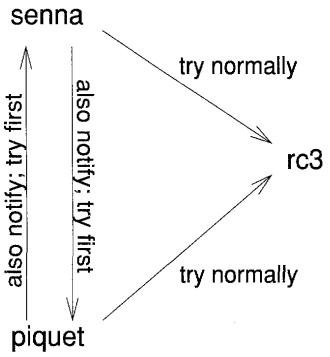

Fig. 8 DNS clustering.

ilar feature. As an example, consider a name service federation consisting of three servers: `senna` and `piquet` are secondary servers, and `rc3` is the primary. The diagram shown in Fig. 8 shows the data flow relationships between these three servers.

5.5 Software Availability

The metazone sample implementation described here is available, like BIND and all other software created at ISC, under a BSD-like license. Please send mail to "`info@isc.org`" to request a free copy of this software.

6. Conclusion

Relationships between operators of primary and secondary name servers are a key element of Domain Name System scalability. Federated name service using DNS metazones is a practical method of automating such relationships. A single standard for name service federation metazone data format is necessary for interoperability. The sample metazone implementation for ISC BIND described here should be easily portable to non-BIND name server implementations.

Acknowledgements

This work was supported by Keio University.

This work was guided by the experiences of the ISC operations team, especially including Peter Losher and Joao Damas.

References

[1] P. Mockapetris, "Domain name system—Concepts and facilities," RFC1034, 1987.
[2] W3C, Web Services Activity—W3C.
[3] W3C, Extensible Markup Language (XML)—W3C.
[4] P. Vixie and A. Kato, "Modern DNS as a coherent dynamic universal database," IEICE Trans. Commun. (Japanese Edition), vol.J87-B, no.10, pp.1534–1541, Oct. 2004.
[5] I.A. Board, "Design choices when expanding DNS," draft-iab-dns-choices-02.txt, 2005.
[6] P. Vixie, "A mechanism for prompt notification of zone changes," RFC1996, 1996.
[7] Perl Foundation, Perl Home Page.
[8] Internet Systems Consortium, Inc. (ISC), ISC BIND Home Page.
[9] P. Vixie, et al., "Secret key transaction authentication for DNS (TSIG)," RFC2845, 2000.
[10] P. Vixie, et al., "Dynamic updates in the domain name system," RFC2136, 1997.
[11] J. Abley, "Hierarchical anycast for global service distribution," Technical Report ISC-TN-2003-1, ISC, 2003.
[12] J. Abley, "A software approach to distributing requests for DNS service," Technical Report ISC-TN-2004-1, ISC, 2004.

Paul Vixie co-founded ISC.ORG in 1994 and was named President in 2002. During the intervening years he was the primary author of BIND8, and served as CTO or CEO for several technology companies including Metromedia Fiber, PAIX.NET, and MAPS.