DNS Zones Revisited

Ward van Wanrooij, Aiko Pras

***Abstract* - Recent research suggests that, due to misconfiguration, DNS reliability and performance is not always as good as it should be. This paper therefore investigates the correct configuration of DNS zones, by checking if main configuration requirements, recommendations and best-practices rules have been followed. Our research shows that almost one out of four zones fail to pass one or more of our tests. Our study reveals an interesting correlation: if the number of name servers for a single zone exceeds a certain number, reliability and performance usually decreases.**

I. Introduction

The correct and error free operation of the Domain Name System (DNS) is crucial for the reliability of most internet applications, like e-mail, web browsing and file transfer. Recent research [1] indicates that, despite DNS's relative robustness against physical failures, DNS may still have performance and reliability problems, because of human configuration errors.

This research embraces and extends the research of Pappas et al., in an attempt to find more and different DNS misconfigurations. In contrast with the checks performed by Pappas et al., our checks will be performed on a random selection of domain names; our selection process does therefore not use criteria like visited websites, popular websites, presence of a reverse DNS record or the possibility to transfer zones. To ensure that our findings will be applicable to the majority of DNS zones worldwide, we started from one of the best managed zones in the world: the .NL zone ([1] Fig. 2a). Our checks are thus performed on a random selection from the .NL Top Level Domain zone file.

Our checklist is based on the technical requirements defined by the SIDN [5] (regulatory body for .NL zone), research by Pappas et al., recommendations extracted from RFCs (Request For Comments, a set of technical and organizational notes concerning the internet) as well as personal experience. The implications of failing a test can range from degraded performance, difficulties in moving a domain name, to being unable to change a name server because SIDN will reject the associated zone since it does not meet its requirements.

The structure of this paper is as follows. Section 2 starts with a recapitulation of the general operation of DNS, followed by an overview of our research method (Section 3). Based on above-mentioned sources, Section 4 proposes a number of checks. The results of these checks are discussed in Section 5. Section 6 provides the conclusions.

Ward van Wanrooij, University of Twente, PO Box 217, 7500 AE, Enschede, The Netherlands, w.a.h.c.vanwanrooij@student.utwente.nl
Aiko Pras, University of Twente, PO Box 217, 7500 AE, Enschede, The Netherlands,[2]a.pras@utwente.nl

II. General Operation of DNS

An organizational model of the DNS system, outlining SIDN's position and its importance for .NL domains, has been presented in [3]. Technically speaking, DNS is responsible for the mapping between host names and IP-addresses, a process called resolving [9] [10]. A simplified view of DNS, referring to Fig. 1, is that of a tree, containing nodes called *zones* (e.g. utwente.nl) and *leafs* called host names (e.g. www.utwente.nl). Zones define in their Resource Records (RRs), amongst others, the mapping between host and IP addresses (A records), mail servers (MX records), addresses of other authoritative name servers for this zone (NS records), as well as administrative data (SOA, TXT records). Zones not only contain RRs for the current zone, but possibly also for deeper zones and deeper host names.

Systems can assume different roles (possibly multiple at any time) in DNS (Fig. 1):

- Content server: a name server providing authoritative answers (e.g. F: ns1.utwente.nl). Content servers can either be masters (serving the zone from local data) or slaves (serving the zone from data provided over the internet by a master server).
- Caching server: a name server resolving a host name recursively for a client (e.g. B: 192.168.1.1). Recursive querying is performed by iteratively executing queries in lieu of the client and passing back the final answer.
- Client: a system using a caching server to resolve a host name (e.g. A: 192.168.1.203).

A typical DNS query is resolved in these steps (simplified, assuming no errors, again referring to Fig. 1):

1. A asks B to resolve www.utwente.nl using recursion.
2. After B checks its cache, B asks C for the address of www.utwente.nl (no recursion).
3. C returns B the address of a server handling the .nl zone (D).
4. B asks D for the address of www.utwente.nl (no recursion).
5. D returns B the address of a server handling the . utwente.nl zone (F).
6. B asks F for the address of www.utwente.nl (no recursion).
7. F returns B the address of www.utwente.nl and the time this information can be considered valid.
8. B returns A the requested information.

If C encounters a serious error querying one server, it proceeds to the next server handling the zone, if available.

III. Research Method

At the time of this study (November 2004), almost 1,300,000 domain names had been registered in the .NL zone. To create a representative test collection, we selected 10,000 names at random from this zone file. A sample of 10,000 names assures a 95% reliability [6], meaning that the maximum deviation stays within 1%. Unfortunately, copies of the .NL zone file can no longer be obtained from the SIDN due to “security and privacy reasons”. For that reason

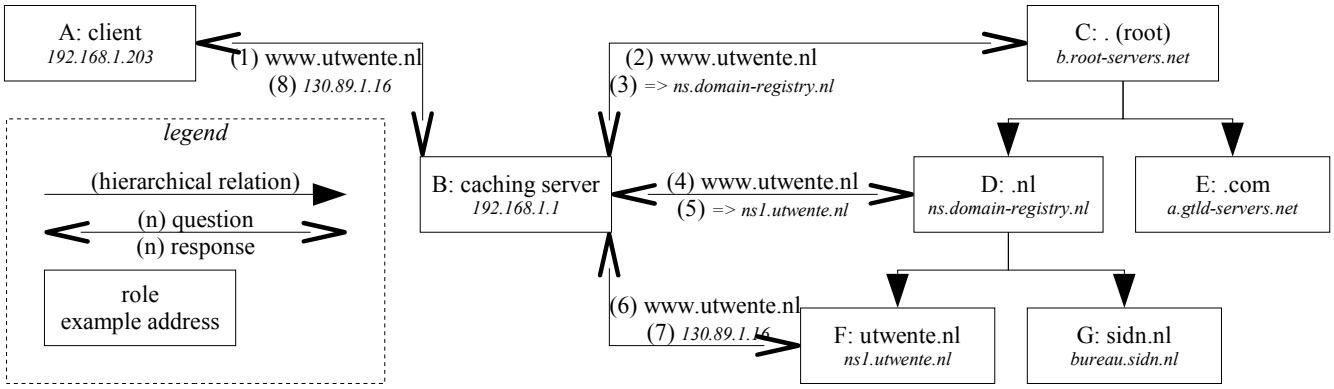

Fig. 1. Path of a typical DNS query

we had to use an older copy, created at 27 February 2002; at that time the rules to copy zone files were more relaxed.

From this zone file, containing 700,000 domain names, domain names were selected in a random fashion, checked whether their registration was current (using the SIDN "whois is" interface) until 10,000 names had been collected. This occurred after 11,741 names had been checked.

All 10,000 names, plus their name servers according to the .NL zone, were inserted into a database as seeds to retrieve. For each tuple of zone name and name server DNS records were retrieved using an ANY query. If the results contained additional name servers, these were inserted into the database for retrieval too.

The software that was written to automate this process uses DiG version 9.3.0 [7] to retrieve and parse responses from name servers. If necessary, this software repeats each query up to three times, using a maximum response time of 40 seconds per query. If responses arrive after these 40 seconds, but within 3*40=120 seconds (see section 7.2 of [15]: Dead / Unreachable servers), this is logged for later analysis. Note however that this did not happen: the maximum response time for our queries was approximately 18 seconds. All these queries were performed in the evening (CET) of November 5th 2004, using 20 parallel processes, in about 4 hours.

IV. Checklist

Requirements, recommendations and best-practices rules for configuring DNS zones have been defined by the SIDN [4], literature [1] and a number of RFCs. Based on this, the following checklist has been created:

- *Does the name server respond with an authoritative answer to an ANY query for the domain? (Section 5.1)*
  No authoritative answer to a query is a serious error condition, also touching end-users in almost all situations.
- *How many name servers serve the zone? (Section 5.2)*
  Having too little or too many name servers can adversely impact performance and reliability.
- *Are the name servers in the answer a superset of the entries in the .NL zone? Can any elements of this superset be registered in the .NL zone? (Section 5.3)*
  NS records in deeper zones must match or extend the information in the .NL zone.
- *Does the SOA record primary name server correspond with one of the entries in the .NL zone? Do the values of the remainder of the SOA record (serial number, refresh, retry, expiry, minimum TTL) confirm to recommendations? (Section 5.4)*
  Correct values in the SOA record are crucial for reliable and efficient zone replication and caching.
- *How many MX servers are defined in a zone? (Section 5.5)*
  The number of MX servers influences the reliability and performance of incoming mail for a domain.
- *What kind of name server software serves the zone? Does the name server also act as a caching server? (Section 5.6)*
  Availability and performance of the server also affect the reliability and performance of DNS.

V. Analysis

This section analyses the outcome of the checks that were presented in the previous section.

A. Authoritative Response

Querying the .NL TLD (Top Level Domain) servers for the 10,000 zones results in 22,311 zone-server tuples and an equal number of DNS queries. The results of these queries add an additional number of 917 zone-server mappings in the database, totalling 23,228 queries distributed over 3,059 unique servers. 21,719 of these queries (93.50%) generate an authoritative answer: in other words these queries are answered correctly.

Fig. 2 illustrates the distribution of different error conditions (collectively called "lame delegations") among the remaining 1,509 responses (6.50%). The observed errors are:

- The designated name server is not resolvable. This can be caused by an error in an intermediary name server or a cyclic zone dependency. Example: makelaarsoverzicht.nl is served by ns.aaaaa-hosting.nl, however ns.aaaaa-hosting.nl is not served by this server, so glue (A) records are absent in the .NL zone. The servers responsible for aaaaa-hosting.nl return NXDOMAIN, non existent domain, for ns.aaaaa-hosting.nl so resolving the name server and also the query fails.
- The designated name server is not reachable; this error occurs if no response is received from the server. Note that besides hard- and software errors configuration can also be the cause for this state. For example ns.vuurwerk.nl is configured to disregard queries for zones that it is not authoritative for.
- The name server returns an error condition. Out of the

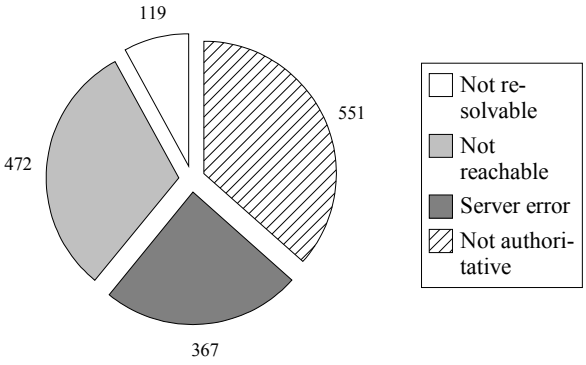

Fig. 2. Failed queries by error condition

five possible errors (FORMERR, NOTIMP, NXDOMAIN, REFUSED, SERVFAIL) only SERVFAIL (357) and REFUSED (10) have been observed. REFUSED means the server has been specifically instructed to answer with an error to the question (possibly based on IP range), SERVFAIL can amongst others indicate missing zone data.

- The name server is not authoritative for the zone. This occurs when the server is not configured as a content DNS server for the current zone. Although such server may respond with valid data (acting as a cache DNS server, retrieving the information from another name server), this data should not be regarded authoritative (hence the missing of the aa bit) and should not be used by clients or cache servers.

The deviation from specifications in all error conditions is obvious: a content DNS server should respond with an authoritative answer to a non- recursive query (for example [5] item 5).

The implications of these errors are:

- If, for a specific zone, none of the servers listed in the . NL zone file responds without errors, that zone (and its sub zones) is not resolvable at all. This happened to 398 of the 10,000 .NL zones.
- If, for a specific zone, one server responds with an error, it will at least slow down the query resolving (more details in the next paragraph) but depending on the behaviour of the resolver might also result in not resolving at all. For example, we detected misconfiguration of the renault.nl domain because the web browser reported www.renault.nl could not be found. We presume the resolver (caching DNS) returned an error, after going through these steps:
  -Asking .NL TLD server for www.renault.nl returns {ns2.xs4all.nl, clara.renault.fr, xenia.renault.fr}.
  -Asking ns2.xs4all.nl for www.renault.nl returns no valid answer (not authoritative).
  -Asking .FR TLD server clara.renault.fr returns {auth60.ns.uu.net, ns1.gip.net, ns2.gip.net, anna.renault.fr, clara.renault.fr, xenia.renault.fr}.
  -Asking auth60.ns.uu.net for www.renault.nl returns no valid answer (not authoritative).
- A zone served by one or more misconfigured servers, may not resolve when one of its correctly configured servers malfunctions (limited or no redundancy).
- The entries in the .NL zone file for a zone can not be modified (e.g. moving name servers, switching providers) while one of these error conditions persists. The Dutch regulatory body SIDN performs a sanity check, based on its technical regulations [5], on a zone before allowing modifications to the .NL zone file.

Table 1 provides insight in the round-trip time of queries and the effect of server misconfigurations. A distinction is made between:

- No errors: Queries returning no error.
- Errors, limited: Queries returning an error.
- Errors, full: Queries returning an error or no answer, substituting 4,000 ms as cut-off time.
- Errors, corrected: Queries returning an error or no answer, substituting 4,000 ms as cut-off time, maximizing individual response times to 4,000 ms.

TABLE 1. STATISTICAL PROPERTIES OF QUERY TIME IN MS

Query set	min	avg	max	stddev
No errors	9	29.2	16375	173.2
Errors, limited	9	492.9	17944	2358.1
Errors, full	9	1866.5	17944	2512.7
Errors, corrected	9	218.3	4000	830.3

Table 1 shows the impact of errors on query response times is substantial: considering the needed extra query after an error, the elapsed combined query-time of a transaction involving an error is almost ten-fold that of a transaction without errors. Broadly speaking, this matches previous conclusions [1].

Our study concludes 1,509 of the total of 23,228 responses were invalid. These responses are excluded from any subsequent test. They affect 802 domain names (8.02%), 398 of them do not resolve at all.

B. Number of Name Servers

The number of name servers, both defined in the .NL zone file and in name server (NS) records of the zone, is counted for the remaining 9.602 zones (Table 2).

SIDN regulations ([5] item 4) define the minimum number of name servers to be 2, [14] section 5 recommends a minimum of three and a maximum of five name servers in most cases. Generally, especially for .NL zones, two name servers should suffice for personal and small-and-medium business domain names. For larger and/or internationally operating companies a number of three name servers should be used. Only in exceptional cases, a higher number (four or five) is suited.

TABLE 2. DISTRIBUTION OF NAME SERVER COUNT AMONG ZONES

Number of name servers	Number of zones (A, n=9.602)	Number of zones (B, n=9.602)
1 NS	0 (0%)	5 (0.05%)
2 NS	6842 (71.26%)	6934 (72.21%)
3 NS	2464 (25.66%)	2427 (25.28%)
4 NS	264 (2.75%)	222 (2.31%)
5 NS	24 (0.25%)	14 (0.14%)
6 NS	7 (0.07%)	0 (0%)
7 NS	1 (0.01%)	0 (0%)

Table 2 shows the number of zones having a specific count of name servers by using two methods.

- Method A counts all distinct server names serving a zone, extracted from both the .NL zone and NS records

in the zone.

- Method B counts all distinct server names serving a zone extracted from the NS records in the zone.

Both counts are equal for a correct second level zone. However due to various possible configuration errors, e.g. renaming name servers without notifying SIDN, omitting name servers in NS records or removing a zone from a secondary or tertiary name servers without notifying SIDN, these numbers can differ. In a later chapter the differences between the records in the first level and second levels is examined in more detail.

For now, it suffices to establish that both counts are tolerably equal, follow the guidelines for the number of domain names.

[14] remarks "More servers also increase the likelihood that one server will be misconfigured, or malfunction, without being detected.". However, a higher server count also suggests more resources (money for hosting, bandwidth, education) and consequently more knowledgeable administrators. Or doesn't it?

Too Many Name Servers?

For each method of counting and each count of name servers, the percentage of zones having at least one server returning an error condition is computed. This preliminary data shows a 70% error rate for zones with 4 name servers as a result of apparent ignorance at ISP Tiscali (records show that many domain names are served by four servers, only two of which return authoritative answers). After correcting the data by excluding these zones from this computation and ignoring the bogus data for 1, 6 and 7 name servers, Fig. 3 emerges.

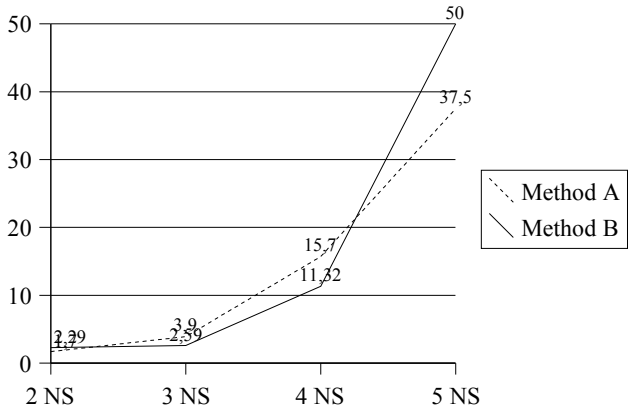

Fig. 3. Percentage of zones containing at least one malfunctioning server grouped by name server count

Clearly, an increased number of name servers increases the number of configuration errors and the chance of receiving an erroneous answer and thus decreases performance (Table 1) and reliability. We expected the number of configuration errors to decrease when the number of servers increases, because we assessed the knowledge level of "professional" hostmasters (responsible for larger or more important zones) to be higher than that of "amateur" host masters (making do with the minimum number of servers required). This hypothesis has been proven false. Possible causes for this interesting correlation can be:

- Errors caused by misconfiguration are less obvious, the chance of hitting a malfunctioning server, either in a normal situation or in a situation where one of the functioning servers is down, is smaller.
- Zone and/or server administrators attach less importance to the correct configuration of a single server, mistaking fault-tolerant for error-free.
- Zones having many (4 or more) name servers are often tailored by hand, in contrast to mass production of low cost web hosting zones which are frequently generated by scripts and third-party control panels. Human intervention increases the chance of making mistakes.

This data has been set against both the method A and method B count, because we wanted to see the result against both the registered (A) and probably intended (B) number of servers, as to not skew the results of higher name server counts, because the percentage might be influenced significantly if even a small percentage of the next lower category contains an additional name server in count A versus count B. The graph shows this fear is unwarranted.

The data reaffirms our beliefs that only in exceptional cases four or more name servers should be defined for a single zone.

C. NS Record

Name Server (NS) records include information on (other) authoritative name servers for the zone (see Section 2). This information can be used e.g. for load-balancing, fail-over and zone replication among servers. For example, the domain utwente.nl has three name servers: ns1.utwente.nl, ns2.utwente.nl and ns3.utwente.nl. These three servers should be known by the .NL zone (which is maintained by SIDN), as well as the utwente.nl zone, which is maintained by the University of Twente.

With respect to the NS records returned by the root (such as .NL) and the leaf (such as .utwente.nl), the following four situations may be observed:

- The NS records provided by the root and leafs are equal (8,975 occurrences).
  This is one of the two correct configurations.
- Different leafs provide different NS records (34 occurrences).
  Although in exceptional cases having different DNS servers hand out different NS records can be advantageous (e.g. load-balancing using different DNS servers for distinct geographical areas), in almost all cases it is a sign of error (inconsistent zones) or at least sub-optimal configuration.
- All leafs omit NS records from root (123 occurrences).
  This is a serious misconfiguration, and an explicit violation of SIDN requirement 6a (leaf zone should at least contain the servers listed in the root zone).
- All leafs provide extra NS records.
  Since the SIDN does not accept more than three NS records in the root zone per .NL domain name, two possibilities exist. The leaf has already registered three NS records (14 occurrences), or the leaf has registered less (456 occurrences). In the first case, this is a correct configuration; in the latter case reliability could significantly be improved by registering the additional NS record.

In conclusion 1.63% of the surveyed (9,602) zones has one or more seriously misconfigured NS records and 4.75% of the zone can obtain additional reliability and performance by registering an additional record in the NL root zone.

D. SOA Record

The SOA record of a zone is an instrumental record for replication and caching of that zone. The SOA record contains seven values:

- MNAME: Name of the primary master server (SIDN regulation 6b states explicitly that this value should correspond with the primary NS record).
- RNAME: E-mail address of the person or role account responsible for maintaining the zone. Although seemingly trivial, several remarks can be made about the choice:
  -Choosing an e-mail address being in the same zone can lead to unwanted situations in case of serious zone errors.
  -Choosing an already in-use personal account can be problematic due to e.g. holidays and employee turnover. [11] recommends the definition of a dedicated account named hostmaster for this reason.
  -Choosing an address indicating the mailbox is read now and in the future is crucial to inviting human feedback on your zone. Although the survey of actually sending mail to the RNAME is out of the scope of this research, e-mail addresses like zulucpanel@[zone] or [account] @hotmail.com predict trouble.
- Serial number: The number serves as a unique identifier[1] of a certain version of a zone and should always be incremented after a change. This value is a 32 bit unsigned, wrappable integer. Although any numeric value can be used, several considerations must be made when choosing a numbering sequence:
  -The sequence should allow for enough updates for the lifetime of the zone.
  -The sequence should allow for human interpretation for debug purposes.
  -The sequence should not be overflowing (for more information, consult [12]).
  -The sequence should not cause any problems when moving providers.
  [11] recommends using the numbering sequence YYYYMMDDnn to satisfy the above mentioned constraints.
- Refresh, retry, expiry: These values control master-slave replication. The value refresh specifies the interval at which a slave server should poll a master to check whether the serial number has changed. If the master server is unreachable, the slave tries again after the retry interval. If after the expiration interval the slave still has not contacted the master, it discards the entire zone. Obviously, the following inequalities should always hold:
  `retry < expiry`
  Logically, the following inequality should also hold:
  `retry < refresh << expiry`
  More formally, the following recommendations [8] exist:
  ```
  expiry < 6 months
  expiry > 7 days
  expiry > refresh+retry
  expiry > 2*retry
  refresh > 2*retry
  ```
  Aside from this polling mechanism, several proprietary push mechanisms exist (DNS NOTIFY, back end replication). This does not relieve the zone administrator of specifying sensible values!
- Minimum TTL: This Time-To-Live value specifies the time a caching server can hand out a record without requerying a content server (this value can be overridden on a per RR basis). [11] and [8] recommend a value of one to five days; however we consider this interval too high and recommend a maximum value of one day: to change a DNS record with a TTL of 3 days, the record's TTL needs to be changed at least three days in advance, or inconsistent DNS data will exist in hosts for at least three days.

The 9,602 active zones contained 76 zones whose SOA records differed between servers. Zones exhibiting this severe error have been excluded from a further SOA examination.

This examination established the following facts:

- The MNAME record of 81 zones (0.85%) does not correspond with any of the NS records (violation of 6b). No check has been performed on the remaining 99% to see whether it pointed to the primary name server and not an arbitrary NS record, because the name system does not provide a reliable way of establishing the primary name server, outside of: SIDN's internal database (not available), SIDN whois records (limited to 10 retrievals per day) and the SOA MNAME record.

TABLE 3A. TOP TEN OF ACCOUNTS IN MNAME;
3B. DISTRIBUTION OF HOSTS IN MNAME;
3C. DISTRIBUTION OF TTL AMONG ZONES

A Account	Count
hostmaster	5255 (55.2%)
postmaster	2284 (24.0%)
root	419 (4.4%)
beheer	388 (4.1%)
admin	214 (2.2%)
info	168 (1.8%)
netmaster	96 (1.0%)
registry	75 (0.8%)
web	71 (0.7%)
dnsadmin	41 (0.4%)

B Host	Count
name server	6679 (70.1%)
zone	1395 (14.6%)
other	1452 (15.2%)

C TTL	Count
0-12 hours	2322 (24.4%)
12-24 hours	6872 (72.1%)
24-36 hours	17 (0.2%)
36-48 hours	278 (2.9%)
48- hours	37 (0.4%)

- RNAME: Table 3A lists the top ten of account names chosen in the MNAME record, combined with the host reference in Table 3B. 70.1% of all contact addresses are located in the same domain hierarchy as one of the zone's NS records (for example RNAME hostmaster@provider.nl and NS records ns1.provider.nl). 14.6% of the addresses are situated at the zone itself. Only 15.2% of the addresses are located elsewhere (though possibly still handled by the same name- and mail servers): though, based on considerations discussed before, this is part of the target scenario. The other part of the target is a dedicated role account: Table 3A shows that the majority of the zones have such an account name, although root and info infringe clearly on this recommendation.
- Serial number: As displayed in Fig. 4 more than 80% of all zones follow the preferred numbering sequence (simpledate2). About 6% uses the unixtime (seconds

1 Virtually all name server software for use in authoritative, public name servers use this record for synchronization purposes. Different implementations, e.g. as used in the Microsoft Active Directory system, are not covered in this survey.

since the epoch, January 1st 1970) sequence. This method also allows for debugging and enough updates. Simpledate1 (YYYYMMDDn) is not a recommendation, but still allows for 10 individual zone changes per day so both this method (1.3%) and unixtime are valid numbering sequences. The usage (1.7%) of simpledate0 (YYYYMMDD) is not recommended, because it severely limits the number of changes in a zone to one per day.

The figure also shows 5.6% uses a counter sequence (1, 2, 3) and 6.2% uses an unknown serial numbering method. We strongly discourage usage of both methods because, besides possible correctness issues, they do not allow for easy human debugging ("Has the zone administrator processed my change request?").

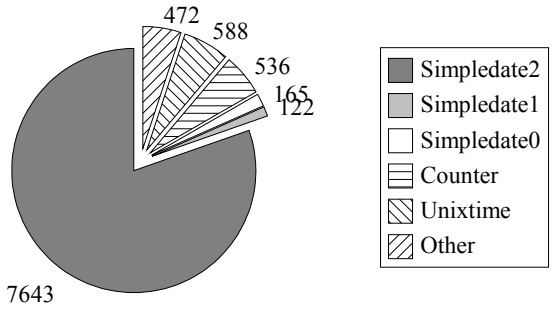

Fig. 4. Distribution of serial number sequences

- Refresh, retry, expiry: For 35 zones (0.37%) the general inequality does not hold. This can (and should for software following standards[2]) cause problems with replication: whenever a server is unreachable during refresh.

  A rather large number of domains, 330 zones (3.46%), does not pass the logical inequality. Most of these have a retry larger than their refresh, meaning that in the event of a failure during replication, a retry will be scheduled later than a normal refresh would. Normally, the retry is a fraction of the refresh, not the other way around

  846 zones (8.88%) fail the complete set of recommendations, most specifying a lower expiry than 7 days. 103 zones specify a definitely too low value of 3 hours or less. In the event of master name server failure, the slaves can start answering not-authoritative after only three hours. Thus a malfunction in the master server at night can have very unpleasant side-effects.
- TTL: Table 3C shows 96.5% of the surveyed zones have a TTL value of one day or less, corresponding to our recommendation and somewhat departing from the formal recommendations. Most probably many administrators (and/or clients) are not prepared to plan zone updates several days in advance. A worthy mention is the single occurrence minimum TTL of 60 weeks.

E. MX Record

An MX record in a zone defines a mail exchanger for the domain: a server capable of receiving mail for the domain. A zone can have multiple mail servers, and thus multiple MX records. Using the priority field in a MX record, a zone administrator can define relationships between MX servers (e.g. primary server, shared secondary, tertiary).

When a domain has MX records defined, a mail server should connect to the A record (if present), for backward compatibility reasons. Domains having no MX records and no A record for the zone are unable to receive email and are in violation of SIDN regulation 2b, as well as best practices [11]. When a domain has only one MX/A record, mail will be queued on the remote site when the mail server is down. For this reason, load-balancing, redundancy and performance many sites define more than one MX record, possibly pointing at an ISP provided server destined to be used only in case of emergencies.

Even though it is possible to specify priorities of mail servers, limiting the amount of mail reaching some servers, adding MX records adds these servers to the pools and adds to the complexity and reliability of the system (example: if any of the mail servers reject a mail with error code 550 or 553 due to the domain name or user name not known to them, other MXs may not be tried). Therefore, we make the same recommendation as we did when reviewing NS records: use two, or maybe three, but not more, strategically chosen mail servers (e.g. primary site, backup site, ISP) because for most organizations we assume this will be the point where reliability can only go down instead of up by adding more MX records. Unlike the optimal number of NS records, the relation between reliability and number of MX records can not be proven simply in this work. So instead, we present two considerations to establish plausibility for the theory that a maximum number of three MX records (equalling to three unique mail configurations, either servers or clusters) is preferable:

- Mail servers are in many management aspects similar to name servers: zones/servers for "larger" domains tend to be administrated by hand and failure to deliver an e-mail/an answer is not evident. This is also expressed in [14]: more servers also increase the likelihood that one server will be misconfigured, or malfunction, without being detected.
- Ignoring performance and focusing solely on reliability: assume that a server has a 0.001% chance to be misconfigured and rejecting e-mail, a server has a chance of 2% to be unreachable and a zone has three mail configurations (MX records), the chance of having not a single misconfigured server in a zone is 99.9970% whereas the chance at least one server being reachable is 99.9992%. When having two configurations, these values are 99.9980% and 99.9600% respectively. All servers being unreachable is a more favourable scenario than one server being misconfigured: all servers unreachable only delays, not discards mail.

Table 4 shows 206 zones having no mail servers at all and 141 having no MX records, but one or more A records. The most common configuration is one mail exchanger, closely followed by two. Almost ten percent of the surveyed zones has four or more mail servers: a risk to reliability.

TABLE 4. DISTRIBUTION OF MAIL SERVER COUNT AMONG ZONES

Number of mail servers	Number of zones
0 MX, 0 A	206
0 MX, 1+ A	141
1 MX	3980
2 MX	3386

2 In this section conclusions are based on [10], [11]; vendor or version specific implementations are not taken into account.

Number of mail servers	Number of zones
3 MX	937
4 MX	641
5 MX	300
6 MX	8
7 MX	3

F. Properties of Servers

Besides the data in the zone, server properties also influence DNS reliability and performance. Some properties affecting these factors are physical link type (e.g. redundancy, speed), and geographical location (extensively covered in [1]). This section focuses on two one-dimensional server properties. These properties have been retrieved along with other data in this survey simply by querying the name server.

- Name server software
  Most of the commonly used name server software allows the retrieval of the software name and version by querying the version.bind zone in class chaos for a TXT RR.
  Obviously a set of heterogeneous, supported name server software versions serves reliability and performance best.
- Recursion policy
  If a name server allows recursive queries then it can be used as a caching server by any client. Recursive queries (Fig. 1) drain server resources and can lead to cache pollution [13] and should be answered by a separate server.
  A name server has been queried by testing whether it returns an address when asked for www.example.com. This assumes no authorised user (e.g. local user) has recursively looked up this domain for the past two days (due to minimum TTL).

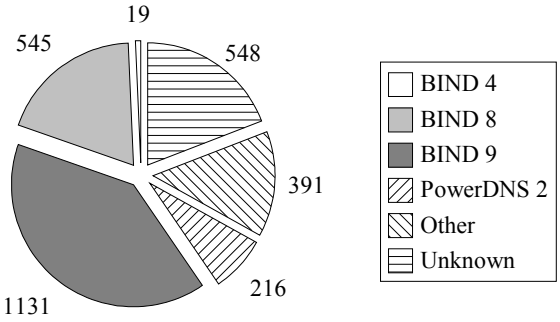

Fig. 5. Distribution of software among name servers

All servers generating at least one authoritative answer (2,850) have been tested on these properties. Fig. 5 details the distribution of software among name servers. A majority (58,8%) of these run a current version of ISC BIND (8 & 9), 7.6% of sampled servers run the Dutch newcomer PowerDNS and almost 20% runs an unidentified name server (not responding to the version.bind attribute, presumably DJBDNS and Microsoft DNS). An additional 14% of name servers answer in an ambiguous way: BIND allows for redefinition of the version.bind RR to any value. The examined values "Go screw someone else!" or "request logged and reported to abuse!" can only lead to provocation. Entering contact information for the DNS administrator or the class of software (BIND 8) seems like a better redefinition, if any.

After mapping the responses of the name servers to a class of software (BIND 4, PowerDNS 2 etc.), the number of different classes of name server software per zone has been established. 91.82% of zones is served by only one class of software, 8.11% is served by two different classes and 6 domains are served by three different classes of which only the domain umbs.nl (Universiteit Maastricht Business School) is verifiably served by three different software classes (BIND 4, 8 and 9).

Recursion is enabled on 1934 (67.86%) servers and disabled on the rest. Even if recursive queries are to be allowed by the servers, e.g. because they serve local or dial-up users, then access restrictions should be imposed and all other hosts should not be permitted to ask recursive queries.

To allow for maximum reliability and performance, the following recommendations can be made:

- Either do not redefine the version.bind RR, or redefine the record to something sensible.
- Employ multiple classes of DNS software to prevent a bug in one name server software from disabling all name servers (on purpose or not) for a specific zone.
- Disable or restrict access to recursion.

VI. Conclusion

In this paper we have investigated 10,000 DNS zones for correctness. The following calculation shows the breakdown of encountered error conditions, not counting any zone more than once:

Starting set	10,000	zones
One or more lame delegations	-802	zones
Four or more NS records	-54	zones
Different leafs different NS records	-5	zones
Leafs omit NS records	-73	zones
Registerable NS record	-242	zones
Different leafs different SOA records	-56	zones
MNAME not in NS records	-9	zones
R, R, E logical equation error	-155	zones
Four or more MX records	-919	zones
	======	
No issues	7,685	zones

It turns out that, of the 10,000 zones, only 7,685 do not have any issues. If we neglect the less problematic “Four or more MX records” error, 14% of all tested .NL domain names does not fulfil all configuration requirements, recommendations or best practices. As a result, the reliability and performance of DNS is not as good as it should be. Since the .NL domain is regarded as one of the best managed zones worldwide ([1] Fig. 2a), we expect these errors to show up in most other top level domain zones as well.

One of the most interesting conclusions of this research is that a higher number of name servers does not automatically lead to an increase in reliability. In fact the opposite is true; if the number of name servers increases beyond a certain point reliability and performance usually decrease (section 5.2).

Further research may be needed to find even more and different error conditions in more Resource Records (RRs). Our findings should also be cross-referenced and verified with other top level domains.

Compared to previous research by Pappas et al. [1] [2], our research is both broader in the number and nature of checks, but also more narrow since we have limited our research to a single domain. Because of these characteristics, we have found more error conditions and also more zones with errors. This survey agrees to the sad state of DNS as presented in commercial research by Credentia [16] and Men & Mice [17].

The results of this research should be used to increase the reliability and performance of the name system, raising it to the levels envisioned at the design time of this redundant, distributed system. Such higher quality of DNS service can be achieved by:

- Education: DNS administrators can gain insight in common errors and misconfigurations.
- Prevention: The .NL regulatory body can adopt and enforce new technical regulations for domain registrations. These regulations could be enforces by an autonomous program surveying the .NL zones, contacting the responsible account about errors and disabling the leaf if necessary.
- Research: The results of this research are, possibly after further investigation, equally applicable to other TLDs.

References

[1] Vasileios Pappas, Zhiguo Xu, Songwu Lu, Daniel Massey, Andreas Terzis, Lixia Zhang "Impact of configuration errors on DNS robustness" in *Proceedings of the 2004 conference on Applications, technologies, architectures, and protocols for computer communications,* New York: ACM Press, 2004, pp. 319-330

[2] Vasileios Pappas, Patrik Fältström, Daniel Massey, Lixia Zhang: "Distributed DNS troubleshooting" in *Proceedings of the ACM SIGCOMM workshop on Network troubleshooting*, New York: ACM Press, 2004, pp. 265-270

[3] Ward van Wanrooij, (2004, February), Juridische aspecten van een domeinnaam door het oog van een informaticastudent, [Online], Available: http://www.ward.nu/computer/files/juraspdom.pdf

[4] SIDN, "SIDN – Home", 2 October 2004,. http://www.sidn.nl/

[5] SIDN, "Bijlage 1 bij het reglement voor registratie: Technische eisen. created 29 January 2003" in Dutch, 2 October 2004, http://www.sidn.nl/sidn/flat/Algemeen/Voorschriften/Reglement_voor_registratie_van_.nl-domeinnamen/Bijlage_1_bij_het_reglement_voor_registratie_Technische_eisen/index.html

[6] Freudenthal Instituut, "WisFaq" (in Dutch), 14 November 2004, http://www.wisfaq.nl/showrecord3.asp?id=11725

[7] Internet Software Consortium, "ISC BIND 9.3", 2 October 2004, http://www.isc.org/index.pl?/sw/bind/bind9.3.php

[8] Cricket Liu, Paul Albitz:, "DNS and BIND" third edition, Sebastopol: O'Reilly, 1998

[9] P. Mockapetris, (1987, November), RFC 1034 Domain names concepts and facilities, [Online], Available: http://www.rfc-editor.org/rfc/rfc1034.txt,

[10] P. Mockapetris, (1987, November), RFC 1035 Domain names implementation and specification, [Online], Available: http://www.rfc-editor.org/rfc/rfc1035.txt

[11] D. Barr, (1996, February), RFC 1912 Common DNS Operational and Configuration Errors, [Online], Available: http://www.rfc-editor.org/rfc/rfc1912.txt

[12] R. Elz, R. Bush, (1996, August), RFC 1982 Serial Number Arithmetic, [Online], Available: http://www.rfc-editor.org/rfc/rfc1982.txt

[13] B. Manning, P. Vixie (1996, October), RFC 2010 Operational Criteria for Root Name Servers, [Online], Available: http://www.rfc-editor.org/rfc/rfc2010.txt,

[14] R. Elz, R. Bush, S. Bradner, M. Patton, (1997, July), RFC 2182 Selection and Operation of Secondary DNS Servers, [Online], Available: http://www.rfc-editor.org/rfc/ rfc2182.txt

[15] M. Andrews, (1998, March), RFC 2308 Negative Caching of DNS Queries (DNS NCACHE), [Online], Available: http://www.rfc-editor.org/rfc/rfc2308.txt

[16] Credentia, (2003, October), ccTLD Name Server Delegation Report Card, [Online], Available: http://www.credentia.cc/research/dns/cctlds/report-2003-Oct.html

[17] Men & Mice, (2003, February), Domain Health Survey for .COM, [Online], Available: http://www.menandmice.com/6000/61_recent_survey.html