Domain Name System

Ahmed Almohsin
amalmohsin21@tntech.edu

ABSTRACT
In this paper, we will discuss What is Domain Name System (DNS) and How does it work?

General Terms
History, Design, Components, Mechanisms.

Keywords
DNS, OS.

1. INTRODUCTION

The naming system for the networked (Interconnected devices) systems is referred as Domain Name System (DNS). It is mostly used in Microsoft Windows Server Operating Systems and it is part of TCP/IP protocol suite. DNS converts human readable web address to machine readable IP addresses. It is the huge database which contains the IP addresses of the different domains. List of domains with their IP addresses are cached and updated frequently. DNS is used as domain controller location mechanism in Windows Server 2003 Active Directory services. For Active Directory operations such as, authentication, updating, or searching, Windows Server 2003 machines use DNS to locate Active Directory domain controllers and these domain controllers use DNS to locate each other.

2. History

HOSTS.TXT file was used before implementation of the DNS to map the host names to their addresses. The origin of the DNS was on the basis of the observation of HOSTS.TXT file which was previously used and centrally maintained and distributed by the SRI-NIC. This text file used to be transferred directly or indirectly via the Internet, but it did not work out very well, also because of the increasing use of the Internet HOSTS.TXT file sharing and maintaining at central level was cumbersome. The initial design of the DNS was specified in [RFC 882, RFC 883].

3. Design

The base design theory for the DNS was to provide at least all of the same information as HOSTS.TXT. It should also allow the database to be maintained in a distributed manner. DNS should not have any size limit for names, its components and the data associated with the name. DNS should inter-operate DARPA and across the DARPA Internet in as many enrollments as possible. The new system (DNS) should be independent of network topology and should be capable of other name spaces. In order to be globally acceptable the system should avoid trying to force a signal OS architecture or organizational style.

3. Components

The DNS has two major types of components, Name Servers and Resolvers. Name servers are the repositories of information and answer queries using all the information they posses. Resolvers are the interfaces to client programs and body the algorithms necessary to find a name server that has the information sort by the client. These both functions may be combined or separated to suit the need of the enrollment. In many cases it is useful to centralize the resolver function in one or more specialized name server for an organization. This kind of structure shares the use of cached information and also allows which makes communication faster.

The DNS internal name space is of variable depth tree structure where each node in the tree has an associated label. The domain name of the node is the concatenation of all the possible paths from the node to the root of the tree (Where DNS database resides). Labels are variable length string of octets with each can be of any 8-bit value. The label with 0 lengths is reserved for the root. The name space searching operation is not case sensitive so google.com would match GOOGLE.COM.

4. Mechanisms

DNS provides two major mechanisms i.e. zones and caching for transferring data from its ultimate source to its ultimate destination. Zone is a complete description of the contiguous section of the total tree structured name space, together with some other pointer information to other contiguous zones. Where zones are sections of the system wide database which are controlled by the specific organization and that organization is also responsible for distributing latest copies of the zones to multiple which makes the zones available. The DNS resolvers with combined name server/resolver programs cache the responses for the use for future queries. TTL (Time-To-Live) is the mechanism used for controlling the caching process. TTL field is attached to each RR (Resource Record) and it is in units of seconds which informs till what time response can be reused. The low TTL minimizes the periods of transient inconsistency, while high TTL minimizes traffic and allows caching.

As DNS has hierarchical structure it is similar as the directory structure which starts from the root and has mapping of all the nodes (Folders) connected to the root so it has exact address of the particular node (file within sub-folders) of the Operating System. When we search for the file/folder in windows it searches for the folder and gives us the folder with the path in exactly same manner DNS works to locate domains or the resources connected to the Internet.

5. How does it Work?

DNS has an associated set of protocols that define the mechanism for querying and updating the database and mechanism for replicating the information as well as schema of the database. The DNS protocol consists of different types of DNS messages that are processed according to the information in their message fields.

There are three different types of DNS messages Queries, Responses and Updates. DNS queries can be sent from a DNS client (resolver) to a DNS server, or between two DNS servers. A DNS query is a request for DNS resource records of a specified RR type with a specified DNS name.

Two types of DNS queries are recursive and Iterative. Recursive query forces DNS server to respond to request either with success or failure response. When a DNS server processes recursive query with the failure response, the query must be sent to a root DNS server, where root server again responses with success or failure. For an iterative query the DNS server is expected to respond with the best local information it has, based on what it knows from local zone files or from caching. If a DNS server sends a negative response then it does not have any local information that can help to resolve the query.

6. Sample Python Script

```
#!/usr/bin/python

import socket

import sys

a = socket.gethostbyname(sys.argv[1])

h = socket.gethostbyaddr(a)[0]

print '\nName: ' + h +'\tIP: ' + a
```

Output:

```
mosin@ubuntu:~/Desktop$./DNS.py
www.ebay.com

Name: hp-core.ebay.com        IP: 66.211.181.11
```

2. ACKNOWLEDGMENTS

Special thanks to ACM and Microsoft for sharing their resources about DNS.

3. REFERENCES

[1] Jaeyeon, Jung., Emil Sit., Hari, Balakrishnan., Robert, Morris., (2002). DNS Performance and the Effectiveness of Caching. IEEE/ACM Transactions on Networking (TON). Vol 10, Issue 5. pg 589-603. http://delivery.acm.org/10.1145/590000/581877/01041066.pdf?key1=581877&key2=0245257521&coll=GUIDE&dl=GUIDE&CFID=61563787&CFTOKEN=40296049

[2] Microsoft TechNet. (2003). What Is DNS? http://technet.microsoft.com/enus/library/cc787921(WS.10).aspx

[3] Microsoft TechNet. (2003). How DNSWorks. http://technet.microsoft.com/enus/library/cc772774(WS.10).aspx

[4] Mockapetris, Paul V., Dunlap, Kevin J., (1995). Development of the Domain Name System. ACM SIGCOMM Computer Communication Review. Vol 25, Issue 1. pg 112-122. http://delivery.acm.org/10.1145/210000/205459/p112-mockapetris.pdf?key1=205459&key2=1906257521&coll=GUIDE&dl=GUIDE&CFID=61566306&CFTOKEN=35143164