Domain Name System

Justin Groce
Systems Programming
jbgroce21@tnech.edu

ABSTRACT

Domain Name System (DNS) is one of the most used and most essential services the internet uses. People like to think that computers automatically know what to do when that person enters in their favorite website address. The truth is that the task of connecting to a website is quite a complex task. Even simpler than that, connecting to another computer is complex. People from any country are used to identifying people and places by their names. When one makes plans to go to a local restaurant, they don't give out the latitudinal and longitudinal coordinates of the restaurant. It sounds a little odd when someone calls and says, "Hey! I'll meet you at 36 degrees 7 minutes North by 86 degrees 41 minutes West tonight." They simply give the name of that restaurant. It is similar when connecting one computer to another. People want to give the "name" of the computer instead of the 32 character long group of 1's and 0's. Usually the lowest level name a person will know is the dotted decimal notation of that number, which is still a human easy abbreviation called an IP address. Computers themselves cannot understand these abbreviations. Computers need the group of 1's and 0's. Therefore, something needs to translate from human readable names to computer readable numbers. DNS fills that need.

Keywords

DNS, Network, IP Address.

1. What is Domain Name Service

Computers work exclusively with 1's and 0's. There are many levels of hardware and software between what users see and what the machines actually use. The most important of these levels is Domain Name System (DNS). DNS provides the ability for humans to name computers with human readable names. Also, DNS allows humans to use the above mentioned names to refer to computers on a network. These computers could be storing information about almost anything. The most common use of DNS is when referring to a website.

To look up a website, you need its name (called a domain name). As detailed in the abstract, the name is a 32 character long group of 1's and 0's. That name is called an IP address. Every domain name has a specific computer that stores the information contained on that site. When you type in the name of your favorite website into a web browser, your browser is directed to the machine with the IP address DNS has listed for that address [1].

Websites are broken up into different tiers. For instance the website "www.csc.tntech.edu" is comprised of four tiers. The tiers are separated by the periods. The first tier is "www," then "csc," then "tntech," and last "edu." These different tier help locate the computer with the information to represent the website. The separation begins with the highest tier being the last three characters of the domain name. This allows the computer to be directed first to the server corresponding to "edu" addresses, then to the server corresponding to "tntech" addresses, and so on [3].

In the case of "www.csc.tntech.edu," the top tier is "edu." This tells you that the website "www.csc.tntech.edu" belongs to an educational institution. The reason that the domain name is assumed to belong to an educational institution is because the tag "edu" is reserved exclusively for educational institutions. There are other reserved tags that make up this top tier. For example, "com" is used for commercial organizations; "gov" is used for federal agencies in the United States; "mil" is used for American military forces such as the army, navy, air force, and marines; "org" is used for domain names that do not necessarily fit into the other categories [2]. These are not all of the top tier tags, but they serve as an example of how domain names are broken into groups to make the DNS servers be able to find any domain name quickly and efficiently. Figure 1 further illustrates the separation of the tiers.

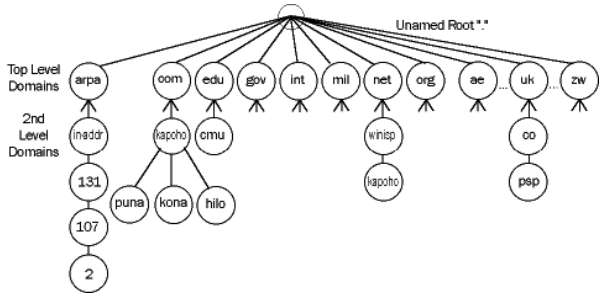

Figure 1 [2]

2. Implementation

As figure 1 clearly shows, DNS servers are set up in a hierarchal structure. This is so that the millions of website on the internet can be delivered to users with the expected speed. People do not want to wait minutes or longer just to view a website. The first efforts at a DNS-like protocol used a file that kept track of names and corresponding IP addresses. That clearly would not work with the multitude of computers and internet compatible devices today. The hierarchal scheme of DNS allows servers to be spread across the world so that they can direct users to the necessary computer, the "host," in a timely manner.

DNS servers are set up to know which other servers can complete the requested domain if they do not control the specific domain the user requested. The server that is connected the closest to the host computer is said to have authority for that domain [4]. When the user enters the domain name into the browser, a request is sent to the nearest DNS server. Either that server has the IP address of the domain name sent to it, or that server sends the request on to the lowest tier it can according to the tags in the domain name and the other servers it knows about. If the request was sent on, the next server does the same thing. Eventually the request will end

up at an authoritative server and the information will be sent back through the chain of servers to the computer that requested the domain. This is important because the servers cache (store information about) that data about the site so that if any computer requests the same domain name from one of the servers in the chain, then they can give an answer back without sending the request all the way back to the authoritative server. Caching speeds up the process of retrieving domain information because people tend to look up the same addresses multiple times. Also, this frees up the servers close to the authoritative server to answer other requests.

An example of how to get the IP address corresponding to a particular domain name using python follows.

```
import dns.resolver, sys

name = ''
if (sys.argv.length < 2):
        name = 'csc.tntech.edu'
else
        name = sys.argv[1]

answers = dns.resolver.query(name, 'ANY')
for rdata in answers:
    print 'Host', rdata.exchange
```

One does not have to use python. There are many languages that have the ability to retrieve DNS information. For example, a script using ruby follows.

```
dns = Resolv::DNS.new
name = ""
if ARGV.legth < 1
        name = "csc.tntech.edu"
else
        name = ARGV[0]
dns.getresources(name, Resolv::DNS::Resource::IN::ANY).collect do |r|
  puts r.exchange.to_s
```

3. Conclusion

People use the internet now more than ever. People can even reach the internet with their cell phones. Anyone connected to the internet has the options to choose from well over a million websites. Also, millions of people are accessing those domains at the same time. DNS provides the opportunity for all of those people to connect to their desired domains very quickly and without even knowing what the actual address is.

DNS is a phonebook of sorts for domain names. When one wants to call their friend, do they have to look up the phone number of that friend? Think about remembering the number of everyone in a town, a state, or even a country. How about the addresses of the homes of those people? This becomes much too difficult very quickly. Because of this, phonebooks are available to look up the number or address of the person you are trying to reach. It is the same concept with the addresses of domain names. Finding the name of a website is relatively simple, usually because one knows what website they are looking for. With DNS, one can just type the name in and all the work is done for you. You don't even have to flip through the pages of the "DNS phonebook!"

DNS is essential for the number of domain name lookups that happen every single day around the world. It keeps users connected to the world.

4. REFERENCES

[1] Brain, Marshall. *What is DNS?* (Oct 2010) http://www.howstuffworks.com/dns.htm.

[2] Lee, Thomas and Joseph Davies. *Domain Name Service (DNS)*. (Oct 2010) http://technet.microsoft.com/en-us/library/bb726935.aspx.

[3] Malone, David. *The Root of the Matter: Hints or Slaves.* (Oct 2010) http://delivery.acm.org.ezproxy.tntech.edu/10.1145/1030000/1028791/p15-malone.pdf?key1=1028791&key2=9726336821&coll=ACM&dl=ACM&CFID=104573925&CFTOKEN=13029875.

[4] Rader, Ross. *One History of DNS.* (Oct 2010) http://www/byte.org/one-history-of-dns.pdf.