INTERNATIONAL TELECOMMUNICATION UNION

TELECOMMUNICATION STANDARDIZATION SECTOR

STUDY PERIOD 2001-2004

ccTLD Doc 46

Original: English

Workshop on Member States' experiences with ccTLD

Geneva, 3-4 March 2003

DOCUMENT FOR ccTLD WORKSHOP

Source: Nominum

Title: A survey of the DNS infrastructure for ccTLDs

INTRODUCTION

The robustness and reliability of the Domain Name System depends on a simple rule: there should be no single point of failure. In addition, the Internet Engineering Task Force (IETF) has produced recommendations on the operation and configuration of name servers to enable operators and technicians to provide a stable DNS infrastructure. The objective of this document is to analyse how the DNS infrastructure for the country code top-level domains (ccTLDs) meets these requirements.

Put simply, the goal of a DNS administrator is to ensure that no matter what fails, DNS lookups continue to work. Without a working DNS service the internet would not work even if other components like the networking infrastructure and servers were operating normally. The reason for this is most users think in terms of easy to use names, like www.itu.int, rather than hard to remember IP addresses such as 156.106.192.163, the Internet address of the ITU web site. [This problem gets even harder for IPv6 addresses that are 4 times as big as currently used IPv4 addresses.] Applications generally use the DNS to translate from domain names to IP addresses. So if DNS breaks, pretty much everything breaks, even if things like routers and web servers are working correctly.

The DNS is effectively the glue that binds applications and services to the infrastructure of data links and routers, essentially translating domain names to network addresses. Mail and web servers could be working fine, but without that name to address translation, clients would be unable to connect to servers. That would mean web servers would appear to be unavailable or email would go undelivered even though the servers were working correctly. If the DNS lookups didn't work nothing would be unable to find these servers. Likewise, routers and switches may be able to send packets and the network links might be able to transmit them but unless the name to address translation works computers don't know which addresses should be in the destination fields of the packets they send.

Many organisations, notably e-commerce companies, spend large amounts of money on making sure their web servers are highly available. However unless the organisation gives the same level of attention to their name servers, all that investment could be for nothing. A web server hosted on high availability server farm in a data centre with backup power supplies is useless if the organisation's name servers are not as well provisioned.

Contact: Jim Reid
Nominum UK
Email Jim.Reid@nominum.com

This paper surveys the DNS infrastructure for the ccTLDs. With more and more of the world's commnications using the internet, DNS becomes even more critical. The robustness of national DNS infrastructure is becoming a very important subject for governments, regulators and businesses.

METHODOLOGY

The raw data for the survey was collected over the weekend of Feb 14-17th 2003. The delegation for each ccTLD was checked from the root servers, giving a list of the name servers for each ccTLD. Those servers were in turn queried to find out more about the ccTLD(s) that they served. The survey checked all 243 ccTLDs.

A packet tracing tool was then used to find the path packets took to reach those name servers from a computer based at the London Internet Exchange, LINX. This is a very important communications hub for the internet with connectivity to most carriers and major Internet Service Providers. The object of this was to find out if there was an excessive concentration of ccTLD name servers at one location or if a ccTLD's servers were in one place, creating a single point of failure.

Each name server was then checked for a number of possible errors. A total of 787 name server names were identified. These resolved to a 756 unique IPv4 and 23 unique IPv6 addresses. 18 of the names for the name servers did not resolve to an IP address. These numbers do not add up to 787 because some servers have more than 1 IP address: for example a multi-homed server or a dual-stack IPv4 and IPv6 system.

A fingerprinting tool developed by Nominum was used to identify the name server software used by all these servers. This sent carefully constructed queries that would generate distinguishing responses from name server implementations.

The results were analysed and compiled into this report.

DELEGATION ERRORS

Unresolvable names

One of the most startling initial findings was that the names of 18 ccTLD servers did not exist in the DNS! In other words, approximately 2.5% of the ccTLD name servers had names that could not be looked up. Essentially the ccTLD hostmaster is advertising the presence of a name server that they should know does not exist. Since those names could not be looked up, those "phantom" name servers could not be queried, assuming they even exist under some other name. This is a glaring error which simply should not happen. It may be understandable when someone new to the DNS sets up a name server and registers a domain name for the first time. However it should never occur in ccTLDs because these important domains should not be operated by naive administrators and any changes to the ccTLD setup should be carefully checked.

Although this is a serious and fundamental error, it is fortunately not a mission-critical one. All of the ccTLDs that advertise these phantom name servers also advertise name servers that do exist and serve the ccTLD. So service for the ccTLD is still available although it will be impaired by the existence of these phantom servers that cannot be queried. The DNS will treat these as unresponsive name servers. The DNS protocol has mechanisms for handling unresponsive servers, something that contributes to the overall robustness of the DNS. However keeping track of these unresponsive servers means more work for everyone else's name servers, extra overheads and longer lookup times.

Basic Delegation Errors

Sadly, one ccTLD is in a very bad way. It appears to have 4 name servers. Only one of them is correctly defined. The other three are listed as dotted decimal strings -- presumably their IP addresses -- rather than valid host names which the DNS requires. This ccTLD has a major single point of failure. If its only correctly named server becomes unavailable, the ccTLD will disappear from the internet.

Another 10 of the name servers do not have host names as required by the DNS protocol. Instead they are present as CNAME records (nicknames). This is illegal, though most name servers tolerate this error. However using CNAMEs instead of hostnames cancause name resolution problems. Again, this is a basic administrator error which should not occur in an important DNS zone like a ccTLD.

Parent/child mismatches

The process of delegation -- creating a new administrative entity in the DNS -- requires a little co-ordination. The parent making the delegation has to agree with the child receiving the delegation on the names of the servers for the new domain. In the case of the ccTLDs, the parent is the root zone and the children are the ccTLDs themselves. This too is a common area where mistakes are made by naive administrators.

In theory, the set of name servers for a domain should be identical in the parent and child. In practice, things will work, though perhaps non-optimally as long as there is an overlap between the two. For experienced administrators and important domains, the parent and child should have an identical set of name servers.

It would be expected that there would be no mismatches between the root and a ccTLD over the ccTLD's name servers. The reality is very different. 155 of the ccTLDs do not have the same NS records for the TLD as in the root, their parent zone. In other words, 64% of the ccTLDs have mismatched delegation information between themselves and the root.

There is no excuse for this. In a well-maintained zone, mismatched delegations simply would not be allowed to happen. These errors generally result from DNS administator error. However for ccTLDs there is a mitigating factor. Changes to a ccTLD delegation involve updating the root zone and this can only be done after the ccTLD makes a request to ICANN. Many ccTLDs are in dispute with ICANN which has created a stalemate. ICANN won't alter ccTLD delegations in the root unless the ccTLD signs a contract with them, something most ccTLDs are reluctant or unable to do. Although this impasse could explain why there are so many mismatches, it does not present a valid excuse in most cases. The ccTLD should not have changed their NS records unless that change was reflected in the root zone.

Fortunately, most of these mismatches are not serious. The ccTLD's set of name servers generally turns out to be a superset of those listed in the root zone delegation for the TLD. This is harmless from an operational perspective. Even so, this is damaging because it suggests that the internet root and ccTLDs cannot keep this simple and fundamental information properly synchronised. The implications of this could be far-reaching.

Lame delegations

A lame delegation in the DNS is a serious problem. This occurs when a server that is supposed to answer authoritatively for some zone does not. This is always caused by administrative error. The problem can be in one of two places. The first is when the zone owner includes an NS record for the zone that points at a name server which is not answering authoritatively for that zone. This should never happen. The zone administrator should not add or change an NS record unless they know that the new name server is answering correctly. The second possibility is that the owner of that name server has been unable to configure it properly or a substantial connectivity problem has prevented the server from getting an up-to-date copy of the zone from its master (primary) name server.

Lame delegations can have unpredictable results. Sometimes DNS lookups succeed, sometimes they fail. It all depends on whether the lookup goes to the lame name server or not. Lame delegations are very common on the internet, usually for small, unimportant zones maintained by naive users. There are a number of tools for checking and reporting lame delegations: most name servers log these errors whenever they find them. This problem should never arise with a correctly operated delgation in the DNS.

Unfortunately 43 ccTLDs had 1 or more lame servers. In the case of 2 ccTLDs ALL of their name servers were lame. A further 8 ccTLDs had more than half of the servers lame. Clearly, some ccTLDs are not taking proper care of their delegations. Put simply a zone should not have NS records pointing at non-authoritative servers.

Lame delegations for ccTLs would not occur provided this simple rule was followed the administrators of these important domains. This is a problem that needs attention, perhaps a ccTLDs, ICANN and other interested parties jointly developing and enforcing a code of conduct.

RECURSIVE SERVERS

Name servers for a ccTLD should be configured to be authoritative-only. Since they should only be queried by other name servers, there should be no reason for a ccTLD server to process recursive queries. These lookups should of course be handled by the name servers that query the ccTLD name servers.

Most name server software can be configured to operate in this authoritative-only mode. In fact, some implementations do not provide any resolving capability at all: NSD, UltraDNS, ANS & djbdns for example. These can also offer improved performance and greater security since the software is simpler. However the most commonly used name server software, BIND, combines a resolver, which handles recursive queries, and authoritative components. By default both of these components are enabled.

By enabling recursion, a name server can be made to query other name servers as it tries to resolve the names it has been asked to lookup. This can cause cache poisoning: when the answering server tells lies. The querying name server will believe those bogus answers and install them in its cache. Subsequent lookups can be given those false answers from the cache. This is a well-known limitation of the DNS because it is not a secure protocol.

Name servers for important DNS infrastructure should not provide recursive service for two reasons. The first is the issue of cache pollution explained above. When a ccTLD server's cache gets poisoned, the bogus data can be propagated to anyone querying that server. Since ccTLD servers get queried often from all over the internet, any bogus data can be spread widely. This would make them an ideal vehicle to cause significant cache pollution throughout the internet. The second reason is that recursive queries make a name server do a lot of work. This could cause a denial of service attack since there is no reason for ccTLD servers to process recursive queries. They should only be queried by servers that are handling recursive requests.

Surprisingly, almost half of the Internet's ccTLD name servers have recursion enabled. 371 of these servers will process recursive queries. This means 47% of the ccTLD servers are needlessly exposed to cache poisoning, and can spread any bogus data they get from queying other name servers. Furthermore those 371 servers could easily be overwhelmed by recursive queries, preventing them from answering genuine queries for the ccTLDs they serve. This needs to be fixed.

OPEN ZONE TRANSFERS

Zone transfers are the usual mechanism in the DNS to propagate zones from the master (primary) server to its slaves (secondaries). As its name suggests, this mechanism involves taking a complete copy of the zone.

Although this is an everyday part of the DNS protocol, ccTLDs should impose restrictions on who is permitted to perform transfers of this zone data. There are several reasons for this. First of all, zone transfers can take a long time because ccTLD zones can be large.

This can place an excessive load on the server if there are no restrictions on who can perform a zone transfer. Large numbers of zone transfer requests could overwhelm a name server or the available bandwidth on its network link. A second reason is that access to a complete copy of the zone can provide useful information to cybersquatters: people who make speculative domain registrations. The zone file will tell them what domain names are and are not registered.

Finally, there are questions of privacy and data protection. Many ccTLDs have refused to provide ICANN with copies of their zone file for this reason.

Out of 243 ccTLDs, 140 permit zone transfers. This makes the dispute with ICANN hard to understand. ICANN can simply make successful zone transfer requests for 58% of the ccTLDs without asking for permission from the zone owner.

Within the ccTLDs that permit zone transfers, the rule or policy is usually inconsistent. Some of the name servers for a ccTLD allow zone transfers, others don't. This is probably explained by the servers being under different administrative control and the ccTLD not establishing and enforcing a consistent policy with all of their slave (secondary) servers.

The consequences of this finding are disturbing. Controls on zone transfers tend not to be used by most ccTLDs. And even when they are enforced, it is usually on a per-server rather than a per-ccTLD basis. Given the potential for abuse from open zone transfers, this seems like a vulnerability waiting to be exploited.

FINGERPRINTING

A fingerprinting tool developed by Nominum was used to identify the software used by ccTLD name servers. This sends queries to the servers which trigger different responses from different name server implementations. The following table gives a breakdown of the software in use by ccTLD name servers:

Implementation	**Number of servers**	**Percentage**
BIND8	364	47%
BIND9	268	34%
BIND4	42	5%
UltraDNS	10	1.3%
Atlas	10	1.3%
Microsoft	5	0.6%
NSD	4	0.5%
PowerDNS	4	0.5%
djbdns	2	0.3%

The remaining servers could not be identified with the tool. These are probably a mixture of very old software and heavily customized versions of BIND 4 or 8.

Clearly, BIND predominates. However there is some diversity as BIND8 and BIND9 use different code bases. They do not share any code, though parts of the design of BIND9 were influenced by the need to keep administrative backwards compatibility with BIND8. BIND 4 is dead and the Internet Software Consortium (ISC) strongly recommend that it does not get used. Even so, it has a non-trivial installed code base in the ccTLD name servers. BIND4 and BIND8 share the same code base.

The high usage of BIND8 is worrying. This software is approaching end of life. It is not actively worked on, other than patches for security problems. Unfortunately, there have been a number of them as can be seen from the number of CERT advisories about security weaknesses in that software. Information about security holes in BIND can be found on the ISC's web site at:

http://www.isc.org/products/BIND/bind-security.html

Although there are a handful of other name server implementations, between them they only account for 4-5% of the ccTLD DNS infrastructure. This level of monoculture is disappointing, though hardly unsurprising. BIND has been the de facto DNS standard for many years and other surveys have shown similar results. In fairness, some of these other implementations are fairly new. It will take time for them to gain a foothold in the market.

Diversity of DNS software should be very important to a ccTLD. If all its name servers run the same software, a bug or security vulnerability could affect all of them. This could mean denial of service or even the remote execution of arbitrary code. With diversity of implementation, a weakness in one code base should not affect another. Therefore at least some of the ccTLD's name servers would continue to work in the event of a catastrophic bug or design error in one DNS implementation.

A further concern is that many ccTLDs are running old versions of BIND4 and BIND8 that are known to have security holes. The ISC recommends that BIND version 9.2.1 should be used, though if BIND 4 or BIND8 must remain, versions 8.3.4 or 8.2.7 or 4.9.11 should be used. At least 15 of the servers running BIND4 are not running version 4.9.11.Of the servers running BIND8, 144 may be running old versions that are vulnerable to security problems. The fingerprinting tool does not yet identify specific versions of BIND4 or BIND8, so the only way to identify what these servers would be to send queries which exploited the vulnerabilities. This was not done for the obvious reasons.

Analysis of the name server implementations in use by each ccTLD was also carried out. Most ccTLDs had at least 2 code bases for their DNS software: some had more than that. However 42 ccTLDs had a monoculture in their DNS software. The table below gives the overall breakdown. For the purpose of this exercise, BIND4 and BIND8 name servers are counted as one code base since the underlying software and its design is derived from the same source.

Number of Code Bases	**Number of ccTLDs**
1	42
2	97
3	88
4	12
5	4

The ccTLDs dependent on just one code base run a variety of DNS software. One is solely dependent on a managed DNS service from UltraDNS, 2 ccTLDs rely exclusively on PowerDNS and 1 uses unidentified DNS software. 22 ccTLDs are depend on the BIND8 code base and 16 are only using BIND9.

LOCATION ANALYSIS

Checking the paths to the ccTLD name servers turned out to be inconclusive because many ISPs blocked or filtered out the ICMP packets used to generate the traces. Even so, this topic is worth further study. Although tools are not readily available, it should be possible to determine which ISP "owns" a particular IP address and the Autonomous System (AS) it belongs to. [An Autonomus System Number (ASN) can be thought of as a unique number used to identify which ranges of IP addresses are reached via particular routers, ISPs or network links.] ASNs are used by the core routing protocols to find out which network providers should be used to reach particular ranges of IP addresses. If all the addresses for some resource are reached via one ASN, that becomes a single point of failure. If the ASN is lost or blocked, the IP address prefixes behind that ASN could be unreachable because routers won't know how to route packets to IP addresses within those prefixes.

Some simple analysis was done by hand. The IP addresses for each ccTLD were compared and a determination made about which address prefixes they were likely to be in. If the high-end bits of two IP addresses were the same, the chances are they would both be in a single address prefix used for routing. This was an arbitrary process and may not be reliable. Judgements were made rather than in-depth analysis of network address allocations and routing policies.

Most ccTLDs use name servers with IP addresses that are located in diverse address prefixes. So if a routing problem means that one of these prefixes is temporarily lost, it does not present a serious problem. It may not be possible to reach a name server in the "lost" address prefix, but the other servers for the ccTLD will still be accessible.

However for 36 of the ccTLDs, at least 50% of their name server's IP addresses appear to be in one address prefix. Though it was not necessarily the same prefix for all of these ccTLDs. If the core routers lost the information about those prefixes, this could have a serious impact on the availability of DNS service for those ccTLDs. Put simply, a routing problem could mean half or more of the name servers for those ccTLDs would be unreachable even if the servers were working correctly.

A further 13 ccTLDs have an even bigger vulnerability. All of the name servers for each one of these ccTLDS share the same address prefix. Again, these 13 ccTLDs don't share a single, common prefix. Each of them has all of its name servers in a single address prefix. In most cases all of the ccTLD's name servers are in a single /24 network: ie the most significant 24 bits of the IP address are identical. If information about those address prefixes get lost or dropped, the routes for these networks could disappear from the core internet routers. That would effectively mean these ccTLDs could disappear from the internet as their name servers could not be reached. Nothing would know how to route traffic to them.

This is somewhat startling. When ICANN recently created the new gTLDs, it insisted that the name servers for these gTLDs were located in different address prefixes assigned to at least 2 Autonomous System Numbers. The rationale was that if there was a routing problem with one ASN, there would still be another that was announcing routes to at least some of the gTLD's name servers. It seems strange that this requirement is not required or even recommended for ccTLDs.

FURTHER AREAS OF CONCERN

The information described above is not secret. It's readily available to anyone who has some understanding of how the DNS works and has access to tools that are usually provided with the system software on any computer. However some areas of ccTLD robustness are not public. This may be because they are subject to contracts or Memos of Understanding (MoUs) between the relevant parties. However it is more likely that they are not public because these things are not documented in any way, let alone by a contract between say a registry and some service provider.

Monitoring

The first area that ccTLD operators, governments and regulators should consider is monitoring of their national DNS infrastructure. This would allow the servers to be monitored for availability, response times and correctness. Monitoring should be done from a number of places on the internet, ideally at locations at the edge of the network. The root servers are continuously checked in this way by Verisign and others. Monitoring from the periphery of the network is best since this is closest to the environment end users experience. Connectivity and routing problems are more visible there than at internet exchanges.

Service Level Agreements

These should be a service level agreement (SLA) for any important zone in the DNS. These should document the procedures and requirements between the zone owner and the organisations which provide slave (secondary) service for the zone. There should of course be multiple organisations providing slave service for the zone to prevent a single supplier being a single point of failure. Other advantages could include competition on service levels and costs.

These SLAs should document details like the location of slave name servers and information about the platform used: hardware, operating system and name server software. They should also cover operational matters such as availability guarantees, response times, zone propagation times, maintenance windows, server configuration, monitoring, problem reporting and escalation procedures and so on.

Few ccTLDs have SLAs with their slave service providers, though this situation is beginning to change. A number of companies are offering managed DNS services and an SLA is usually a central component of their service offering. For many ccTLDs, ad-hoc and often undocumented arrangements are in place. It is quite common to find that ccTLDs have arranged mutual slave service on the basis of a handshake. Sometimes these arrangements were put in place by DNS administrators who later left the ccTLD.

The absence of SLAs and adequate problem reporting should be a major concern. There may also be legal issues here: for example liability considerations in event of a service outage or name server that hands out incorrect data. At a strategic level, this could also be seen as a matter of national interest. Governments and regulators may want to be certain that their country's DNS infrastructure is adequately protected and there is a formal framework for its operation and management. This could be an area for collaboration between governments, regulators and ccTLD operators.

Distributed Denial of Service Attacks

The prevalence of distributed denial of service (DDoS) attacks has increased on the Internet in recent years. These attacks can be hard to defend against. In a conventional denial of service attack, the objective is to prevent some service from meeting the expectations of its users. In the case of a network server, this is typically done by swamping the server with traffic. The volume of data or connections overwhelms the server and either causes it to crash or be starved of resources to service requests from genuine clients. Sometimes, the servers are not attacked directly. Instead the routers and network links are saturated with traffic, preventing the packets for genuine traffic to get through and thus creating a denial of service. In a DDoS attack, the source of the traffic is distributed. It comes from a variety of locations rather than a single source.

The root servers are frequently subjected to DDoS attacks. The most recent example occurred in November 2002. Although this attack prevented access to 8 of the 13 root servers, its impact was minimal. The main reason for that was the robustness of the DNS protocol. Caching meant few name servers needed to contact a root server -- and few of them were unable to reach one -- while the attack was under way. In addition, the root server operators had processes in place to defend against the attack, further minimising its impact.

However is unlikely that most ccTLDs have comparable defences against DDoS attacks. If one of these attacks was directed at a ccTLD's DNS infrastructure, it could well be successful. This could mean a ccTLD disappearing from the internet because its name servers could not be reached. The consequences of this do not bear thinking about. So far, DDoS attacks have tended to result from accidental or malicious acts by people who are technically naive. These have been straightforward to defend against. An attack by a well-motivated and sophisticated adversary could be far-reaching. This is a very real danger for ccTLDs since few, if any, have adequate defences in place to detect and prevent DDoS attacks. In fact a well designed DDoS attack against the DNS would be almost impossible to distinguish from genuine DNS queries. After all name servers are expected to get lots of requests for different names from a large and random number of addresses on the internet.

CONCLUSIONS

It is suprising that many ccTLDs suffer from fundamental DNS errors. There are too many basic mistakes that would normally only be expected from inexperienced DNS administrators. This is all the more surprising given that there are two documents from the Internet Engineering Task Force (IETF) advising on how to set up and operate important name servers. These are RFC2870 -- Root Name Server Operational Requirements -- and RFC2182 -- Selection and Operation of Secondary DNS Servers. These documents have been available for years and much of the information was well-known before these RFCs were published. It seems odd that so many ccTLDs do not seem to have followed the advice in these documents.

Aside from the technical errors and bad practices, it is likely that policy and contractual issues such as service level guarantees are also less than optimal. These matters should be addressed by all the interested parties.

The number of ccTLDs using old DNS software with known security flaws is disappointing. A further concern is the lack of diversity of DNS software that's in use. These problems should be urgently addressed. There is no excuse for running old, buggy software. This is the root cause of many security problems. So far attackers have tended not to spend much effort on name servers. The main reason for that is there are easier targets -- web servers, mail systems, etc. DNS skills are also needed and these are comparatively rare. However it cannot be assumed that these conditions will continue to hold. Eventually the so-called script kiddies will have toolkits to compromise name server security.

The use and dependence on the Internet will continue to grow. It will become even more important that critical Internet infrastructure is operated carefully. For ccTLDs, this may mean that the robustness and resilience of their name servers is given more attention. Denial of service attacks on ccTLD name servers are likely to become more common and more destructive. Nations should analyse their susceptibility to these attacks, prepare an impact analysis and deploy effective safeguards for this important resource. It is hoped that this paper sheds some light on the steps that should be taken.