# How does DNS server work

Source: <http://www.slashroot.in/how-dns-works>

Hi allsmiley...in this post we will be discussing the most important and inevitable resource in the world of internet, which each and everyone of us uses knowingly or unknowingly.

Each and every machine in a network is identified by a numerical address. This address can be used by other machines in the network to communicate with each other. But without a relevant name associated with that numerical address, it will be very difficult to memorize  the numerical address of all the machines in the network. Even for a handful of hosts in a local network, it will be very difficult to memorize all the numerical address of each of them, so forget about memorizing numerical addresses of more than 400 million hosts in the internet.

initially the concept of host file was born to solve the problem, each and every machine in the network used to have a host file, /etc/hosts where the name to address mapping was done.. But with the passage of time,problems like the below emerged.

1.each and every machine needed to update the the newly added entries themselves.

2.there was no kind of notification available for clients to know a new entry has been added.

3.by the passage of time, a single file became large and very large, making it difficult to handle.

During the mid 1970's the concept of name servers came into place. the basic idea behind this name servers was that, people find it easy to remember names rather than numbers, especially when that name describes some attributes of a resource.

This main problem of converting names to numbers in networking is as old as computer networking itself.

When a name server is present in a network the machines in the network only needed to know, the numerical address of the name server and the name of the destination machine or a website. With these information in hand the machines in the network can ask the name server in the network for the numerical address (IP address) of the destination.

a centralized server for the name server was much better than /etc/hosts file solution. because now with a central server dedicated for name to address mapping, the machines in the network only needed to know the numerical address of the name server, and the name server will return the numerical address of a name, whenever asked by the clients.

the major advantage of having a central name server was that the numerical address or the IP address of the server, can be changed without the clients being aware of the change. In such situations the name server just needed to be updated or modified with the new IP address.

But there were some drawbacks of a central name server also, like what if the central name server is not available? Hence came the idea of multiple name servers, in the network, one acted like a master or primary, and the other a slave or secondary. If master is not available then the secondary name server of the slave name server, is queried for the answer.

There were some main drawbacks of even this kind of an implementation (primary and secondary name servers). they are as follows.

1.As the names in the network goes on increasing, it becomes too much difficult for a name server to retrieve an information from millions of entries. So we needed a method to organize the names.

2.Imagine a single name server getting lot and lots of queries per second, in such cases the load on the name server increases. So we need to find a method to spread the load.

3.We needed a mechanism to separate the administration of the entries in the name server, as many different administrators used to add entires.

**The Domain Name System of the Internet.**

the complete DNS functionality is explained in the following RFC's

RFC 1034

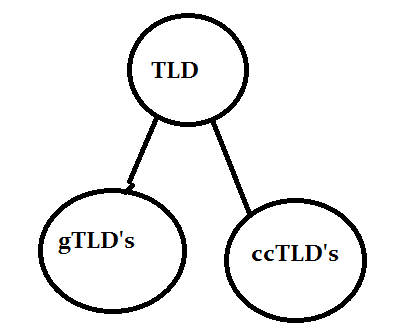
RFC 1035

The domain name system of the internet works in a inverted tree structure.At the top of the tree is the root name server(don't worry, i will explain whats a root server).The root server is followed by TLD's or Top Level Domains,and then TLD's are followed by SLD's or Second Level Domains. All of these are seperated by dots.

Understanding the above explained thing which is underlined is very much important in understanding the concept of DNS.

The root server is represented by a .(a dot).

TLD's are split into two types as follows.

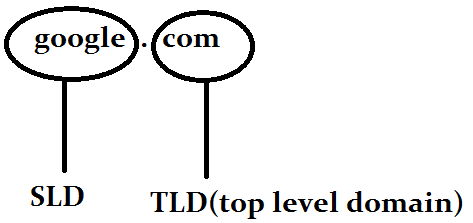


Generic Top Level Domains(gTLD's) are TLD's like .com,.net,.org,.edu etc.

Country Code Top Level Domains are domains such as .in,.us,.uk etc.

Now when we call www.slashroot.in a domain name, this domain name is a combination of gTLD,SLD(Secondry Level Domain) and the host name.We will come back to this in some time.

When we normally call a domain like google.com its the combination of TLD,SLD.



Each and every node in this Domain Name system is assigned to an authority or organization for its administration. And that organization resposible for a particular node is authoritative for that node.The term authoritative will be used many times in DNS system.

Now the authority of the .(root name server) which is at the top of the heirarchy lies with an organization named ICANN(Internet Corporation for Assigned Names And Numbers.).

gTLD's like (.com,.net) and others are also administered by ICANN and are also delegated to ICANN accredited registrars. ccTLD's are accredited to different countries for administration by ICANN.

Delegation in DNS is an important concept...I will keep another dedicated post only for delegation.

It is very much important to understand the fact that, the left most part (www) in any address, like for example www.slashroot.in, is the hostname. WWW is used by websites only by convention, there is no rule to use www for a website. A web site can also be named xyz.example.com.

**what happens when I type www.example.com in the address bar of the browser?**

the root name server(.) is the most important resource in the name server heirarchy. when any name server is asked for an information which it does not have, the first thing that name server does is asking one of the (.)root name server.

there are 13 root name servers as follows.

a.root-servers.net.  
b.root-servers.net.  
c.root-servers.net.  
d.root-servers.net.  
e.root-servers.net.  
f.root-servers.net.  
g.root-servers.net.  
h.root-servers.net.  
i.root-servers.net.  
j.root-servers.net.  
k.root-servers.net.  
l.root-servers.net.  
m.root-servers.net.

Now the ip address of all the root servers mentioned above are known to all the DNS software packages, by default. Which means all the DNS servers can reach these root servers without any other DNS server.

**Step1:** the client types www.example.com in his browser

**Step2:** the operating system looks at /etc/host file,first for the ip address of www.example.com(this can be changed from /etc/nsswitch), then looks /etc/resolv.conf for the DNS server IP for that machine

**Step3:** the dns server will search its database for the name www.example.com, if it finds it will give that back, if not it will query the root server(.) for the information.

**Step4:** root server will return a referral to the .com TLD name server(these TLD name servers knows the address of name servers of all SLD's).In our case we searched for www.example.com so root server will give us referral to .com TLD servers.

If it was www.example.net then root server will give, .net TLD servers refferal.

**Step5:** Now One of the TLD servers of .com will give us the referral to the DNS server resposible for example.com domain.

**Step6:** the dns server for example.com domain will now give the client the ip address of www host(www is the host name.)

Now lets practically have a look at how this process works.

[root@myvm1 ~]# dig +trace www.google.com  
  
; <<>> DiG 9.3.4-P1 <<>> +trace www.google.com  
;; global options:  printcmd  
.                       5       IN      NS      a.root-servers.net.  
.                       5       IN      NS      b.root-servers.net.  
.                       5       IN      NS      c.root-servers.net.  
.                       5       IN      NS      d.root-servers.net.  
.                       5       IN      NS      e.root-servers.net.  
.                       5       IN      NS      f.root-servers.net.  
.                       5       IN      NS      g.root-servers.net.  
.                       5       IN      NS      h.root-servers.net.  
.                       5       IN      NS      i.root-servers.net.  
.                       5       IN      NS      j.root-servers.net.  
.                       5       IN      NS      k.root-servers.net.  
.                       5       IN      NS      l.root-servers.net.  
.                       5       IN      NS      m.root-servers.net.  
;; Received 228 bytes from 192.168.159.2#53(192.168.159.2) in 49 ms  
  
com.                    172800  IN      NS      a.gtld-servers.net.  
com.                    172800  IN      NS      b.gtld-servers.net.  
com.                    172800  IN      NS      c.gtld-servers.net.  
com.                    172800  IN      NS      d.gtld-servers.net.  
com.                    172800  IN      NS      e.gtld-servers.net.  
com.                    172800  IN      NS      f.gtld-servers.net.  
com.                    172800  IN      NS      g.gtld-servers.net.  
com.                    172800  IN      NS      h.gtld-servers.net.  
com.                    172800  IN      NS      i.gtld-servers.net.  
com.                    172800  IN      NS      j.gtld-servers.net.  
com.                    172800  IN      NS      k.gtld-servers.net.  
com.                    172800  IN      NS      l.gtld-servers.net.  
com.                    172800  IN      NS      m.gtld-servers.net.  
;; Received 504 bytes from 198.41.0.4#53(a.root-servers.net) in 153 ms  
  
google.com.             172800  IN      NS      ns2.google.com.  
google.com.             172800  IN      NS      ns1.google.com.  
google.com.             172800  IN      NS      ns3.google.com.  
google.com.             172800  IN      NS      ns4.google.com.  
;; Received 168 bytes from 192.33.14.30#53(b.gtld-servers.net) in 12 ms  
  
www.google.com.         300     IN      A       74.125.236.48  
www.google.com.         300     IN      A       74.125.236.50  
www.google.com.         300     IN      A       74.125.236.51  
www.google.com.         300     IN      A       74.125.236.49  
www.google.com.         300     IN      A       74.125.236.52  
;; Received 112 bytes from 216.239.34.10#53(ns2.google.com) in 108 ms

Now you can clearly see from the dig with trace output that, the request first went to root servers. a.root-servers.net replied me with the addresses of all .com gtld servers, and b.gtld-servers.net gave me the name servers for google.com and finally ns2.google.com replied me with the ip address of www.google.com