ONGERUBRICEERD

Technical Sciences
Brassersplein 2
P.O. Box 5050
2600 GB Delft
The Netherlands

www.tno.nl

T +31 88 866 70 00
F +31 88 866 70 57
infodesk@tno.nl

TNO report

Report number: 35550

DNS Services, alternative ways of using DNS infrastructures

Date
7 October 2011

Our reference
-

E-mail
oskar.vandeventer@tno.nl

Direct dialling
+31 88 866 70 78

Direct fax
+31(0)152857349

Date	16 September 2011
Author(s)	Brook Abegaz
Supervisors	Dr.ir. M. Oskar van Deventer, ir. Bart Gijsen
Project name	Future DNS
Project number	035.33957/01.01
Key Words	DNS Domain Name System Service Resolver

Summary

This report is the result of a three-month intership at TNO. The goal of the project was to achieve better understanding in ways that DNS technologies and the global DNS infrastructure is used and may be used.

The report starts with a survey of DNS services at the client, resolver and server level. A benchmark test was performed on TNO's DNS servers. Finally, a DNS-based advertisement-blocking prototype was built.

Attached to this reports are the slides of the final presentation by the author.

Contents

1 Introduction

Whenever someone uses the Internet, it is quite likely that he/she is also using the Domain Name System at the same time. When a user browses the internet using a domain name like “tno.nl” or “tudelft.nl”, it is the Domain Name System that changes the user-friendly domain name into an Internet Protocol address like 134.221.1.64 or 131.180.77.102 that computers use to identify each other on the network, and viceversa.

The Domain Name System is a hierarchical distributed naming system for computers, services or any resource connected to the Internet. According to RFC 1035, RFC 1123 and RFC 2181, which define the rules for assigning domain names to groups of internet services, it is possible to assign domain names to groups of internet resources and users in a meaningful manner independent of each entity’s physical location. [1]

The Domain Name System (DNS) is also a basis for a number of services and infrastructure that rely on the DNS for their functioning. Such services vary from services like E-mail, Web browsing and File transfer to value added services like Dynamic DNS, User and Infrastructure ENUM, Blacklisting, Parental control, Software version updating, Redirection, Context aware naming and Performance improvement.

When comparing DNS based solutions to other solutions on the internet, for example to Web based solutions, there are a number of difference considered in terms of speed, cost, complexity, scalability, reliability and global access. As Reference [51] indicates, since the Domain Name System is designed as a globally distributed system without a single point of failure, it is found to be highly scalable and reliable. In most cases, DNS based solutions are less expensive and faster than Web based solutions. For example, in cases where NAPTR records are used, the small size of NAPTR records allows them to be cheaply transmitted over a data network. In such cases, User Datagram Protocol is used to transmit NAPTR records which makes the transmission faster with little requirement on reliability.

In comparison, most Web based services are slower because webpages are large and transmitted over HTTP connections. The big size of webpages also means the solutions are expensive. In addition, publishing webpages is complex and time-consuming and sometimes prone to single points of failure. Therefore this serves as an introductory reasoning why solutions based on DNS are a focus of interest at the moment.

The role of the DNS in the internet is increasing from the usual internet browsing to the relatively newer value added services, and a number of services and infrastructure can be built on or inside the Domain Name System. However, a proper inventory or classification of existing DNS services and infrastructure as currently used in the internet has not been available in literature. These, therefore, are some of the motivations of the DNS services/infrastructure internship.

As a result, the following parts of this report comprise of an inventory of DNS services and infrastructure that have been surveyed as currently used in the internet. Based on the inventory, a proper classification of the services is presented. Moreover, a technical analysis on the performance of existing DNS resolvers is performed. Then, a practical implementation of a selected DNS services follows. Finally, a conclusion of the what has been accomplished in the internship work and a recommendation for future work is presented.

2 Inventory and Classification of DNS Services and Infrastructure

As mentioned in the introduction of this report, one of the motivations of this work is the unavailability of a proper inventory and classification of DNS Services and Infrastructure in literature. Therefore, in this internship work, DNS Services and Infrastructure as currently used in the internet have been surveyed and their inventory has been prepared.

For a proper classification of DNS services and infrastructure as currently used in the internet, one mechanism is to identify what can be regarded as “Services” and what can be regarded as “Infrastructure”. Such a classification mechanism could rely on the fact that services can be directly managed and configured by the user while infrastructure are managed and controlled by a higher level entity which could be the resolver, the service provider or the authoritative name server. This classification mechanism, however, is not free from flaw since there are some DNS based solutions like Security which could not single-handedly be classified as Services or Infrastructure.

Another way of classification of DNS services and infrastructure as currently used in the internet could make use of the existing hierarchical usage of the Domain Name System. Such a classification could identify services and infrastructure at the Client side, at the Resolver side and at the Server side. This way of classification is used in this report since it makes it relatively clearer to identify at which stages of the Domain Name System each of the services and infrastructure are implemented.

The following parts of this survey report discuss the inventory of DNS based services and infrastructure that have been surveyed. First, services and infrastructure by existing DNS resolvers is presented in 2.a. Then value added services and infrastructure that have been identified from literature is presented in 2.b. These parts also discuss about the stages where the services and infrastructure could be implemented; whether it is at the Client side, the Resolver side, or at the Server side.

2. a Services and Infrastructure by Resolvers

Under the category of already existing DNS server solutions, there are a number of Domain Name System based services on the internet. Most of these services have their own DNS Infrastructure, Control, Security and Administration and perform the task of a recursive DNS resolver. The task of a recursive DNS resolver is mainly to process Recursive Queries. [1] A *Recursive Query* is one for which the DNS server will fully answer the query (or give an error) by querying other name servers as needed. Some of these existing Recursive DNS resolvers are Open DNS, Google Public DNS, Ultra DNS and Power DNS.

OpenDNS is one of the most dominant of DNS services which is also an open source DNS Service. [2] To configure the network to use OpenDNS, the user should point its external DNS at each of their locations to OpenDNS’s two Anycast IP addresses 208.67.222.222 and 208.67.220.220. OpenDNS includes different kinds of DNS based controls some of which are *web content filtering*, *proxy/anonymizer blocking*, *blocking page bypass*, *whitelisting and blacklisting domains and redirection for a non-existing domain search.* Such services are provided along with security related *phishing protection*, *botnet protection* and *malware site protection*. Based on its globally distributed network as a DNS infrastructure, Open DNS employs *Anycast routing technology*, and *Smart Caching* to increase the speed of responses. [3]

Google's Public DNS is another one of these services, which is a freely provided closed-source DNS service announced on 3rd December 2009. [4] Google Public DNS is configured by setting the network to use the IP addresses 8.8.8.8 and 8.8.4.4 for IPv4, and 2001:4860:4860::8888 and 2001:4860:4860::8844 for IPv6 as its DNS servers. The primary targets of Google's Public DNS are *Performance* and *Security*.

For **Improved Performance** and to support high-volume input/output and caching, Google's Public DNS performs load-balancing user traffic to ensure shared caching. In addition, it has uniquely implemented *Smart Caching* to increase the speed of responses. This means that Google's Public DNS independently resolves domain names and keeps the resolutions in the cache until their time-to-live (TTL) expires, at which point they are automatically refreshed. Google also claims that their Public DNS cycle of caching and refreshing is performed *offline* in an *asynchronous* manner. However since the system is closed source, it was not possible to find out what kind of asynchronous caching algorithm they are using.

For **Security**, since DNS is vulnerable to spoofing attacks that can "poison" a name server's cache and route its users to malicious sites, Google's Public DNS adds *entropy* to requests and rate-limits client traffic as indicated in Reference [5]. Other than those important functions, Google Public DNS does not block, filter or redirect users unlike some open resolvers and ISPs.

Enhanced DNS is an outsourced DNS solution from Akamai which provides a secure and fault tolerant DNS infrastructure solution as Reference [6] indicates. Akamai has included a number of technologies like *IP Anycast*, *secured zone transfers*, *non-BIND based DNS* and *router protected name servers* into EDNS. *IP Anycast* allows each advertised name server's IP address to be associated or backed with multiple physical machines each of which are located across multiple networks. This mechanism is used by Enhanced DNS to direct their users to the topologically nearest node in a group of potential receivers all identified by the same destination address.

As explained by Akamai, Enhanced DNS primarily provides *zone delegation* and *name resolution* services to their users. Delegation of zones is performed to the Akamai name server platform after a user configures DNS zones at its primary name server. According to the need from the user, Akamai's name servers also resolve zones authoritatively. In addition, Akamai claims that their widely spread servers provide name resolution directly to end users as specified by *refresh* and *retry* parameters which are in the Start of Authority (SOA) of the DNS record.

Reference [7] indicates that for enhanced *data integrity*, reduced infrastructure cost and increased scalability, current solution of Enhanced DNS from Akamai also integrate DNSSEC so that secure zone transfers could be performed from the customer's master name server and uploaded to multiple name servers that are administered by Akamai. To ensure that the data is *authentic* before signing it, Transaction Signature (TSIG) authentication is configured for the customer zone transfer. Furthermore, their customers can also instruct the Domain Registrar that the Akamai name servers to authoritatively resolve their customer's zone and give the Delegation Signer (DS) record to the registrar.

Infoblox is another DNS infrastructure solution whose appliances deliver core network services including DNS and DNSSEC. [8] They provide solutions to the basic problems in distributed enterprises where each server had to be individually deployed, configured, managed and upgraded and that each server individually was not able to ensure the availability, accuracy and timeliness of data from network services.

Infoblox addresses problems faced by legacy solutions for DNS and DNSSEC. [8] Since previously DNS and DHCP services were deployed on assembled or so called "white-box" servers using software such as ISC BIND or Microsoft Windows Server, it was very insecure, expensive and unreliable as the

standard operating systems like Microsoft Windows, Unix, Solaris or Linux were vulnerable by themselves. Furthermore, deploying, securing and managing servers at remote locations used to be expensive and time consuming.

As explained by Reference [8], Infoblox's Grid technology is implemented by securely networking together the databases embedded within each appliance. Whenever there is a change in data residing in any appliance, it is reflected across the Grid in real time, preventing any possible loss of data, inconsistencies and errors and ensuring nonstop availability of distributed core network services. Furthermore, to make use of *increased performance* power of grids, Trivial File Transfer Protocol (TFTP) is used into a centrally managed, distributed service for VoIP deployments.

2. b Value Added Services and Infrastructure

Content Filtering

Content filtering is widely used currently in the internet. An effective filtering mechanism based on DNS should be strongly based on filtering based on IP addresses and based on individual domain names. Filtering based on public IP addresses can sometimes be a better mechanism because IP addresses are far scarcer than domain names and although there are endless amounts of domain names, there are only a limited number of IP addresses. [9]

What is DNS Based Content Filtering?

Content Filtering is a DNS based web filtering mechanism that is based on blocking particular DNS resolution requests that are associated with websites that host undesirable content. In such manner, only the desired content can be accessed from a web site and undesired content can be filtered out.

How does DNS Based Content Filtering Work?

Content Filtering can either be performed at the Client machine or at the Local DNS. At the Client machine, if the client already knows which specific sites are usually associated with undesired content, the client can specify in the *Hosts* file how those specific websites should be resolved.

A better mechanism to use DNS Based Content Filtering is at the Local DNS, since it would be easier to block a particular domain without specifying for each level domain associated with the particular domain. For example, if *doubleclick.net* is one site hosting advertisements, multiple third-level domains can include *crappyad.doubleclick.net* or *anothercrappyad.doubleclick.net*. And since it is not directly possible to use a wildcard (*) in *Hosts* file at the Client PC, DNS Based Content Filtering at the Local DNS server is a better mechanism.

What is the Added Value of DNS Based Content Filtering?

Content Filtering based on DNS provides the user a better control of his/her web experience by choosing between different contents on the internet. As indicated in [10], although the service might lower the response time of DNS resolution, the effect of slow response can often be regarded as *not noticeable* from the user side.

Where can DNS Based Ad Blocking be implemented?

Client Side, Resolver Side

Blacklisting/Whitelisting

What is DNS Based Blacklisting/Whitelisting?

DNS-based Blacklisting/Whitelisting is enlisting of IP addresses and publishing the list through the Internet Domain Name Service (DNS). DNSBL and DNSWL are either published as a zone file that can be used by DNS server software, or as a live DNS zone that can be queried in real-time. [11]

IP address Blacklisting is one of DNS based filtering mechanisms which can greatly reduce the *leasing value* of the scarce IP addresses so network operators are cautious of leasing their IP addresses to bad actors. Although there can be limitless supply of domain names, since IP addresses given to bad actors are limited, any new domain name that is hosted on the same block of identified IP addresses will automatically get blacklisted.

How does DNS Based Blacklisting Work?

DNS-based Blacklist (DNSBL) is a software mechanism, rather than a specific list or policy. There are dozens of DNSBLs in existence, which use a wide array of criteria for listing and delisting of addresses. As indicated in Reference [11], these blacklists may include listing addresses of computers being used to send spam, listing addresses of ISPs or listing addresses which have sent spam to a honeypot system. An effective blacklist, according to Reference [12], needs to be *complete*, containing a reasonable fraction of all spamming IP addresses; and it should be *responsive*, having a low response time so that other recipients can subsequently block spam originating from the respective IP addresses.

DNS based Blacklist check is heavily based on DNS, each check consisting of a DNS Query for every blacklist to be checked. For example, for a just received e-mail message, more than 20 DNS queries can be sent out for DNS Blacklist check. To handle this huge DNS traffic, the following points are important. [13]

- The caching name server needs to be fast and directly connected to the Internet and if possible direct DNS resolution should be done, rather than forwarding to other name server.
- The network flow should be optimized for small DNS queries.
- Since most DNSBL queries are aiming to get NXDOMAIN answers, which mean that the IP address is not listed, the name server should correctly cache negative responses for the DNSBL queries.

What is the Added Value of DNS Based Blacklisting?

DNSBL have first been created in 1997, and they have faced different operations and policies so far. Most users and e-mail systems operators regard them as very useful tool to share spam-related information, while others object them as a form of censorship. On a study reported in Reference [12], it has been noted that blacklists should have relatively *faster response time* if they are to keep up with Botnets that frequently change their domains.

Moreover, DNSBL have stayed a valuable mechanism to fight against spammers; this has happened to the extent that some DNSBL operators have been targeted by spammers in an effort to turn down their blacklists.

Where can DNS Based Blacklisting be implemented?

Client Side, Resolver Side

Parental Control

What is DNS Based Parental Control?

Parental Control is a software based DNS service that blocks a particular website on the internet from being loaded on client computers.

How does DNS Based Parental Control Work?

DNS Based Parental Control works by categorizing DNS resolution requests into groups according to a particular category that is set by the service administrator and configured by the user.

Currently, most of the leading Parental Control software are web based and accessible on subscription, meaning that the software license needs to be renewed timely. Some of these kinds of services are Google's Public DNS, DNS Advantage, Norton DNS and K9 Web. Other web based Parental Control service is OpenDNS which is free for use and works with all operating systems.

The OpenDNS service checks every web page in OpenDNS database of phishing. OpenDNS operates by using *PhishTank* as the source of phishing data. *PhishTank* is also used by other internet services like Yahoo Mail. In OpenDNS, information for Parental Control and adult site data is provided by *St. Bernand's iGuard* which is a human-reviewed URL database from *St. Bernanrd Software*. Once the user has registered an account with OpenDNS, they need to configure the networks that should be protected by OpenDNS. The DNS server addresses should be set as

Preferred DNS server address: 208.67.222.222
Alternate DNS server address: 208.67.220.220

What is the Added Value of DNS Based Parental Control?

Most home owners would like to take control of what each member of the family accesses through the internet, and would like to avoid inappropriate content from reaching an intended age group. Therefore Parental Control has an important value to home owners. The same applies to work places where the employer wants to take control of the internet usage based on content and time of his/her employees.

Where can DNS Based Parental Control be implemented?

Client Side, Resolver Side

Service Discovery

What is DNS Based Service Discovery?

DNS Based Service Discovery (DNSSD) is a way of using standard DNS programming interface, servers and packet formats to browse the network for services. [14] The IEEE draft on DNS Based Service Discovery specifies how DNS Resource Records are named and structured to facilitate service discovery. According to Reference [15], given a type of service that a client is looking for, and a domain in which the client is looking for that service, DNSSD allows the clients to discover a list of named instances of that desired service, using standard DNS queries.

How does DNS Based Service Discovery Work?

For DNS Based Service Discovery to work, records that advertise selected services to clients should be added to the DNS server, with no configuration (zero configuration) required from the client side. To do that, it is required to have administrative access to the domain name server so that necessary records are directly added to help the client discover network services like web pages or printers. In cases where the administrator has access to the DHCP server but no control over the DNS server, he/she can change the DHCP server to return a different domain for which the administrator has control and can add the necessary records for service discovery.

The points which are used to set up Bonjour name server for an Apple computer can be used as an example for DNS-SD. They have been included in the Appendix of this report in Section 8.

What is the Added Value of DNS Based Service Discovery?

When a client needs to contact a particular service, identified by a Service Instance Name, previously discovered using browsing or Service Instance Enumeration, it queries for the SRV and TXT records of that name. The SRV record for a service gives the port number and target host name where the service may be found whereas the TXT record gives additional information about the service.

As indicated in Reference [14], SRV records are very useful because they remove the need for pre-assigned port numbers. This is because of the limited number of available TCP port numbers, which are 65535. These port numbers are being allocated one-per-application-protocol. Using a different TCP port for each different instance of a given service on a given machine is possible but allocating each application its own large static range is not very practical. On any given host, most TCP ports are reserved for services that will not run. This is not a good utilization of the limited port space.

Each host can therefore benefit from allocating its available port numbers dynamically to services that are actually running on the host and it can advertise the allocated port numbers using SRV records. This can create a much better *utilization* of available port space than it is currently used in the internet.

Where can DNS Based Service Discovery be implemented?

Server Side

Home Remote Controlling

What is DNS Based Home Remote Controlling?

Nowadays, most electronic devices that are used at home are equipped with network connectivity which allows them to be accessed at some distance range in the surrounding. This ability initiates many users to consider remote controlling these devices using their mobile phones from elsewhere. As it is specified in Reference [16], for most home connections, remote access can be enabled with a combination of existing IP based technologies.

How does DNS Based Home Remote Controlling Work?

For Home Remote Controlling to work, a home gateway device that is to be accessed remotely has to be accessible from the internet using a public domain name and a unique and routable IP address. Under that, each home device needs to have a unique host name and dynamic IP addresses so that whenever

the IP address changes, the DNS record is updated so that the device is mapped to a static domain name.

A home gateway provides DNS and DHCP services to the home devices. A device needs to send DNS query to the gateway whenever it wants to connect to another device and wait for the DNS reply from the gateway with the IP address requested for. Therefore, to set up a Home DNS system, WLAN connectivity, a DHCP and DNS server are required.

Remote Clients can make use of Virtual Private Network (VPN) to create a secure communication channel to a home network over a public and shared network. Reference [16] states doing so makes the client feel as if he/she is connected directly to the home network.

What is the Added Value of DNS Based Home Remote Controlling?

Home DNS enables the remote access of home devices avoiding manual set up requirements. Moreover, a user is not required to remember the IP address of his/her home devices. There are a number of recent researches in the area including References [16] and [17] that aim to provide such a system or a prototype in a feasible manner with little issues regarding deployment.

Where can DNS Based Home Remote Controlling be implemented?

Client Side, Server Side

Telephone Number Mapping (ENUM)

What is Telephone Number Mapping?

As defined by the Internet Engineering Task Force RFC 2916, [18] Telephone Number Mapping (ENUM) is a standard of using DNS for storage of E.164 numbers so that DNS is used for identifying available services connected to an E.164 number.

How does Telephone Number Mapping Work?

Telephone Number Mapping works by using Domain Name System (DNS) resource records to map a telephone number into a collection of IP based addresses and service addresses. ENUM DNS is accessed from a VoIP gateway in order to check if a dialled number is reached via an IP service. The particular DNS response consists of NAPTR records for each service URI. The gateway then can choose a service from the collection.

The following steps are used to change a given E.164 number which is a number used for international public telecommunication according to ITU-T standard to a DNS name: [18]

1. Remove all non-digit characters from the E.164 number.
2. Put dots between each digit.
3. Reverse the order of the digits, and append the string ".e164.arps" to the end. Example, +31-9-87654321 can be changed to 1.2.3.4.5.6.7.8.9.1.3.e164.arpa.

On the other hand, as indicated in Reference [19], .tel is a top level domain for keeping contact information and data storage using NAPTR, TXT and LOC records within DNS. Differently from ENUM, .tel is not directly regulated by the telephone regulations and it is more portable as it is not linked to a particular phone number.

What is the Added Value of Telephone Number Mapping?

Most often used services including those that are IP based and PSTN based can converge into a single service that ties the two domains. [20] ENUM has lots of added values; it can map a phone number to a domain name, an email address, a web address or addresses of services that are identifiable by a URL. Therefore, many regard ENUM as an alternative to already existing PSTN operators.

Where can Telephone Number Mapping be implemented?

Client Side, Resolver Side, Server Side

Server Selection

What is DNS Based Server Selection?

Whenever a webpage is accessed, along with the normal domain name resolution, a redirection of clients to the nearest server can take place using *server selection* function. Such a technique is simple to implement in that there is no requirement to perform change on protocols and that it works on any IP-based application regardless of the transport-layer protocols used.

How does DNS Based Server Selection Work?

DNS-based server selection bases on the fact that client-side caching of DNS information should be avoided so that changes in network or server conditions could be properly reflected. Also, as pointed out in Reference [21], clients and their local name servers are assumed to be near to each other, and nearest server selection decisions rely mainly on the local name server, and not on the requesting client.

For DNS based server selection, there should be a trade-off between TTL values that should be set to very small values so that clients contact the authoritative name server for every name resolution request and increased latency in effect.

To analyse the impact of DNS TTL values, name resolution overhead and impact of embedded objects were analysed in Reference [21]. As explained in Reference [21], a web page download consists of *server name resolution*, *TCP connection establishment*, *transmission of the HTTP request*, *reception of the HTTP response*, *reception of data packets*, and *TCP connection termination*. To see the effect of Name Resolution Overhead, the paper measured the name lookup overhead by timing the *gethostbyname()* system call for each server hostname, with three levels of caching as follows:

(i) the local name server cache not having the server address and the authoritative name server address of the sub-domain
(ii) the local name server cache having the authoritative name server's address and
(iii) the local name server cache having the server's address in its cache.

When web pages contain a number of embedded objects that are not co-located, each object access may require an additional name resolution. The paper used *iptrace* tool to obtain the logs from the ISP proxy.

Furthermore, to determine client-name server proximity, the paper used measurement of network hops by probing site in the network using *traceroute*. In most of the cases which were found from

experimentation by Reference [21], clients were on average 8 hops away from their name servers. Therefore their initial assumption was slightly incorrect.

Thus, the general assumption by DNS-based server selection that clients and name servers are close to each other was addressed by Reference [21] by modifying the DNS protocol to carry additional information to identify the *actual* client making the request. Thus, they were able to make the DNS server which is involved in server selection/load balancing to make use of the client IP address in choosing the accurate selection.

The standard DNS message format consists of five sections: *header*, *question*, *answer*, *authority*, and *additional*. The modified DNS message which is proposed by Reference [21] has *additional* records section that consist a new DNS resource record with type CA (client address) along with the query. The TTL is zero for zero caching as the record applies only to the current transaction.

In conclusion, the paper discussed that when DNS TTL values are chosen for server selection/load balancing, additional mechanisms like new DNS resource record that keeps the originating client IP address are required in cases where client proximity is a decisive factor.

Where can DNS Based Server Selection be implemented?

Client Side, Resolver Side

Load Balancing

What is DNS Based Load Balancing?

In Reference [22], load balancing for Session Initiation Protocol based VoIP service based on the open source project Domain Name Relay Daemon (DNRD) for intercepting the prerequisite name resolution process in a client-server application was discussed. The technique presented by the paper indicated intercepting the prerequisite name resolution process by the *probing* mechanism DNRD to use it as a domain name resolution based load balancer DN-LB. In general, DNS based Load Balancing can increase the reliability and fault tolerance for the VoIP service with a low cost.

How does DNS Based Load Balancing Work?

The implementation of DNS Based Load Balancing requires the use of a dynamically configurable DNS DNRD (Domain Name Relay Daemon) and SIP Service Probing Daemon (SSPD). As Reference [22] explains, the open source project - DNRD (Domain Name Relay Daemon) on Linux can be used as a delegation name server not only to forward DNS queries to the appropriate name server but also to act as the primary name server. By default, DNRD acts as the primary name server for hosts found in its local *Host* file (/etc/hosts). The TTL for the SIP serving host entry is set to zero to prevent the clients from a cached IP but keep the clients have an up-to-date IP. Furthermore, a SIP Service Probing Daemon can be used to probe the health of all SIP proxy servers sequentially by use of the socket system call to build up a probing connection toward the SIP proxy servers. The SSPD is scheduled to send a dummy SIP message to check the service availability of the probed host and always selects the IP of the failure-proof and least-recently-set SIP proxy server as the on-duty IP in the DNRD configuration file (/etc/hosts).

In general, the basic knowledge of DNS based load balancing remains assigning multiple IP addresses to the *Host* record for the front end server and then administering the DNS server to rotate using those addresses in a *round-robin* fashion, so that workload is divided among the members of the server cluster equally.

In comparison to Network Load Balancing Service clusters, DNS based cluster nodes don't require to have multiple network interface cards, as each machine can simply have a single network interface card with a unique IP address. As Reference [23] points out, a DNS based cluster looks easier to configure than a network load balancing service, however, it is not as fault tolerant as a network load balancing service cluster. This is because no service provides a guarantee of fault tolerance or dynamic load rebalancing.

Edgedirector is a DNS service that covers global server load balancing of multiple distributed servers as explained in Reference [24]. By placing multiple servers at geographically distributed data centers, it ensures that when a website is visited from any different locations, visitors are served by the closest server.

The Edgedirector system operates in such a way that the backend system generates multiple records matching the geodns parameters set by the DNS administrator. When queries arrive from a visitor, the address record matching the geographical location of the query source is selected as the answer record. Following that, the visitor receives the DNS answer and initiates a connection to the server.

Further issues regarding DNS Based Global Server Load Balancing (GSLB) are pointed out in Reference [25] where the effect of *Browser DNS Caching* is discussed. Reference [25] further indicates that the main reason why multiple A records could not be used with GSLB is that the order in which addresses are returned could be changed by the client's DNS server. This can be due to a number of reasons one of which is that traffic could be evenly distributed to multiple sites. Therefore it is difficult to determine which site to choose from the multiple A records received. *Session persistence* is another issue that relates to sites that are hosted in multiple locations. Site cookies are used to avoid problems relating to browser resolving after a given period of time.

Where can DNS Based Load Balancing be implemented?

Resolver Side, Server Side

Multicast DNS

What is Multicast DNS?

Multicast DNS (mDNS) as explained in Reference [26] is a way of using familiar DNS programming interfaces, packet formats and operating semantics in a small network where no conventional DNS server has been installed. mDNS gives the ability to perform operations similar to a domain name system in a local environment in the absence of any conventional unicast DNS server and with little configuration.

How does Multicast DNS Work?

Multicast DNS implementation has been defined by IETF Zero Configuration Networking (zeroconf) so that manual administration or configuration of networking in a small office home office (SOHO), airplane and home networks is performed automatically. [27] The application programming interfaces used by multicast domain name system are similar to a unicast domain name system but it is implemented over a multicast protocol.

Reference [26] explains that an mDNS packet contains an IP TTL in the IP header used as a hop-count limit for the packet. Each Resource Record also contains a RR TTL which is the number of seconds for which the Resource Record may be cached.

Multicast DNS supports two different kinds of queries; One Shot queries similar to the ones made by legacy DNS resolvers and Continuous Ongoing Multicast DNS Queries made by fully-compliant Multicast DNS Queriers that support asynchronous operations including DNS based Service Discovery. [28]

In a multicast DNS network, when each computer joins the network, it enlists its own DNS Resource Records, (A, MX, SRV). Therefore, when a client in the network inquires for the IP address of a computer which it knows by name, the client sends a request to a well-known multicast address, and the computer with the corresponding A record replies with that IP address. Reference [29] specifies that the mDNS multicast address is 224.0.0.251 for IPv4 and ff02::fb for IPv6 link local addressing.

Where can Multicast DNS be implemented?

Client Side, Resolver Side

Performance Improvement

What is DNS Based Performance Improvement?

In relation to DNS, a number of factors affect the performance of network systems. The effectiveness of caching on performance of domain name system has been discussed in Reference [30]. Varying TTLs and degrees of cache sharing have various effects on DNS cache Hit rates. As expressed in Reference [31], the explosive growth of the domain namespace has decreased the effectiveness of DNS caching. On the other hand, widespread caching of mappings in the DNS prohibits fast propagation of unanticipated changes. Manual configuration errors, such as lame delegations introduce latency in performance of domain name systems. With respect to attacks targeting the domain name system, vulnerability comes from DoS attacks since in most cases there is limited redundancy in name servers. For example, as stated in Reference [31], approximately 80% of domain names are served by only two name servers and some percentage by only one.

How does DNS Based Performance Improvement Work?

Reference [31] proposes Cooperative Domain Name System (CoDoNS) as a replacement of the legacy DNS, incorporating *High Performance*, *Lower Latency*, *Faster Update Propagation* and *Resiliency to Attacks*. CoDoNS combines structured peer-to-peer overlays and analytically informed proactive caching. [32]

CoDoNS provides the same query resolution service to clients as legacy DNS and therefore requires no change to client resolvers. However, each CoDoNS server implements a recursive, caching DNS resolver with an architecture consisting of globally distributed nodes that form a peer-to-peer network.

Cooperative DNS has a proactive caching layer named in Reference [31] as Beehive. It is replication framework that makes use of prefix matching Distributed Hash Tables in order to achieve lookup performance in O(1). Using prefix matching, each node routes a request for an object for example 10101 by successively matching *one more digit* with the object until it reaches at the home node, 10101 for this example. This kind of matching requires O (log N) hops at the worst case to reach at the home node.

What is the Added Value of DNS Based Performance Improvement?

From the results that have been found, Reference [31] has mentioned that CoDoNS achieves *lower latencies* in 90 percentile of deployment scenarios with and without redirections as compared to the legacy DNS. Furthermore, it is mentioned that CoDoNS *resists denial of service attacks*, *automatically distributes load* and *supports fast updates*. Reference [32] mentioned that CoDNS *reduces average lookup latency* by 27-82%, greatly *reduces slow lookups*, and *improves DNS availability*. Further work on configuration of CoDNS that provides *improved security*, *reliability* and *performance* has been discussed in References [33] and [34].

Where can Performance Improvement be implemented?

Resolver Side, Server Side

Estimation

What is DNS Based Estimation?

DNS Based Estimation is measuring the relative popularity of a website or measuring the number of users accessing a website over a given period of time. This kind of estimation is usually performed by the deployment of client-side measurement agents some examples of which are Alexa, ComScore and Nielsen in References [36], [37] and [38]. As it is pointed in Reference [39], such act is sometimes perceived as infringing on users' privacy limiting the wide scale adoption of such a technique. Considering such aspects, DNS cache probing can be a better estimation technique to infer the density of clients accessing a given service. DNS cache probing is less invasive as discussed in Reference [39] since it does not reveal user specific characteristics and is more robust against manipulation.

How does DNS Based Estimation work?

DNS Based Estimation works by making use of DNS cache probing as a technique to probe the DNS resolver cache for each Domain Name of interest at regular time intervals and examine the observed cache *hits* or *misses*. For each cache probe, a cooperative resolver will report a *hit* if the Domain Name is in cache or a *miss* if the Domain Name is not in cache. Moreover, Reference [39] discusses that the probes should be sent at a regular time interval so that it is possible to know for how long the specific Domain Name stays in the resolver's cache.

What is the Added Value?

Measuring the relative popularity of websites is very useful for a number of purposes. Marketing professionals can make use of its results to advertise their products and services in an effective manner. According to Reference [39], reliable determination of an infected population or botnet from a network is also possible by DNS cache probing.

Where can DNS Based Estimation be implemented?

Resolver Side

Dynamic DNS

What is Dynamic DNS?

The Wikipedia article on Dynamic DNS, Reference [40], defines Dynamic DNS as a method, protocol, network service that provides the capability for a networked device, such as a router or computer system using the Internet Protocol Suite, to notify a Domain Name System (DNS) name server to change, in real time, the active DNS configuration of its configured hostnames, addresses or other information.

How does Dynamic DNS work?

'Dynamic DNS Update' or 'DDNS' in Reference [41] describes how to dynamically update name server records. Unlike the DynDNS-type updates, RFC 2136 is a protocol with its own security mechanisms. It supports all DNS record types including zone and user and it is used primarily as an extension of the DHCP system, and in which the authorized DHCP servers register the clients' records with the nameserver. This form of support for RFC 2136 is provided by a client and server software including directory services like Lightweight Directory Access Protocol and Windows Active Directory.

What is the Added Value?

Dynamic DNS can be used to give a well-known hostname to variable and changing IP addresses. In a residential network, a service called DynDNS is useful in which the gateway to the internet can have a changing IP address with a unique host name that can be resolved through standard DNS queries.

Where can Dynamic DNS be implemented?

Server Side

Security

What is DNS Based Security?

DNS Based Security is a way of providing a better, trustworthy and safer name resolution protocol for the Internet as expressed in Reference [42]. The primary goal of DNSSEC is to provide authentication and integrity for data received from the DNS database using digital signature schema based on public key cryptography. [43]

How does DNS Based Security work?

One way in which DNSSEC works is by associating each node in the DNS tree with a public key. Each message from DNS servers is signed under the corresponding private key. Using one or more authenticated DNS root public keys, a signature can be generated that binds the identity information of each top-level domain to the corresponding public key and thus certificates can be generated on behalf of top level domains, which on their own sign the keys of their subdomains and so on. As described in Reference [43], each parent signs the public keys of all its children in the DNS tree.

The two different kinds of signatures for DNS messages are Transaction Signatures (TSIGs) and Public Key Signatures (SIGs). Transaction Signatures are based on symmetric techniques and are used for transactions between local servers whereas Public Key Signatures are used for protecting the authenticity and integrity of the message. [43]

What is the Added Value?

If properly implemented, DNSSEC can give a level of security that is desirable for authentication and data integrity of the DNS.

Where can Security be implemented?

Resolver Side, Server Side

NXDomain Redirection

What is NXDomain Redirection?

NXDomain redirection is a DNS response modification service by a name server provider that responds for a queried name that does not exist in the hosted zone file with a chosen IP address and a Name Exists reply rather than a Non-Existing Name reply.

How does NXDomain Redirection work?

Whenever the name server provider receives a Name Error response code, the contents of the DNS response should be changed so that the response code indicates that the name exists rather than it does not exist. The provider should also modify the response so that a desired IP address is mapped for the queried name.

To explain the Redirection process at the Registry level of the DNS, the following steps are indicated in Reference [44].

1. A client queries for a domain name to an iterative resolver.
2. The iterative resolver starts the resolution by forwarding the query to a root name server.
3. The root name server returns a list of name servers that could resolve for top level domain.
4. The iterative resolver sends the query to resolve the domain name to one of the top level domain's name servers identified by the root name server.
5. The top level domain then finds out that the domain name does not match a specific label in top level domain's zone file therefore in effect it returns DNS response message resolving the

domain name to a desired IP address rather than responding a Name Error, the process is called NXDomain Redirection.

6. The iterative resolver forwards the positive response message to the client that originated the request.

What is the Added Value?

As indicated in Reference [44], NXDomain Redirection can be done for a number of reasons including *revenue generation*, *enhancing the user's web experience*, *enforcing a policy*, *providing remedial notices* and *abetting criminal activities*.

More importantly, in case of *revenue generation*, the website for which the user is redirected to will get the benefit of more users coming to its site. And in case of *enhancing users' web experience*; users benefit from not bothering about error messages for queried domains that may not exist.

Much of the work that has been done on NXDomain Redirection is on DNS based and Web based applications, how it affects other IP-based services including email, voice, and other routing based Internet operations needs further study.

Where can DNS Based NXDomain Redirection be implemented?

Resolver Side

Clustering

What is DNS Based Clustering?

DNS Based Clustering is a context-aware clustering service applied to DNS query responses to be able to develop their aggregation in a desired manner. In Reference [45], such aggregation is divided into Canonical (RFC intended behaviour), overloaded (black-list servers) and unwanted (un-succeeding queries). These provide a useful perspective for real time analysis and visualization of network traffic and management.

Although port numbers could also be used to categorize network traffic, in cases where standard protocols like HTTP are used, these categories based on port numbers might not be sufficient. Therefore non-port based network traffic identification like DNS Based Clustering is important.

How does DNS Based Clustering work?

One way to implement DNS Based Clustering as discussed in Reference [45] is to use data-driven and context-aware mechanisms for clustering. This includes using DNS syntax and semantics to classify DNS query or response traces. Furthermore, IP prefix and domain name search trees can be used to classify the clusters further.

What is the Added Value?

As pointed out in Reference [45], the most important advantage of DNS Based Clustering is the identification of network traffic for its proper classification and management; and also for traffic engineering and network security.

Where can DNS Based Clustering be implemented?

Resolvers Side, Server Side

Anonymous Resolution of DNS Queries

What is Anonymous Resolution of DNS Queries?

Anonymous Resolution of DNS Queries is a service for avoiding unprotected data exchange between servers and clients. By doing so, DNS query can be performed with integrity and authenticity.

How does Anonymous Resolution of DNS Queries work?

As discussed in Reference [46], Anonymous Resolution of DNS Queries works by using a model called Privacy Information Retrieval in which the authors discussed that a random noise needs to be introduced in DNS queries.

What is the Added Value?

Anonymous Resolution of DNS Queries makes queries to be more secure against those who would like to make use of the information gathered from it.

Where can Anonymous Resolution of DNS Queries be implemented?

Resolver Side

Object Naming Service

What is Object Naming Service?

The Object Naming service makes use of the Domain Name System to resolve information about a given Electronic Product Code. According to Reference [47], this should be performed by changing the EPC to a domain name, having the result in form of a valid DNS resource record so that the query and response formats are according to the DNS standards.

How does Object Naming Service work?

In Reference [48], Object Naming Service is explained to work by using Radio Frequency Identification (RFID) and assigning a globally unique number to every tagged object which provides associated information regarding the object. Additionally, since the information regarding the object is stored on the internet, the usage of DNS is found essential. Further specification about DNS records for ONS and how ONS queries are processed is explained in Reference [47].

What is the Added Value?

As discussed in Reference [48], the main advantage of Object Naming Service is the real time processing of tagged objects which might be moving from one place to another.

Where can Object Naming Service be implemented?
Client Side

Software Version Updating

What is DNS Based Software Version Updating?

DNS Based Software Version Updating is the use of Domain Name System and Software Update Service (SUS) to update software versions running on client machines. Rather than each client directly accessing the same internet connection to download the same software update pack which consumes bandwidth greatly and saturates inbound link, a Software Update Service can download the required packages and clients can use DNS to contact the Software Update Service and make required updates from it.

How does DNS Based Software Version Updating work?

In Reference [49], a detailed explanation of how DNS Based Software Version Updating works is presented. First the internal DNS should have CNAME representing the Software Update Service. Then in the server administration, all traffic that queries for contacting the antivirus website has to be redirected to the Software Update Service.

What is the Added Value?

The main advantage of having a DNS Based Software Version Updating is to have a network in which clients can make use of the DNS to update the software version running on their machines. Rather than each client independently accessing the web site of the software updating website, they can indirectly access the website through the Software Update Server using DNS. This greatly saves bandwidth of the network.

Where can Software Version Updating be implemented?

Client Side, Resolver Side

Context Aware Naming

What is DNS Based Context Aware Naming?

DNS Based Context Aware Naming is a DNS Based service that enables context based Domain Name resolution. In Reference [50], the architecture for context based name resolution service has been presented.

How does DNS Based Context Aware Naming work?

Context Aware Naming is implemented by tagging context information to be able to translate a given context into a locator or identifier or a record containing contact information. This requires the implementation of context description layer that directly interacts with sensors and observers to obtain

context information. What is proposed in Reference [50] is an ontology description language, a repository and a reasoning mechanism to relate the given ontology to the context information.

What is the Added Value?

The added value of DNS Based Context Aware Naming is that it provides a name resolution service that could be developed for the future internet. As expressed in Reference [50], Context Aware Naming enables discovery of resources that have been given a name in the name space.

Where can DNS Based Context Aware Naming be implemented?

Server Side

3 Technical Analysis on the Performance of DNS Resolvers

DNS Benchmark

DNS Benchmark is a tool that is used to properly determine the *performance* of local and remote DNS resolvers mainly in terms of their *response time*. DNS Benchmark gives the practical information about what is going on in the Domain Name System resolvers by comparing their performance with available alternative DNS servers.

Why is DNS Benchmark Useful?

One important thing to notice about DNS Benchmark is that the *suitability* of a DNS resolver is really dependent on the particular *location* a user is found. A specific DNS resolver might feel as *very fast* from one *location* but that same resolver might not feel *equally as fast* from some other *location*; or at least there is no guarantee if it does. Therefore a DNS Benchmark has to be used from every *location* possible to perform the accurate performance evaluation of a particular domain name server.

The DNS Benchmark used in this technical analysis is from GRC Research, a computer software development firm founded by Steve Gibson. The idea of using a DNS Benchmark is to take a better measure of DNS Servers that a system might be using, for example to use a number of DNS Servers in the proper ordering corresponding to their speed of response. As indicated in Reference [35], GRC'S DNS Benchmark performs a detailed analysis and comparison of the operational performance and reliability of around two hundred resolvers.

How does DNS Benchmark Work?

Once the GRC Benchmark is started, it identifies the user's DNS resolvers and publicly available "alternative" nameservers. Then each DNS nameserver in the benchmark list is identified if it is a suitable DNS resolver based on its characteristics. For example, based on *how it handles NXDOMAIN replies*, *whether or not it returns an error for a bad domain request*, or *if it redirects a user's web browser to a commercial marketing webpage*. Such behaviour, as explained by GRC Research, can be acceptable for some users, but might not be suitable for others; therefore it is used as one way of characterization.

Once the GRC Benchmark is performed, statistical results that give a summary of findings and conclusion are provided by the benchmark; accompanied with a recommendation about the system's resolvers or any other alternative resolvers based on their feature comparison.

What are Notable Results?

In the GRC DNS Benchmark, different colours identify different kinds of resolvers analysed in the DNS Benchmark. "Green" dots are good and functional; "Orange" dots represent the Redirecting Nameservers that do not return errors when asked to lookup an Invalid Domain or NXDOMAIN. "Red" dot represent those DNS servers that refuse to reply to queries. For example, from the current location of where the benchmark is run, the available public resolvers may be inaccessible to the computer, thus dead servers although they may be accessible to other users on the internet.

Regardless of the colour, a dot that is "filled-in" indicates that the server is *used* by the system at the moment, where as a dot that is "hollow" indicates that the server is *not used* by the system at the moment.

According to Reference [35], the outer circle of the resolver status icon shows if there is *DNS Rebinding Attack Protection* provided by the nameserver. These attacks use DNS information so that a browser thinks that the local resources of computer or a router are located in the web domain of the scripts source. If a particular DNS nameserver is *security aware*, it can block these *DNS Rebinding Attacks* by never returning IP addresses that fall within the ranges of private IP addresses commonly used with *private LAN networks* behind a router or the Localhost IP of 127.0.0.1.

For the *DNS Rebinding Attack Protection* explained above, the Benchmark used in this paper tests each resolver to find out if it blocks the return of *reserved private IP addresses* both in IPv4 and IPv6 IP addresses. As explained in the benchmark specification, it is not required to return a private IP address from a public DNS request therefore all resolvers should avoid returning *private IP addresses*. The following diagram shows how this is performed.

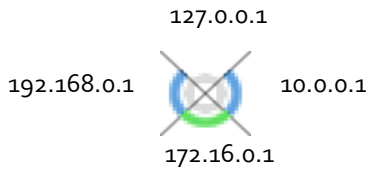


Figure 1: DNS Rebinding Attack Protection, Reference [35]

In this case, the diagram represents that the outer circle is divided into four quadrants representing the *Protection against DNS Rebinding Attack*. The outer circle is empty for 127.0.0.1 indicating *No Blocking* or *Filtering* is provided by the nameserver for that network IP. For 192.168.0.1 and 10.0.0.1 private IP addresses, the Blue arc indicates that *blocking* or *filtering* is provided for either the IPv4 or IPv6 address, but not both. A Green arc for the 172.16.0.1 indicates that *blocking* or *filtering* is provided for both IPv4 and IPv6 addresses.

The effect of DNS Caching is also taken into consideration. A “Red Bar” indicates that the Domain Name lookup is already Cached, a “Green Bar” indicates that the Domain Name lookup is Uncached, while a “Purple Bar” indicates that the Domain Name lookup is a Dot Com such as the most popular domains like Bing, Google, Yahoo or others upon which the look up of another request depends.

In addition to sorting servers according to *caching performance*, the GRC DNS Benchmark has additional “Discoverable” *Power-User Features* that include *Removing a nameserver*, *Removing X dead nameservers*, *Removing slower nameservers*, *Sorting servers by Uncached performance* and *Testing DNSSEC Authentication*. Those were not directly used in this paper.

Results of Running GRC DNS Benchmark

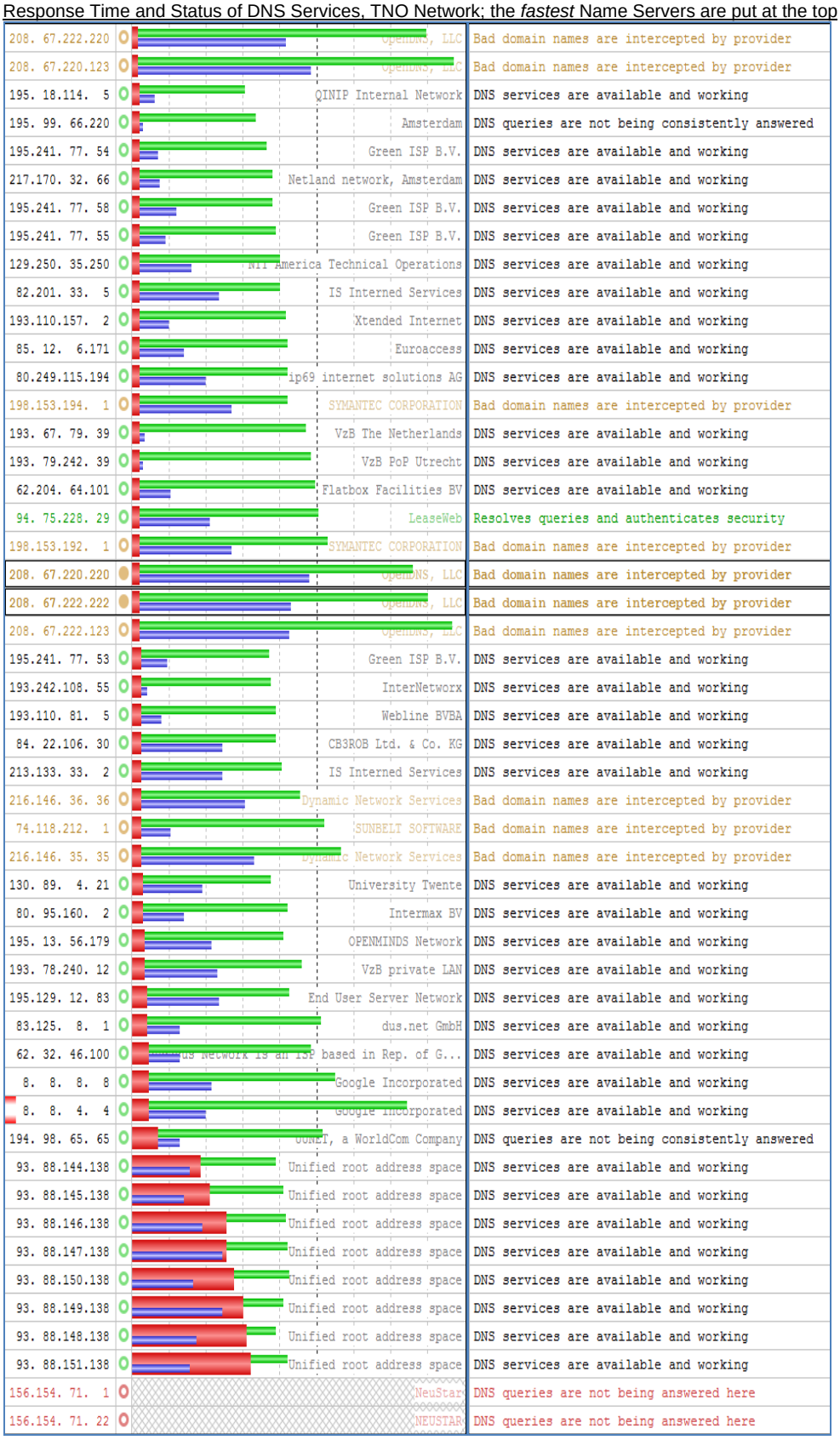
The *fastest* name servers are put at the top if the *Sort Fastest First* checkbox is *checked*. The Analysis is presented based on the *Company Name*, the *Address of the Domain*, the *Response Time* and the *Status of DNS Services*.

The following results are observed after running the GRC DNS Benchmark from the TNO Network. The results show that for example for this particular time (benchmark run on 13/09/2011), OpenDNS (specifically the 208.67.222.220 OpenDNS resolver) is faster and performs better that any of the other available name servers including NeuStar and Google’s Public DNS.

Company and Name of DNS Services, TNO Network; the *fastest* Name Servers are put at the top

OpenDNS, LLC	208. 67.222.220	resolver3.opendns.com
OpenDNS, LLC	208. 67.220.123	resolver2-fs.opendns.com
QINIP Internal Network	195. 18.114. 5	opaal.qinip.net
Amsterdam	195. 99. 66.220	nsr1.nl-ams2.eu.bt.net
Green ISP B.V.	195.241. 77. 54	ns2.tiscali.nl
Netland network, Amsterdam	217.170. 32. 66	ns.netland.nl
Green ISP B.V.	195.241. 77. 58	cns2.net.telfort.nl
Green ISP B.V.	195.241. 77. 55	cns1.net.telfort.nl
NTT America Technical Operations	129.250. 35.250	x.ns.gin.ntt.net
IS Interned Services	82.201. 33. 5	ns2.is.nl
Xtended Internet	193.110.157. 2	ns.xtdnet.nl
Euroaccess	85. 12. 6.171	vm1.rootspirit.com
ip69 internet solutions AG	80.249.115.194	··· no official Internet DNS name ···
SYMANTEC CORPORATION	198.153.194. 1	··· no official Internet DNS name ···
VzB The Netherlands	193. 67. 79. 39	cache0200.ns.eu.uu.net
VzB PoP Utrecht	193. 79.242. 39	cache0205.ns.eu.uu.net
Flatbox Facilities BV	62.204. 64.101	ns1.flatbox-facilities.net
LeaseWeb	94. 75.228. 29	privacybox.de
SYMANTEC CORPORATION	198.153.192. 1	··· no official Internet DNS name ···
OpenDNS, LLC	208. 67.220.220	resolver2.opendns.com
OpenDNS, LLC	208. 67.222.222	resolver1.opendns.com
OpenDNS, LLC	208. 67.222.123	resolver1-fs.opendns.com
Green ISP B.V.	195.241. 77. 53	ns1.tiscali.nl
InterNetworx	193.242.108. 55	wijkradenbrummen.nl
Webline BVBA	193.110. 81. 5	ns1.webline.be
CB3ROB Ltd. & Co. KG	84. 22.106. 30	ns2.public-root.net
IS Interned Services	213.133. 33. 2	ns1.is.nl
Dynamic Network Services	216.146. 36. 36	resolver2.dyndnsinternetguide.com
SUNBELT SOFTWARE	74.118.212. 1	··· no official Internet DNS name ···
Dynamic Network Services	216.146. 35. 35	resolver1.dyndnsinternetguide.com
University Twente	130. 89. 4. 21	dc1service.service.utwente.nl
Intermax BV	80. 95.160. 2	ns1.intermax.nl
OPENMINDS Network	195. 13. 56.179	host4.karakas.be
VzB private LAN	193. 78.240. 12	cache0204.ns.eu.uu.net
End User Server Network	195.129. 12. 83	cache0206.ns.eu.uu.net
dus.net GmbH	83.125. 8. 1	ns2.dus.net
Caucasus Network is an ISP based in Rep. of G...	62. 32. 46.100	ns2.mach-six.com
Google Incorporated	8. 8. 8. 8	google-public-dns-a.google.com
Google Incorporated	8. 8. 4. 4	google-public-dns-b.google.com
UUNET, a WorldCom Company	194. 98. 65. 65	cache0300.ns.eu.uu.net
Unified root address space	93. 88.144.138	DNS1.UNIFIEDROOT.COM
Unified root address space	93. 88.145.138	DNS2.UNIFIEDROOT.COM
Unified root address space	93. 88.146.138	DNS3.UNIFIEDROOT.COM
Unified root address space	93. 88.147.138	DNS4.UNIFIEDROOT.COM
Unified root address space	93. 88.150.138	DNS7.UNIFIEDROOT.COM
Unified root address space	93. 88.149.138	DNS6.UNIFIEDROOT.COM
Unified root address space	93. 88.148.138	DNS5.UNIFIEDROOT.COM
Unified root address space	93. 88.151.138	DNS8.UNIFIEDROOT.COM
NeuStar	156.154. 71. 1	rdns2.ultradns.net
NEUSTAR	156.154. 71. 22	··· no official Internet DNS name ···

Response Time and Status of DNS Services, TNO Network; the *fastest* Name Servers are put at the top



208. 67.222.220	OpenDNS, LLC	Bad domain names are intercepted by provider
208. 67.220.123	OpenDNS, LLC	Bad domain names are intercepted by provider
195. 18.114. 5	QINIP Internal Network	DNS services are available and working
195. 99. 66.220	Amsterdam	DNS queries are not being consistently answered
195.241. 77. 54	Green ISP B.V.	DNS services are available and working
217.170. 32. 66	Netland network, Amsterdam	DNS services are available and working
195.241. 77. 58	Green ISP B.V.	DNS services are available and working
195.241. 77. 55	Green ISP B.V.	DNS services are available and working
129.250. 35.250	NTT America Technical Operations	DNS services are available and working
82.201. 33. 5	IS Interned Services	DNS services are available and working
193.110.157. 2	Xtended Internet	DNS services are available and working
85. 12. 6.171	Euroaccess	DNS services are available and working
80.249.115.194	ip69 internet solutions AG	DNS services are available and working
198.153.194. 1	SYMANTEC CORPORATION	Bad domain names are intercepted by provider
193. 67. 79. 39	VzB The Netherlands	DNS services are available and working
193. 79.242. 39	VzB PoP Utrecht	DNS services are available and working
62.204. 64.101	Flatbox Facilities BV	DNS services are available and working
94. 75.228. 29	LeaseWeb	Resolves queries and authenticates security
198.153.192. 1	SYMANTEC CORPORATION	Bad domain names are intercepted by provider
208. 67.220.220	OpenDNS, LLC	Bad domain names are intercepted by provider
208. 67.222.222	OpenDNS, LLC	Bad domain names are intercepted by provider
208. 67.222.123	OpenDNS, LLC	Bad domain names are intercepted by provider
195.241. 77. 53	Green ISP B.V.	DNS services are available and working
193.242.108. 55	InterNetworx	DNS services are available and working
193.110. 81. 5	Webline BVBA	DNS services are available and working
84. 22.106. 30	CB3ROB Ltd. & Co. KG	DNS services are available and working
213.133. 33. 2	IS Interned Services	DNS services are available and working
216.146. 36. 36	Dynamic Network Services	Bad domain names are intercepted by provider
74.118.212. 1	SUNBELT SOFTWARE	Bad domain names are intercepted by provider
216.146. 35. 35	Dynamic Network Services	Bad domain names are intercepted by provider
130. 89. 4. 21	University Twente	DNS services are available and working
80. 95.160. 2	Intermax BV	DNS services are available and working
195. 13. 56.179	OPENMINDS Network	DNS services are available and working
193. 78.240. 12	VzB private LAN	DNS services are available and working
195.129. 12. 83	End User Server Network	DNS services are available and working
83.125. 8. 1	dus.net GmbH	DNS services are available and working
62. 32. 46.100	dus Network IS an ISP based in Rep. of G...	DNS services are available and working
8. 8. 8. 8	Google Incorporated	DNS services are available and working
8. 8. 4. 4	Google Incorporated	DNS services are available and working
194. 98. 65. 65	GUNET, a WorldCom Company	DNS queries are not being consistently answered
93. 88.144.138	Unified root address space	DNS services are available and working
93. 88.145.138	Unified root address space	DNS services are available and working
93. 88.146.138	Unified root address space	DNS services are available and working
93. 88.147.138	Unified root address space	DNS services are available and working
93. 88.150.138	Unified root address space	DNS services are available and working
93. 88.149.138	Unified root address space	DNS services are available and working
93. 88.148.138	Unified root address space	DNS services are available and working
93. 88.151.138	Unified root address space	DNS services are available and working
156.154. 71. 1	NeuStar	DNS queries are not being answered here
156.154. 71. 22	NEUSTAR	DNS queries are not being answered here

The Tabular Data for the TNO Network is provided below. It compares the Speed of resolving a *Cached Name*, an *Uncached Name* and a *DotCom Lookup* using the different Nameservers.

Final benchmark results, sorted by nameserver performance:
(Average cached name retrieval speed, fastest to slowest)

208. 67.222.220	Min	Avg	Max	Std.Dev	Reliab%
- Cached Name	0.003	0.003	0.005	0.001	100.0
- Uncached Name	0.004	0.159	1.088	0.242	100.0
- DotCom Lookup	0.006	0.083	0.272	0.063	100.0

resolver3.opendns.com
OpenDNS, LLC

208. 67.220.123	Min	Avg	Max	Std.Dev	Reliab%
- Cached Name	0.003	0.003	0.004	0.000	100.0
- Uncached Name	0.004	0.174	1.275	0.306	100.0
- DotCom Lookup	0.008	0.097	0.358	0.090	100.0

resolver2-fs.opendns.com
OpenDNS, LLC

195. 18.114. 5	Min	Avg	Max	Std.Dev	Reliab%
- Cached Name	0.004	0.004	0.005	0.000	100.0
- Uncached Name	0.004	0.061	0.270	0.079	100.0
- DotCom Lookup	0.004	0.012	0.099	0.022	100.0

opaal.qinip.net
QINIP Internal Network

195. 99. 66.220	Min	Avg	Max	Std.Dev	Reliab%
- Cached Name	0.004	0.004	0.006	0.000	100.0
- Uncached Name	0.005	0.067	0.331	0.086	100.0
- DotCom Lookup	0.005	0.006	0.007	0.000	100.0

nsr1.nl-ams2.eu.bt.net
Amsterdam

195.241. 77. 54	Min	Avg	Max	Std.Dev	Reliab%
- Cached Name	0.004	0.004	0.007	0.000	100.0
- Uncached Name	0.006	0.073	0.372	0.100	100.0
- DotCom Lookup	0.008	0.014	0.118	0.016	100.0

ns2.tiscali.nl
Green ISP B.V.

217.170. 32. 66	Min	Avg	Max	Std.Dev	Reliab%
- Cached Name	0.004	0.004	0.006	0.000	100.0
- Uncached Name	0.003	0.076	0.412	0.111	100.0
- DotCom Lookup	0.004	0.015	0.129	0.030	100.0

ns.netland.nl
Netland network, Amsterdam

129.250. 35.250	Min	Avg	Max	Std.Dev	Reliab%
- Cached Name	0.004	0.004	0.005	0.000	100.0
- Uncached Name	0.005	0.080	0.342	0.093	100.0
- DotCom Lookup	0.005	0.032	0.168	0.043	100.0

x.ns.gin.ntt.net
NTT America Technical Operations

82.201. 33. 5	Min	Avg	Max	Std.Dev	Reliab%

- Cached Name	0.003	0.004	0.005	0.000	100.0
- Uncached Name	0.005	0.080	0.314	0.096	100.0
- DotCom Lookup	0.005	0.047	0.090	0.042	100.0

ns2.is.nl
IS Interned Services

193.110.157. 2	Min	Avg	Max	Std.Dev	Reliab%
- Cached Name	0.004	0.004	0.006	0.000	100.0
- Uncached Name	0.004	0.083	0.420	0.117	100.0
- DotCom Lookup	0.004	0.020	0.133	0.032	100.0

ns.xtdnet.nl
Xtended Internet

85. 12. 6.171	Min	Avg	Max	Std.Dev	Reliab%
- Cached Name	0.004	0.004	0.005	0.000	100.0
- Uncached Name	0.005	0.084	0.404	0.101	100.0
- DotCom Lookup	0.008	0.028	0.132	0.028	100.0

vm1.rootspirit.com
Euroaccess

80.249.115.194	Min	Avg	Max	Std.Dev	Reliab%
- Cached Name	0.004	0.004	0.005	0.000	100.0
- Uncached Name	0.005	0.084	0.563	0.120	100.0
- DotCom Lookup	0.005	0.040	0.266	0.067	100.0

··· no official Internet DNS name ···
ip69 internet solutions AG

198.153.194. 1	Min	Avg	Max	Std.Dev	Reliab%
- Cached Name	0.004	0.004	0.005	0.000	100.0
- Uncached Name	0.004	0.084	0.287	0.088	100.0
- DotCom Lookup	0.006	0.054	0.151	0.050	100.0

··· no official Internet DNS name ···
SYMANTEC CORPORATION

193. 67. 79. 39	Min	Avg	Max	Std.Dev	Reliab%
- Cached Name	0.003	0.004	0.006	0.000	100.0
- Uncached Name	0.005	0.094	0.426	0.112	100.0
- DotCom Lookup	0.005	0.007	0.013	0.002	100.0

cache0200.ns.eu.uu.net
VzB The Netherlands

62.204. 64.101	Min	Avg	Max	Std.Dev	Reliab%
- Cached Name	0.003	0.004	0.005	0.000	100.0
- Uncached Name	0.007	0.099	0.338	0.102	100.0
- DotCom Lookup	0.007	0.021	0.122	0.028	100.0

ns1.flatbox-facilities.net
Flatbox Facilities BV

94. 75.228. 29	Min	Avg	Max	Std.Dev	Reliab%
- Cached Name	0.004	0.004	0.007	0.000	100.0
- Uncached Name	0.005	0.101	0.358	0.097	100.0
- DotCom Lookup	0.007	0.042	0.131	0.043	100.0

privacybox.de
LeaseWeb

198.153.192. 1	Min	Avg	Max	Std.Dev	Reliab%

- Cached Name | 0.004 | 0.004 | 0.005 | 0.000 | 100.0 |
- Uncached Name | 0.005 | 0.106 | 0.392 | 0.112 | 100.0 |
- DotCom Lookup | 0.005 | 0.054 | 0.127 | 0.046 | 100.0 |

··· no official Internet DNS name ···
SYMANTEC CORPORATION

208. 67.222.222 | Min | Avg | Max |Std.Dev|Reliab%|

+ Cached Name | 0.003 | 0.004 | 0.007 | 0.000 | 100.0 |
+ Uncached Name | 0.005 | 0.160 | 1.292 | 0.265 | 100.0 |
+ DotCom Lookup | 0.006 | 0.086 | 0.161 | 0.056 | 100.0 |

resolver1.opendns.com
OpenDNS, LLC

193.242.108. 55 | Min | Avg | Max |Std.Dev|Reliab%|

- Cached Name | 0.004 | 0.005 | 0.007 | 0.001 | 100.0 |
- Uncached Name | 0.005 | 0.075 | 0.366 | 0.095 | 100.0 |
- DotCom Lookup | 0.004 | 0.008 | 0.010 | 0.001 | 100.0 |

wijkradenbrummen.nl
InterNetworx

193.110. 81. 5 | Min | Avg | Max |Std.Dev|Reliab%|

- Cached Name | 0.004 | 0.005 | 0.005 | 0.000 | 100.0 |
- Uncached Name | 0.006 | 0.078 | 0.346 | 0.100 | 100.0 |
- DotCom Lookup | 0.009 | 0.016 | 0.091 | 0.012 | 100.0 |

ns1.webline.be
Webline BVBA

84. 22.106. 30 | Min | Avg | Max |Std.Dev|Reliab%|

- Cached Name | 0.005 | 0.005 | 0.006 | 0.000 | 100.0 |
- Uncached Name | 0.007 | 0.078 | 0.293 | 0.087 | 100.0 |
- DotCom Lookup | 0.009 | 0.049 | 0.178 | 0.039 | 100.0 |

ns2.public-root.net
CB3ROB Ltd. & Co. KG

213.133. 33. 2 | Min | Avg | Max |Std.Dev|Reliab%|

- Cached Name | 0.003 | 0.005 | 0.008 | 0.001 | 100.0 |
- Uncached Name | 0.005 | 0.081 | 0.361 | 0.093 | 100.0 |
- DotCom Lookup | 0.005 | 0.049 | 0.102 | 0.037 | 100.0 |

ns1.is.nl
IS Interned Services

216.146. 36. 36 | Min | Avg | Max |Std.Dev|Reliab%|

- Cached Name | 0.004 | 0.005 | 0.006 | 0.000 | 100.0 |
- Uncached Name | 0.006 | 0.091 | 0.285 | 0.086 | 100.0 |
- DotCom Lookup | 0.007 | 0.061 | 0.152 | 0.050 | 100.0 |

resolver2.dyndnsinternetguide.com
Dynamic Network Services

74.118.212. 1 | Min | Avg | Max |Std.Dev|Reliab%|

- Cached Name | 0.005 | 0.005 | 0.008 | 0.001 | 100.0 |
- Uncached Name | 0.007 | 0.104 | 0.358 | 0.110 | 100.0 |
- DotCom Lookup | 0.006 | 0.021 | 0.167 | 0.036 | 100.0 |

··· no official Internet DNS name ···
SUNBELT SOFTWARE

130. 89. 4. 21 | Min | Avg | Max |Std.Dev|Reliab%|

- Cached Name	0.006	0.006	0.007	0.000	100.0
- Uncached Name	0.010	0.075	0.297	0.086	100.0
- DotCom Lookup	0.011	0.038	0.159	0.040	100.0

dc1service.service.utwente.nl
University Twente

80. 95.160. 2 | Min | Avg | Max |Std.Dev|Reliab%|

- Cached Name	0.005	0.006	0.007	0.000	100.0
- Uncached Name	0.006	0.084	0.394	0.111	100.0
- DotCom Lookup	0.006	0.028	0.125	0.035	100.0

ns1.intermax.nl
Intermax BV

195. 13. 56.179 | Min | Avg | Max |Std.Dev|Reliab%|

- Cached Name	0.006	0.007	0.008	0.000	100.0
- Uncached Name	0.011	0.082	0.366	0.092	100.0
- DotCom Lookup	0.011	0.043	0.110	0.038	100.0

host4.karakas.be
OPENMINDS Network

193. 78.240. 12 | Min | Avg | Max |Std.Dev|Reliab%|

- Cached Name	0.007	0.007	0.009	0.000	100.0
- Uncached Name	0.013	0.092	0.429	0.101	100.0
- DotCom Lookup	0.014	0.046	0.179	0.050	100.0

cache0204.ns.eu.uu.net
VzB private LAN

195.129. 12. 83 | Min | Avg | Max |Std.Dev|Reliab%|

- Cached Name	0.007	0.008	0.010	0.001	100.0
- Uncached Name	0.013	0.085	0.428	0.102	100.0
- DotCom Lookup	0.014	0.047	0.172	0.050	100.0

cache0206.ns.eu.uu.net
End User Server Network

83.125. 8. 1 | Min | Avg | Max |Std.Dev|Reliab%|

- Cached Name	0.008	0.008	0.010	0.000	100.0
- Uncached Name	0.008	0.102	0.454	0.116	100.0
- DotCom Lookup	0.008	0.026	0.137	0.032	100.0

ns2.dus.net
dus.net GmbH

62. 32. 46.100 | Min | Avg | Max |Std.Dev|Reliab%|

- Cached Name	0.008	0.009	0.010	0.001	100.0
- Uncached Name	0.008	0.097	0.280	0.082	100.0
- DotCom Lookup	0.008	0.026	0.186	0.033	100.0

ns2.mach-six.com
Caucasus Network is an ISP based in Rep. of Georgi

8. 8. 8. 8 | Min | Avg | Max |Std.Dev|Reliab%|

- Cached Name	0.008	0.009	0.020	0.002	100.0
- Uncached Name	0.012	0.110	0.649	0.121	100.0
- DotCom Lookup	0.018	0.043	0.137	0.033	100.0

google-public-dns-a.google.com
Google Incorporated

8. 8. 4. 4 | Min | Avg | Max |Std.Dev|Reliab%|

- Cached Name | 0.008 | 0.009 | 0.020 | 0.002 | 100.0 |
- Uncached Name | 0.011 | 0.149 | 1.457 | 0.232 | 100.0 |
- DotCom Lookup | 0.017 | 0.040 | 0.136 | 0.026 | 98.0 |

google-public-dns-b.google.com
Google Incorporated

194. 98. 65. 65 | Min | Avg | Max |Std.Dev|Reliab%|

- Cached Name | 0.007 | 0.014 | 0.053 | 0.011 | 100.0 |
- Uncached Name | 0.012 | 0.103 | 0.426 | 0.120 | 100.0 |
- DotCom Lookup | 0.017 | 0.026 | 0.083 | 0.012 | 100.0 |

cache0300.ns.eu.uu.net
UUNET, a WorldCom Company

93. 88.144.138 | Min | Avg | Max |Std.Dev|Reliab%|

- Cached Name | 0.005 | 0.037 | 0.299 | 0.055 | 100.0 |
- Uncached Name | 0.009 | 0.078 | 0.401 | 0.097 | 100.0 |
- DotCom Lookup | 0.014 | 0.031 | 0.130 | 0.026 | 100.0 |

DNS1.UNIFIEDROOT.COM
Unified root address space

156.154. 71. 1 | DNS queries are not answered at this IP.

rdns2.ultradns.net
NeuStar

156.154. 71. 22 | DNS queries are not answered at this IP.

··· no official Internet DNS name ···
NEUSTAR

UTC: 2011-09-13, from 09:42:05 to 09:44:55, for 02:49.733

The following analysis has been made from observing the Benchmark results for the TNO Network.

- The TNO Network has multiple redundant nameservers configured.

The TNO Network is currently configured to use two separate nameservers for DNS name resolution. According to the GRC DNS Benchmark, this is best practice because DNS name resolution will be performed almost on at all times.

- All the TNO Network nameservers are actively replying to queries.

The TNO Network's two nameservers are replying to queries and working properly.

- TNO Network's nameservers are not optimally ordered.

According to the GRC DNS Benchmark, client machines use DNS servers in the order they are listed under the network adapter's properties, or when obtained automatically from an ISP, in the order provided by the ISP. The order of nameserver listing should match their order of decreasing performance which is not how the TNO Network is currently configured:

Usage Order	Nameserver IP	Speed Rank
1	208.67.222.222	2
2	208.67.220.220	1

The benchmark collected approximately one hundred and fifty DNS performance samples from each nameserver being tested. Although this is sufficient to generate a good average performance estimate, if the variance of the collection of sampled values is high, in other words, not a lot of agreement among samples, it is impossible to know with statistical certainty how individual nameservers compare to each

other. Therefore, even if the ranking shown above appears to be out of order, the differences may not be significant.

- The TNO Network resolvers are slower than 7 public alternatives.

There were seven available DNS Resolvers that were enlisted after running the GRC DNS Benchmark than those currently being used by the TNO Network.

- The TNO Network resolvers are 100% reliable.

During this benchmark test, all of the TNO Network nameservers tested returned a reply for every request sent.

- The TNO Network resolvers intercept name errors.

Some of the TNO Network's available resolvers intercept errors and redirect web browsers to a custom page in response to an invalid DNS lookup request or NXDOMAIN. This is used as a marketing effort to redirect mistaken web browser URL entries to the DNS provider's own advertising-laden marketing-related pages. However, some resolvers like OpenDNS allow customization of NXDomain redirection so that erroneous queries can be configured to return an error.

- The TNO Network resolvers are replying to all query types.

This means that the resolvers used reply to all types of queries as long as they are valid. This is also a desirable characteristic because the nameserver is able to respond to many unusual types of queries which may look invalid but are actually valid and have to be identified.

4 Implementation of a DNS Service

In this part of the report, a particular DNS Service, Content Filtering based on DNS that has been selected for implementation is presented. For the practical implementation, a component of Content Filtering based on DNS which is Blocking Advertisements is implemented.

One of the mechanisms to filter content on the internet using the DNS to block specific DNS resolution that relates to a specific content. This is possible because different contents that are loaded on a webpage when the domain is accessed have their own DNS request associated with them. The following steps show how Ad Blocking, a specific component of Content Filtering based on DNS, can be implemented either on a Client PC or at a Local DNS resolver.

4.1 Content Filtering from a Client PC

At a Client PC, if the client already knows which specific sites are usually associated with advertisements, the client can specify in the *Hosts* file how those specific websites should be resolved. For example, one of the most popular advertisement websites, *ad.doubleclick.net*, can be stopped from being resolved by editing the Hosts file located in *C:\Windows\System32\drivers\etc\hosts* in a Windows computer. For Windows to accept this change, Notepad should be opened as administrator using "*Run as Administrator".* The resolution needs to be specified so that the IP Address is written first and the domain name is written in the following column.

In many webpages of today, the most dominant advertisements from Google Ads can be blocked using the following resolution which redirects the resolution to a Loopback IP address like,

127.0.0.1 pagead.googlesyndication.com
127.0.0.1 pagead2.googlesyndication.com

or any other IP address that should be resolved in place of the advertisement from Google.

As indicated in Reference [10], one of the challenges of using Ad Blocking at the Client side Hosts file is that many advertisements use multiple third-level domains. For example, if *pagead.googlesyndication.com* is one site hosting advertisements, multiple third-level domains can include *pagead2.googlesyndication.com* or *pagead3.googlesyndication.com*. And since it is not directly possible to use a wildcard (*) in *Hosts* file, Ad Filtering at the Local DNS server is a better mechanism than implementation at the Client PC.

4.2 Content Filtering from a Local DNS Server running Ubuntu Server and BIND

First Ubuntu Server has to be configured to act as a DNS server as indicated in Reference [57] and [58]. Then, at the Local DNS server running Ubuntu Server, to perform Ad Blocking with BIND, a file in */etc/bind/zones/* called *blockanad.com.db* needs to be created as

```
sudo nano /etc/bind/zones/blockanad.com.db
```

This zone definition is where I have put all the addresses and machine names that the DNS server needs to know. The file is configured as

```
$TTL 24h

@ IN SOA localhost. root.localhost. (
        2011100701 86400 300 604800 3600 )

@ IN NS localhost.
@ IN A 127.0.0.1
* IN A 127.0.0.1
```

When saving the file in */etc/bind/zones/* as *blockanad.zone*, BIND requires that there is line break at the end of the file, and it will consider it as an error if no line break is provided at the end of the file.

Then, */etc/bind/named.conf* is opened and edited by leaving out or commenting the following line which corresponds to *rndc* that is the command program used to control BIND as

```
keys{ “rndc-key”;};
```

to

```
//keys{“rndc-key”;};
```

For each domain to block, a line at the end of */etc/bind/named.conf* is added. For example, here we are blocking *Advertisements from Google* as

```
zone “googlesyndication.com” {
        type master;
        file “blockanad.zone”;
        };
```

Then BIND is restarted as

```
sudo /etc/init.d/bind9 restart
```

Technical Analysis of the Implemented Service

With the particular Content Filtering service implemented, it is expected that the response time to the particular domain might take a little longer than before the service is implemented. In order to check this by measuring the *response time* to the particular domain, the GRC Benchmark that is discussed in Section 3 has been used. With Ad Blocking, the first visit to the domain takes an extra one and half to two seconds, which is sometimes noticeable by the user. However, *subsequent* visits to the same domain are faster because of the effect of caching.

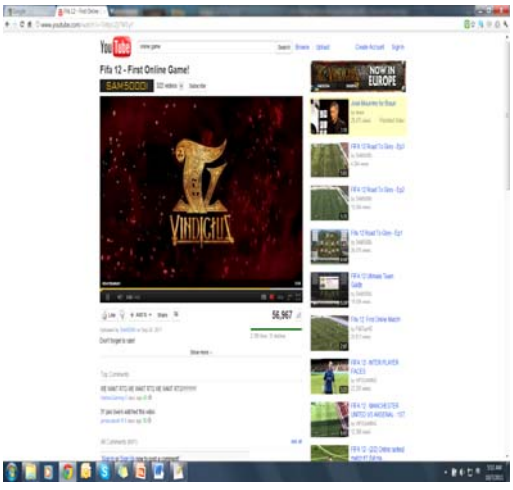
Results

The Implementation of DNS Based Content Filtering has been implemented to filter advertisements from two websites which mostly likely contain advertisements. These websites are Youtube and Livestation, References [59] and [60].

The following shows the results of Implementing the Content Filtering DNS Service to filter out Advertisements from Google on www.youtube.com.

1 Youtube.com

Before (With Google Ads, actual video takes time to load)



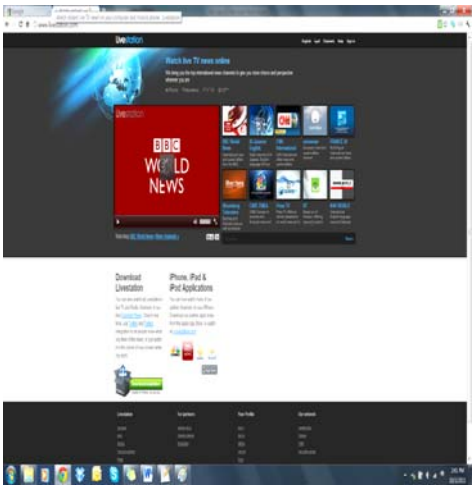
After (Google Ads Filtered Out, video loads immediately)

The following shows the results of Implementing the Content Filtering DNS Service to filter out Advertisements from Google on www.livestation.com.

2 Livestation.com

Before
(With Google Ads)

After
(With Google Ads Filtered Out)



5 Conclusion

In this report, an inventory and classification of existing DNS services and infrastructure as currently used in the internet has been presented. This explains alternative ways of using the DNS infrastructures and contributes to the understanding of the impact of DNS in the internet of today, and the its future usage. Moreover, it is hoped that it fills the gap in the current available literature concenrning a proper inventory and classification of existing DNS services and infrastructure as currently used in the internet.

Furthermore, a technical analysis on the performance of existing DNS resolvers is presented. This is based on comparing performance samples from DNS resolvers that were tested from the TNO Network using GRC Research's DNS Benchmark. This is regarded as important since it can be used to analyse the performance of a list of available DNS resolvers from a given location, and it can also be used to check how much the performance of a given DNS resolver is affected, interms of its speed of response, after a particular DNS service is deployed on the DNS resolver.

Regarding the implementation of a DNS service, advertisement blocking, a specific service from Content Filtering based on DNS, has been implemented. The Effect of such implementation has been checked using the GRC DNS Benchmark and it has been noted that the implementation of the service does not largely affect the performance of the DNS resolution and there were little noticeable effects from the user's point of view.

The *integration* of the enlisted DNS services and infrastructure requires some points to consider. For examaple, DNSSEC and DNS Filtering could be integrated together to help the performance and security of the DNS Resolution. This is done in such a way that in cases where there is *no need* to connect *securely*, using DNSSEC, to illegal or insecure web sites, those web sites could be *filtered out* in the first place using DNS Filtering. This shows the importance of DNS Filtering via reputation, blacklists and whitelists as a necessary function. Regarding DNSSEC in particular, considering X.509 and existing PKI systems, the development of the future DNSSEC will help alleviate the need for contacting a third party certificate authority.

Having the DNS services and infrastructure discussed in the paper, a question comes in mind; What is next in the evolution of the DNS and Internet name spaces? The DNS can enable new applications in areas that have similar globally distributed structure, which are hierarchical and could make use of a dynamic database. In such a way, the Internet name space can include a number of other things that exist physically and that could be routed from one place to another.

6 Recommendation and Future Work

There are a number of possibile areas where the DNS can be used in the future. It can be a basis for the services/infrastructure enlisted in this paper, and furthermore, it can be made use of in areas which have similar distributed infrastructure like Refereces [53], [54] and [55]. Some of these could be financial inquiries, payment systems, search and discoveries. In some of these areas, DNSSEC could be used as a security mechanism for services which require authentication and data integrity.

The DNS Based Content Filtering service implemented in this paper can be improved in future work so that it can make use of a *classifier* to filter content automatically. This could work by correlating the use of tokens like specific words from the URL and calculating their inference to the undesired domain with a probability. Such kind of techniques are currently used to classify emails and identify spam, like what is used with *Bayesian Spam Filtering*, as indicated in Reference [56], however they have not yet been implemented in DNS Based Content Filtering. It would be interesting to see their effects in making the web a safer place.

7 References

[1] http://en.wikipedia.org/wiki/Domain_Name_System

[2] http://www.opendns.com/about/announcements/213

[3] http://www.opendns.com

[4] http://en.wikipedia.org/wiki/Google_Public_DNS

[5] http://www.webmasterview.com/2009/12/opendns-vs-google

[6] *Enhanced DNS*, Akamai Technologies, 2009.

[7] *DNSSEC for Enhanced DNS*, Akamai Technologies, 2010.

[8] *Delivering Next-Generation Solutions for Nonstop Core Network Services*, Infoblox Grid Technology, Infoblox, 2009.

[9] George Ou, *DNS Filtering is Essential to the Internet*, High tech forum Jun 2011.

[10] http://www.patrickpatoray.com/index.php?Page=105

[11] http://en.wikipedia.org/wiki/DNSBL

[12] Anirudh Ramachandran, David Dagon, and Nick Feamster**,** *Can DNS-Based Blacklists Keep Up with Bots*, College of Computing, Georgia Institute of Technology, 2006.

[13] Carsten Strotmann, *Using DNS based Blacklists to stop SPAM*, Men & Mice, 2004.

[14] http://www.dns-sd.org

[15] S. Cheshire, M. Krochmal, *DNS-Based Service Discovery*, Apple Inc., Feb 2011.

[16] P. Belimpasakis, A. Saaranen and R. Walsh, *Home DNS: Experiences with Seamless Remote Access to Home Services*, IEEE, 2007.

[17] Ximin Zhang, Junding Sun, Lihua Zhou, *Development of an Internet Home Automation System using Java and Dynamic DNS Service*, PDCAT, 2005.

[18] P. Faltstrom, E.164 number and DNS, Request for Comments: 2916, Sep 2000.

[19] *.tel, An Innovative Use of the DNS in .tel technology*, Telnic, 2011.

[20] Patrik Fältström, Olav Kolkman, *Telephone numbers in the dns – ENUM*, Cisco Systems, May 2003.

[21] Anees Shaikh, Renu Tewari, and Mukesh Agrawal, *On the Effectiveness of DNS-based Server Selection*, 2001.

[22] Jenq-Shiou Leu, Hui-Ching Hsieh, Yen Chiu Chen, and Yuan-Po Chi, *Design and Implementation of a Low Cost DNS-based Load Balancing Solution for the SIP-based VoIP Service*, IEEE Asia-Pacific Services Computing Conference, 2008.

[23] Microsoft Exchange Server Clustering, *The pros and cons of a DNS-based front-end Exchange Server cluster*, Apr 2006.

[24] Edgedirector Global Server Load Balancing (GSLB), http://edgedirector.com/how/load.htm

[25] Pete Tenereillo, *Why DNS Based Global Server Load Balancing (GSLB) Doesn't Work*, May 2004.

[26] www.multicastdns.org

[27] Jaehoon Jeong, Jungsoo Park and Hyoungjun Kim, *DNS Name Service based on Secure Multicast DNS for IPv6 Mobile Ad Hoc Networks*, 2004.

[28] Internet Engineering Task Force, *Multicast DNS*, Feb 2011.

[29] http://en.wikipedia.org/wiki/Zero_configuration_networking

[30] Jaeyeon Jung, Emil Sit, Hari Balakrishnan, and Robert Morris, *DNS Performance and the Effectiveness of Caching*, Oct 2002.

[31] Poornima College of Engineering, Jaipur, Department of Computer Science, *A Report on Cooperative DNS (CoDNS)*.

[32] KyoungSoo Park, Vivek S. Pai, Larry Peterson and Zhe Wang, *CoDNS: Improving DNS Performance and Reliability via Cooperative Lookup*, 2004.

[33] Lindsey Poole, Vivek S. Pai, *ConfiDNS: Leveraging Scale and History to Detect Compromise*, 2008.

[34] Edith Cohen, Haim Kaplan, *Proactive caching of DNS records: addressing a performance bottleneck*, Oct 2002.

[35] http://www.grc.com/dns/benchmark.htm

[36] Alexa Web Information Company, www.alexa.com

[37] ComScore, www.comscore.com

[38] Nielsen NetRatings, www.nielsen-netrating.com

[39] Moheeb Abu Rajab, Fabian Monrose, Andreas Terzis, and Niels Provos, *Peeking through the Cloud - DNS-based estimation and its applications*, Johns Hopkins University, Google Inc., 2008.

[40] http://en.wikipedia.org/wiki/Dynamic_DNS

[41] P. Vixie, S. Thomson, Y. Rekhter, J. Bound, RFC2136, *Dynamic Updates in the Domain Name System (DNS UPDATE)*, Network Working Group, Apr 1997.

[42] Antonio Lioy, Fabio Maino, Marius Marian, Daniele Mazzocchi, *DNS Security*, Politecnico di Torino, May 2000.

[43] Giuseppe Ateniese, Stefan Mangard, *A New Approach to DNS Security (DNSSEC)*, ACM, Nov 2001.

[44] SAC 032, *Preliminary Report on DNS Response Modification, DNS based NXDomain Redirecting*, ICANN Security and Stability Advisory Committee, Jun 2008.

[45] David Plonka, Paul Barford, *Context-aware Clustering of DNS Query Traffic*, Oct 2008.

[46] S. Castillo-Perez, J.Garcia-Alfaro, *Anonymous Resolution of DNS Queries*, Universitat Autonoma de Barcelona, Universitat Oberta de Catalunya, 2008.

[47] EPC global, *Object Naming Service version 1.0*, Oct 2005.

[48] J.G. Alfaro, M. Barbeau, E. Kranakis, *Evaluation of Anonymized ONS Queries*, 2008.

[49] http://barbariangroup.com/posts/2322-howto_create_a_transparent_local_apple_software_update_server

[50] Rodolfo Villaca, Fabio Luciano Gerdi, Mauricio Ferreira Magalhaes, *Context-Based Name Resolution Service for the Next-Generation Internet*, 2009.

[51] http://www.nameshield.net/public/comparative-analysis-of-dns-versus.html

[52] http://www.dns-sd.org/ServerSetup.html

[53] S. Karnouskos, A. Vilmos, P.Hoepner, A. Ramfos, N. Venetakis, *Secure Mobile Payment – Architecture and Business Model of SEMOPS,* Oct 2003.

[54] *SEMOPS Broschure*, SEMOPS H-1022, Budapest, Hungary, 2008.

[55] Teppo Halonen, *A System for Secure Mobile Payment Transactions*, Helsinki University of Technology, Telecommunications Software and Multimedia Laboratory Jan 2002.

[56] http://en.wikipedia.org/wiki/Bayesian_spam_filtering

[57] https://help.ubuntu.com/community/BIND9ServerHowto

[58] https://help.ubuntu.com/8.04/serverguide/C/index.html

[59] www.youtube.com

[60] www.livestation.com

8 Appendix

The points which are used to set up Bonjour name server from Apple Inc. can be used as an example for DNS-SD as described in Section 2 of this report. They have been included in the Appendix of this report as follows.

Configuring BIND

A Bonjour name server can be configured by editing the BIND configuration file (/etc/named.conf) as follows.

```
options {
        // specifying the directory where zones files are located directory "/var/named";
        // to answer to DNS queries outside the bonjour.example.com. zone put addresses of normal
        // DNS servers
        forwarders { 1.2.3.4; 5.6.7.8; };
};
// entry for the new zone
zone "bonjour.example.com." {
        type master;
        file "db.bonjour.example.com";
        allow-update { key bonjour.example.com.; };
};
// key for updating the zone
key bonjour.example.com. {
    algorithm hmac-md5;
    secret "CnMMp/xdDomQZ4TelKIHeQ==";
};

// If a shared secret is not to be used, the line allow-update should be replaced with:  allow-update { any; };
// to create a key per user, and avoid having to include the entire list of keys explicitly in the "allow-update"
// use an "update-policy" declaration instead, like:  update-policy { grant * wildcard *.bonjour.example.com.;
};
```

Writing the Zone File

Next a zone has to be created using a file named db.<zone>, for example, "db.bonjour.example.com" and it needs to be copy it into /var/named. The only change is to replace the two instances of "wab.example.com." with the hostname of the server.

```
$TTL 3600 ; One hour default TTL

; Replace wab.example.com. below with the machine's hostname
@ IN SOA wab.example.com. unused-email (
```

```
2011100701    ; serial number (Year, Month, Day and No. of Change)
10800         ; refresh (3 hours)
3600          ; retry   (1 hour)
604800        ; expire (1 week)
60            ; minimum (1 minute)
)

; Specify the server as the nameserver for the zone, substituting the machine's hostname.
@ IN NS wab.example.com.

; Specify the server which handles DNS updates for the zone, typically port 53 on the same server as above.
; to run dnsextd then the port number is typically 5352
_dns-update._udp IN SRV 0 0 53 wab.example.com.
; Add PTR records telling clients that they can browse and register here
b._dns-sd._udp  IN PTR @          ;  "b" = browse domain
lb._dns-sd._udp IN PTR @          ; "lb" = legacy browse domain (include domain in empty-string browses)
r._dns-sd._udp  IN PTR @          ;  "r" = registration domain
```

Discovering the Server

If there is access to the parent zone's DNS server, it is possible to delegate the new zone to the new server by adding an entry in the example.com. zone file:

```
bonjour.example.com.    86400 IN  NS      wab.example.com.
```

If there is no access to the parent zone's DNS server, it is also possible to simply add the IP address of the new server to the "DNS Servers" field of the Networking Preference Pane in each client computer. However, clients should learn the DNS server for a given domain by following the chain of delegation (NS records) from the root and not by manual configuration. Likewise, for reliable operation, the subdomain should be properly delegated from its parent.

Discovering Domains

Computers running Mac OS X Tiger or later and computers running Bonjour for Windows will issue domain enumeration queries to automatically discover, browse and register domains on the network. The easiest way for clients to discover the domain is by creating PTR records pointing from the DHCP domain name to the new zone. This requires administrative control of that domain. For example, if the DHCP "Domain Name" option (option code 15 [RFC 2132]) that the DHCP server sends to its clients is "example.com", then it is needed to create the following entries in the "example.com" zone file to tell those DHCP clients about the new "bonjour.example.com" domain:

```
; Added for applications to discover the domain as a potential place to browse
b._dns-sd._udp.example.com.  3600 IN PTR bonjour.example.com.
; The domain to be chosen is added as the default browse domain in the Bonjour Preference Pane
db._dns-sd._udp.example.com. 3600 IN PTR bonjour.example.com.
; Added for this domain to show up in the list of potential registration domains
```

```
r._dns-sd._udp.example.com.  3600 IN PTR bonjour.example.com.

; Added for the domain to be chosen as the default registration domain in the Bonjour Preference Pane

dr._dns-sd._udp.example.com. 3600 IN PTR bonjour.example.com.

; Added the following line so that applications that do empty-string domain browses will browse the zone in addition to "local."

lb._dns-sd._udp.example.com. 3600 IN PTR bonjour.example.com.
```

If there is no administrative control of that domain, it is possible to manually force a client to "discover" the new "bonjour.example.com" domain by adding it to the "Search Domains" field in the Network Preference Pane on each client. This will only work if there are the domain enumeration PTR records in the bonjour.example.com zone as shown in the "db.bonjour.example.com" zone file above.

Starting named

A backup of the zone file has to be created before running named for the first time. Once running the server with DNS Update turned on, it is not possible to edit the zone files by hand. If it is needed to reset the zone for any reason, simply revert to the saved copy, delete any .jnl files, and restart named and dnsextd.

```
named normally runs with no arguments:

root# named
```

Then, the syslog (/var/log/system.log) should be checked for errors. One can ignore any errors that say "/private/etc/rndc.key: file not found" or "couldn't add command channel". If any other errors occur, periods need to be put in exactly the right places in all files. For debugging, one may wish to run it in the foreground, with enhanced logging:

```
root# named -g -d 5
```

Starting dnsextd

The dnsextd daemon communicates with named using a shared secret which is specified on the command-line:

```
root# dnsextd -z bonjour.example.com.
```

If one uses DNS-SEC authentication, the key name has to be entered as well as the shared secret:

```
root# dnsextd -z bonjour.example.com. -k bonjour.example.com. CnMMp/xdDomQZ4TelKIHeQ==
```

To run in the foreground with verbose logging for debugging, add "-vf". Run with a single argument, "-h", for help and a full list of options.

Configuration Clients

At this step, each client on the network can be configured to use the new server.


TNO innovation for life

DNS Services, alternative ways of using DNS infrastructures

Project Name: Future DNS

Internship Period: Jul – Sep 2011

By: Brook Abegaz, 7th Oct, 2011

Supervisors: Dr. Ir. M. Oskar van Deventer,
Ir. Bart Gijsen


TNO innovation for life

Introduction

- The Domain Name System (DNS) is a vital component of the internet
- The DNS, a hierarchical distributed naming system for computers, services or any resource connected to the internet
- A number of value added services on or inside the DNS
  - Blacklisting , Content Filtering, Dynamic DNS, Estimation
  - Parental Control, Performance Improvement, Redirection
  - SW Version Updating, User and Infrastructure ENUM
- A proper inventory and classification of DNS Services and Infrastructure not available in literature
- The internship consisted of: -
  - A proper inventory and classification of DNS Services and Infrastructure,
  - A technical analysis of DNS resolvers,
  - A practical implementation of DNS Service.


TNO innovation for life

Inventory of DNS Services and Infrastructure

- Alternative ways of using DNS Infrastructure
- Inventory of twenty DNS Service/Infrastructure
    - Blacklisting/Whitelisting, Content Filtering, Parental Control
    - Estimation, Clustering, Anonymous Resolution of DNS Queries
    - Multicast DNS, Service Discovery, Home Remote Controlling
    - Telephone Number Mapping (ENUM), Software Version Updating
    - Performance Improvement, Load Balancing, Server Selection
    - Dynamic DNS, NXDomain Redirection, Security
    - Context Aware Naming, Object Naming Service


TNO innovation for life

Classification of DNS Services and Infrastructure

- Proper Classification of DNS Services/Infrastructure for easier Implementation
- Identification of where the DNS Service/Infrastructure can be implemented
    - Client Side: Blacklisting, Content Filtering, Parental Control, Home Remote Controlling, Telephone Number Mapping (ENUM), Multicast DNS, Object Naming Service and Software Version Updating
    - Resolver Side: Blacklisting, Content Filtering, Parental Control, Telephone Number Mapping (ENUM), Multicast DNS, Software Version Updating, Server Selection, Load Balancing, Performance Improvement, Estimation, Security, NXDomain Redirection and Clustering
    - Server Side: Service Discovery, Home Remote Controlling, Telephone Number Mapping (ENUM), Load Balancing, Performance


TNO innovation for life

Technical Analysis on Performance of DNS Resolvers

- Running the GRC DNS Benchmark from TNO Network; OpenDNS, Google's Public DNS
- Publicly available DNS resolvers reliably faster than TNO Network resolvers.

GRC DNS Benchmark (Response Time)

TNO Network's OpenDNS Resolvers

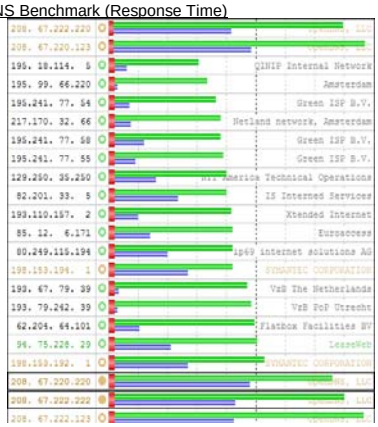
208. 67.222.222	Min	Avg	Max	St.Dv	Relb%
- Cached Name	0.004	0.005	0.007	0.001	100.0
- Uncached Name	0.005	0.160	1.292	0.265	100.0
- DotCom Lookup	0.006	0.086	0.161	0.056	100.0

Better Alternative OpenDNS Resolver

208. 67.222.220	Min	Avg	Max	St.Dv	Relb%
- Cached Name	0.003	0.003	0.005	0.001	100.0
- Uncached Name	0.004	0.159	1.088	0.242	100.0
- DotCom Lookup	0.006	0.083	0.272	0.063	100.0

- Current resolvers are not in proper order

Usage Order	Nameserver IP	Speed Rank
1	208. 67.222.222	2
2	208. 67.220.220	1




TNO innovation for life

Implementation of a DNS Service

- Blocking Advertisement from *Google Ads*
- Content Filtering from Client Machines
  - Editing the *"Hosts"* file like

```
127.0.0.1  pagead.googlesyndication.com
127.0.0.1  pagead2.googlesyndication.com
127.0.0.1  pagead3.googlesyndication.com
```

- Content Filtering from Client Machines
- Delegating the zone in BIND */etc/bind/zones/* called *blockanad.com.db* as

```
$TTL 24H
@ IN SOA localhost.root.localhost. (
        2011100701 86400 300 604800 3600 )
@ IN NS localhost.
@ IN A 127.0.0.1
* IN A 127.0.0.1
```

- Then at */etc/bind/named.conf* the domain to block is defined as

```
zone"googlesyndication.com"{type master; file
```

Results

- Results of Implementing the Content Filtering DNS Service
  - Advertisements have been filtered out

1 **Livestation.com**

Before *After*

With Advertisements form Google *With Advertisements from Google Filtered Out*

Results

- Results of Implementing the Content Filtering DNS Service
  - Advertisements have been filtered out

2 Youtube.com

Before *After*

With Google Ads, actual video takes time to load *Google Ads Filtered Out, actual video loads immediately*


TNO innovation for life

Results

- An inventory and classification of DNS Services/Infrastructure
  - Classification based on where to implement
- A technical analysis of DNS Resolvers
  - Alternative, faster and more performing resolvers
  - Effects of caching analysed
- Implementation of a DNS Service
  - Content Filtering using DNS implemented both from Client machine and Local DNS resolver to block advertisements
  - Effect of such implementation on the performance of DNS resolution checked by DNS Benchmark


TNO innovation for life

Conclusion

- A number of alternative Services/Infrastructure on DNS with good monetary value
- Implementation of a Service/Infrastructure keeping up with the required Performance


TNO innovation for life

Recommendation and Future Work

- How to develop better mechanisms for Content Filtering Based on DNS
- Future DNS Services
- DNSSEC as a security mechanism for services which require authentication and data integrity


TNO innovation for life

Thank You!