DNS

Domain Name System (DNS)

The application layer consists of various applications. Out of those one is DNS, which stands for Domain Name System. The very first question arise: 'what is the need of this application?'. To begin with let's start with a real world example. There are many identifiers to be a unique person in the world, such as SSN, name, and Passport number along with the county who issued it, etc. In the similar fashion, every computer or host and router in the world has a unique identifying 32-bit 'IP' address. Say if we need some information that is on other part of the world. We need to know the IP address of that machine. Remembering IP addresses is difficult, as it contains all numbers. To remember IP addresses of more than one host becomes cumbersome. Therefore a name has been assigned to almost every IP address which makes it easier for humans to remember. DNS provides mapping of IP address and Domain name. (More details of IP address will be covered in further sections.)

How often is this mapping needed?

Answer. Every time when a host needs to convert a domain name to the IP address, a DNS query is called.

DNS Service

1. Host name to IP address translation

The primary purpose of DNS is to provide translation of host name to IP address and vice versa. The backward facility (translating IP address to domain name) is known as Reverse DNS.

2. Host aliasing

Host aliasing is referred to another name given to the same machine on the network. It is used because a hostname may have a complicated name instead of that a simple term may be used. E.g. relay.eastcost.rediff.com may have an alias name rediff.com

3. Mail server aliasing

It is highly desirable that an email address should contain simple letters, or should be something that can be easy to remember. E.g. richard@gmail.com can be remembered easily but if the original mail server address, say la4.mail1.google.com, were to be used it would be difficult to remember

4. Load distribution

A set of IP address is provided to one canonical name which prevents the load to be present only on one server. “When the request comes to the DNS server to resolve the domain name, it gives out one of the several canonical names in a rotated order. This redirects the request to one of the several servers in a server group. Once the BIND feature of DNS resolves the domain to one of the servers, subsequent requests from the same client are sent to the same server.”[1]

Why not centralize DNS?

Problems that arise when we try to centralize DNS.

1. Single point of failure
2. Increase in traffic volume
3. Distant centralized database
4. Maintenance

As centralized DNS does not scale because of the reasons mentioned above, a need arose to implement DNS in a distributed manner . The DNS is a distributed system, implemented in a hierarchy of many name servers. The decentralized administration is achieved through delegation.

Structure of DNS

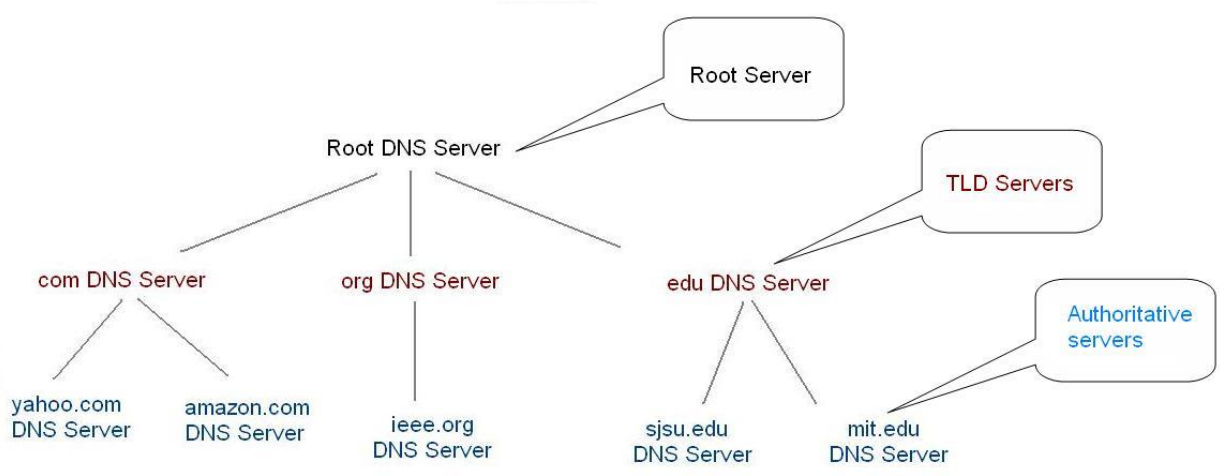

Fig. Structure of Domain Name System (DNS)

The structure of DNS is similar to the structure of Unix file system. It is a tree-like structure in which the root is known as the Root DNS sever. Each node in the tree is associated with a resource record

which holds the information associated with it, and can have any number of branches. There can be a maximum of 127 levels in a tree; however, you will never find any domain name that long. Each node in a tree represents its part in a domain name which can contain a maximum of 63 characters long.

The full domain name of any node in the tree is the sequence of each node in that path from the node to the root. Domain name is read from the node to the root with a dot placed separating the names in the path. No two nodes can have the same name if and only if they have the same parent. This guarantees that each domain name in the DNS tree corresponds to unique domain name in the entire DNS structure. E.g. you can not have multiple directories named “Program Files” in one single directory, but if you wish you can have a directory name “Program Files” in your root directory of your C drive, and in the “C:\Windows” directory (or in any number of distinct directory locations).

“A domain is simply a subtree of the domain name space. The domain name of a domain is the same as the domain name of the node at the very top of the domain.”[2] Consider the figure below

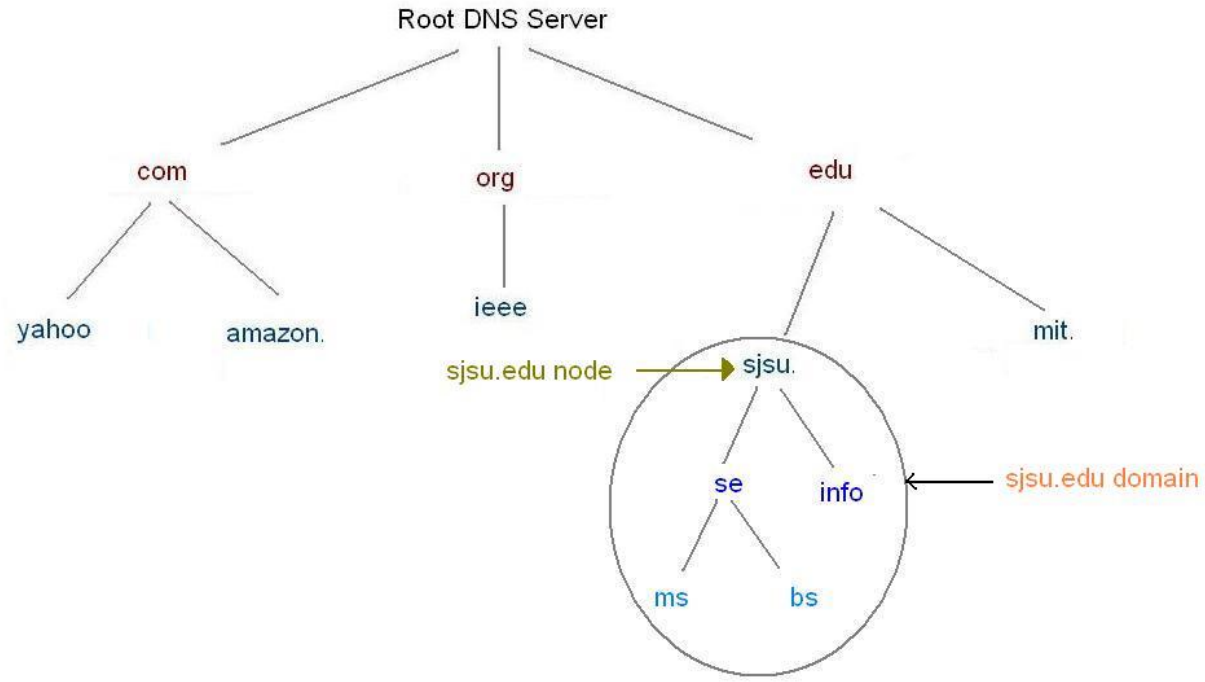

Fig. sjsu.edu domain

As you can see in the figure above that the “sjsu” domain is a part of the edu domain. In the similar fashion there can be many domains in the “sjsu” domain. “Any domain name in the subtree is consider a part of the domain. Because a domain name can be in many subtrees, it can be in many domains.”

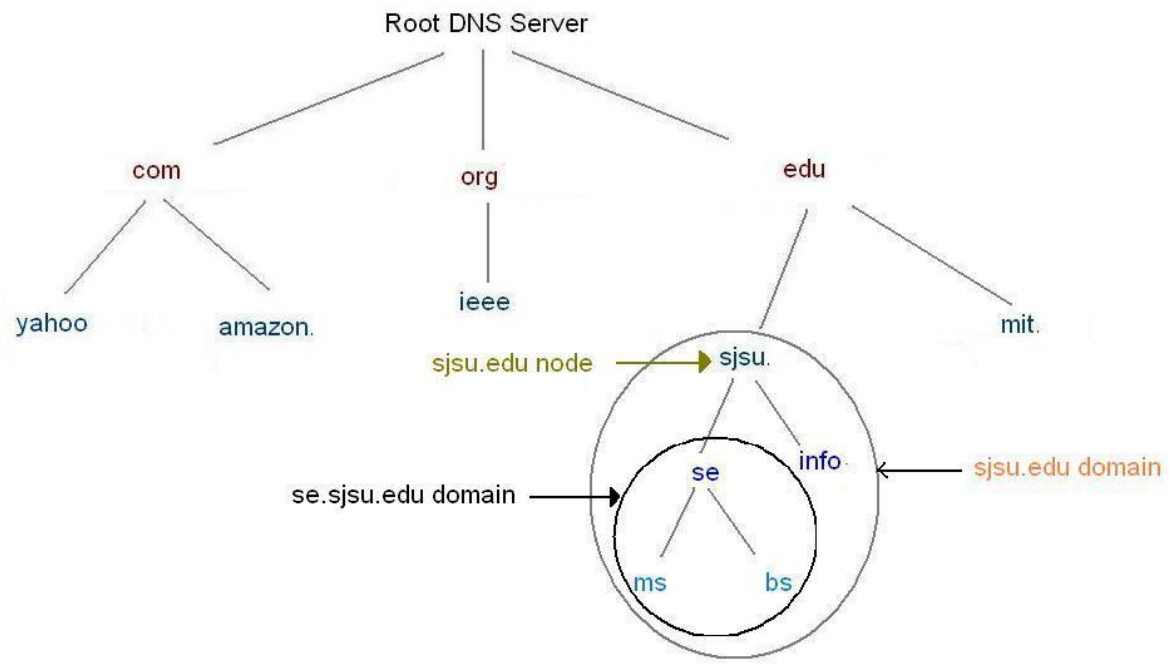

Fig. Subdomain (domain under domain)

After going through all these details, the very first question arises: if there are domains inside a domain, then where are all the hosts? If you would remember we had discussed earlier that the domain names are indexed into DNS. A domain may have a single host or a collection of host. Hosts are connected logically and may be dispersed to geographical locations. You may have 100 hosts connected to same domain that are located in different countries, or maybe all those host would be in different network too. Usually, the leaves in the tree represents hosts and may point to single network address, hardware information, and mail routing information. The interior nodes of the tree that represent a domain can also be used to represent a host on the network. E.g. In the above fig.sjsu.edu can be both 'San Jose State University domain' and also the name of the host (more specifically a web server) that run that domain.

A domain may contain many subdomains inside it. To identify if domain is a subdomain of another domain, you need to compare the domain name with its parent domain name. E.g. se.sjsu.edu is a subdomain of the sjsu.edu domain. The other way to determine the subdomains is through looking at the levels of the tree.

Clients wants IP of sjsu.edu

1. Client queries a root server to find edu DNS server
2. Client queries edu DNS server to get sjsu.edu DNS server
3. Client queries sjsu.edu DNS server to get IP address for www.sjsu.edu

Root DNS Servers

The root DNS servers (root name servers) keep track of all the authoritative name servers of each of the Top Level Domain (TLD) name servers. The client queries the root name servers to resolve a request for the given domain name. In response the root name server provides the address of the TLD name server for the given query in which the domain name ends with. E.g. If a client requests for google.com, the root name server will address the client to the com DNS server so as to solve his query. The top level name server holds the list of authoritative name server in their respective domain. E.g. The com domain holds the address of yahoo.com, google.com etc. The querier (the term querier is given because there are two approach for resolving a request – iterative and recursive approach) then queries the top level name server to resolve the query which returns the address of the authoritative name server. Each name server query gives the querier the required information or takes him one step ahead towards its destination.

Root name servers play a key role in resolving any DNS query. Usually DNS provides caching which reduces the load at the root name server. In event when the local name server is unable to find the given domain in its cache the query arrives at the root name server.

Fig. 13 root name servers worldwide

There are 13 root name servers worldwide. If a situation arises that the entire 13 root name servers are unreachable, the Internet would fail. Usually a host sends a query to its nearest root name server. If any one of these servers fails the requests are diverted to another nearest server. E.g. If

you are in India the nearest root name server is at Tokyo. If the root name server at Tokyo is down, all the DNS queries or traffic is diverted to server at Europe, which is the next nearest server to India.

Top Level Domain (TLD)

As discussed earlier, each domain name is made up of a series of character strings (called "labels") separated by dots. The right-most label in a domain name is referred to "Top-Level Domain" (TLD). Every TLD includes many second-level domains E.g. sjsu.edu. Every second level TLD may include number of third level domains. E.g. se.sjsu.edu. This process can go on.

Refer to the fig. structure of domain to find out where exactly are TLD located in the DNS hierarchy. The TLD divides the internet domain name space into several domains. Most commonly used domains are:

1. com – Usually used by commercial organization. E.g. Yahoo (yahoo.com)

2. edu – Usually used by educational institutes. E.g. San Jose State University (sjsu.edu)

3. org – Used by non profit organizations. E.g. IEEE (ieee.org)

4. mil – used by military organizations. E.g. US army (army.mil)

5. net – In earlier days it was used to represent the network infrastructure. Nowadays it is open public for any commercial organization.

6. gov – used to represents government organization. E.g. NASA (nasa.gov)

Apart from those mentioned above there are many more domains available. Every country has its own domain name space, which is represented by the name of the country. E.g. United States has a domain name ‘us’, India has a domain name ‘in’.

Zones

Before jumping to the authoritative name servers, we will have a quick overview of zones and delegates. A zone is similar to a domain except a subtle difference. The Top Level Domain and the domains under a Top Level Domain are divided into smaller units with the help of delegation. The need arises to divide these domains into small units, so that it can be managed easily. These small units are called zones. E.g. There would be many zones which are present under the root name servers. They might be com zone, edu zone, org zone etc. similar to a sub-domain concept; here too we have zones present inside zones. Therefore we would have many zones insides the com zone, edu zone, org zone etc. E.g. The edu zone may hold a zone name sjsu.edu, mit.edu etc. In the similar fashion, the com zone may hold yahoo.com zone, cisco.com zone, etc. By dividing the DNS structure into zones, it is each zones responsibility to manage their own domain. E.g. If the edu domain was not divided into different zones, it would be the responsibility of edu to manage sjsu.edu, mit.edu etc. which would become cumbersome for the people who manage the edu domain, therefore it is a natural to break up into different zones depending upon their responsibility.

Authoritative DNS Servers

Refer to fig.1 structure of DNS to check out where exactly are authoritative name servers are located in the DNS hierarchy. All the organizations such as yahoo, msn, sjsu, ieee, etc. contain their own authoritative servers. The goal of authoritative name servers is to provide the mapping of hostname

to IP address. All the details for that organization such as web pages, mail routing information etc. are also mentioned in the authoritative name server. Each authoritative name server has to be maintained by their independent organization or by the service provider for that organization.

Each domain or subdomain has one or more authoritative DNS servers that publish information about that domain and the name servers of any domains "beneath" it. Each zone is served by at least one authoritative name server, which contains the complete data for the zone. To make the DNS tolerant of server and network failures, most zones have two or more authoritative servers. Responses from authoritative servers have the "authoritative answer" (AA) bit set in the response packets. We will discuss about the authoritative answer in detail when we will come to the Resource Record section. This makes them easy to identify when debugging DNS configurations using tools like ‘dig’.

Local DNS Server

Apart from the Root DNS Server, Top Level DNS server, and the Authoritative DNS server we have a Local DNS Sever. The local name server does not belong strictly to the hierarchy, and that’s the reason you won’t find the local name server in the fig. Structure of Domain Name System. However it still plays an important role indirectly while resolving a DNS query.

Every ISP has a Local DNS server which is some times referred to as default name server. When the host is connected to the ISP, the ISP issues a single IP address through DHCP mechanism (More details of DHCP is covered in section 5.1). You can check your IP address of your computer with the ‘ipconfig’ command in Windows or 'ifconfig' command in Linux.

A host makes a DNS request and when it arrives to the Local DNS server it first checks in cache (more details on DNS caching is covered in further) to see if it can solves the DNS request. If it finds the requested information in the cache, it returns the response to the host. The response it return is called Not Authoritative answer. If the requested information is not found in the cache, then there are two approaches to resolve the given query.

1. Iterative

2. Recursive

In short we can say that the Local DNS server is a proxy to forward the query into the DNS hierarchy. The entire process of resolving the query through the hierarchy remains transparent to the user.

Iterated Queries

Lets have a look on how an iterative approach works. We assume that a host name se.sjsu.edu is requesting the IP address of mail.yahoo.com. We also assume that the authoritative DNS server for mail.yahoo.com is dns.yahoo.com. The way the DNS resolves the request is shown below.

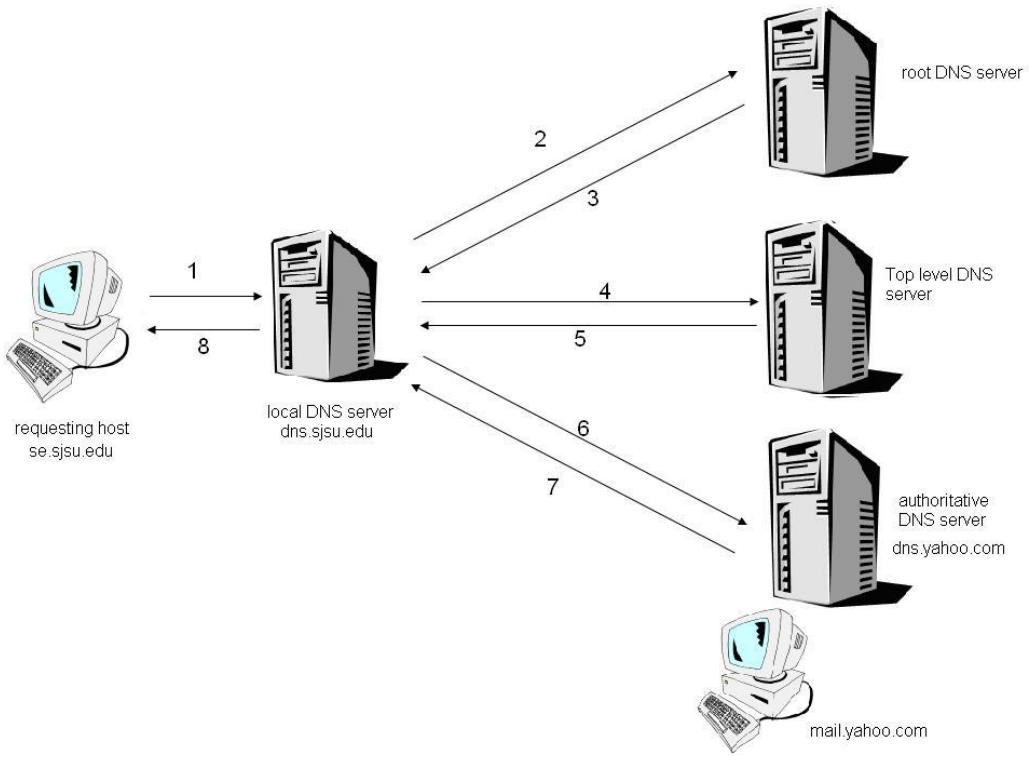

Fig. Host 'se.sjsu.edu' is requesting for IP address of mail.yahoo.com – Iterative Query

The host se.sjsu.edu sends a DNS query to the local DNS server to translate the hostname 'mail.yahoo.com', provided in the query, to the IP address. In response, the local DNS server i.e. dns.sjsu.edu forwards the query to the root DNS server. The root DNS server finds the suffix as 'com' and returns a list of IP address of the top level DNS server responsible for 'com'. The local DNS server then sends the same query to one of the top level DNS servers which were provided by the root DNS server. The top level DNS server finds a suffix yahoo.com and returns the local DNS server with an IP address of the authoritative DNS server for Yahoo i.e. yahoo.com. Finally, the local DNS server sends the same query again to the authoritative DNS server dns.yahoo.com, which in turn responds with the IP address of mail.yahoo.com.

The above process can be summarized as

```
        User's computer: "What is the IP Address of mail.yahoo.com?"
        Local name server: "I don't know that. But I'll check with a name
server that does."
        Local name server: "What is the IP Address of mail.yahoo.com?"
        Root name server: "Here are the addresses for the authoritative
name servers for .com."
        Local name server: "What is the IP Address of mail.yahoo.com?"
```

```
          Authoritative .com name server: "Here are the addresses of the
authoritative name servers for yahoo.com."
          Local name server: "What is the IP address of mail.yahoo.com?"
          Authoritative mail.yahoo.com name server: "Here is the IP address
for mail.yahoo.com, its 205.139.94.60."
```

Although it is mentioned that the above method is an iterative approach but actually it makes use of iterative as well as recursive. The query given by se.sjsu.edu to dns.sjsu.edu is recursive query, but the one which was from the local DNS server to the root DNS server, top level DNS server and the authoritative DNS server are iterative since all the results are return back to the local DNS server (dns.sjsu.edu).

Recursive Queries

Just now we had a look on how the iterative query works for solving any DNS query. Let's now have a look on how recursive approach.

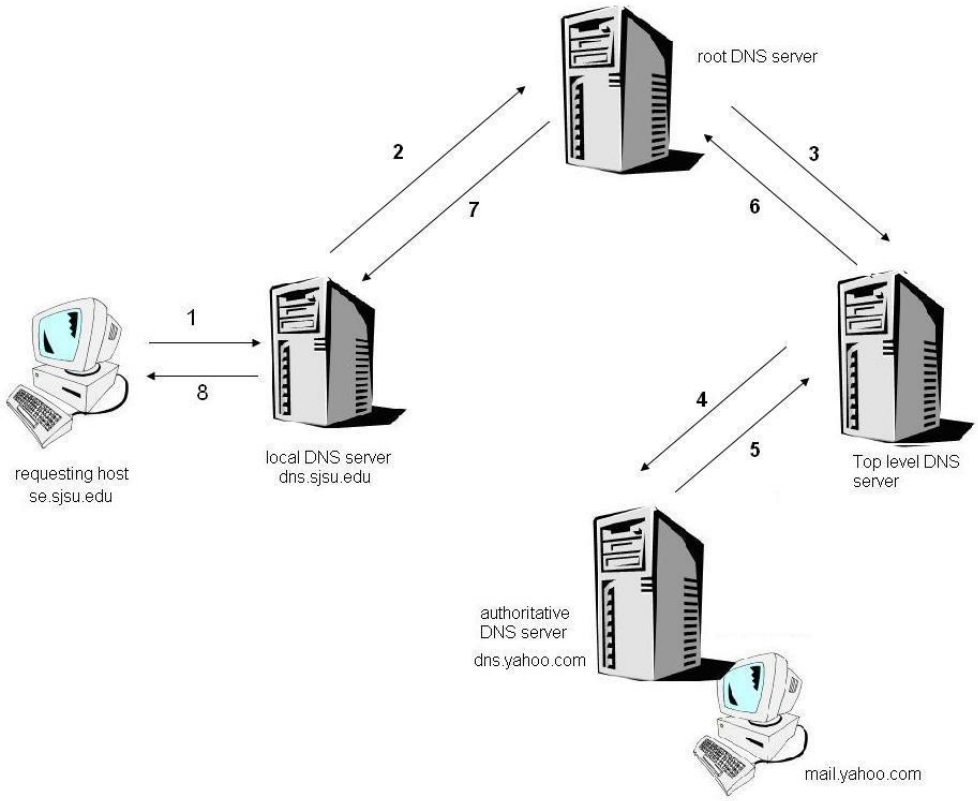

Fig. Host 'se.sjsu.edu' is requesting for IP address of mail.yahoo.com – Recursive Query

The complete flow of the recursive query is given below

1. Requesting host ‘se.sjsu.edu’ request its local DNS server ‘dns.sjsu.edu’ to solve a DNS query ‘mail.yahoo.com’ and to give its IP address

2. The Local DNS query asks the root DNS server for the IP address of ‘mail.yahoo.com’

3. The root DNS server finds the ‘com’ suffix in the query and request one of the top level DNS server responsible for com

4. The com, top level DNS server keeps track of the entire authoritative DNS server; it asks the authoritative DNS server of Yahoo (dns.yahoo.com) for the IP address of mail.yahoo.com

5. The authoritative DNS server of Yahoo returns the IP address to the com Top Level DNS server who queries the authoritative DNS

6. The Top level DNS server returns this IP address to root DNS server

7. The root DNS server in turn returns the IP address to the local DNS query.

8. The host receives the IP address of its desired query.

In theory, a recursive query is resolved in the manner explained above. But in practice, recursive query is not used as it is not much efficient, and soon it will overrun its stack.

DNS Caching

In the above 2 approach (iterative and recursive query) of solving the query we saw that a total of 10 messages had been sent. This reduces the efficiency of DNS. Therefore a caching mechanism is designed so as to reduce the flooding of DNS packets in the cyberspace. The DNS extensively make use of cache so as to improve the performance which is otherwise decrease by going through the root DNS server, TLD server and the authoritative DNS server. Caching reduces traffic of DNS packets over the internet.

When a DNS query is resolved and the IP address of that domain is obtained the DNS server simply cache the required information from the reply. In the above example (refer to fig 5) each time when the local DNS server dns.sjsu.edu receives a DNS reply from any DNS server, it can cache the any of the required information in its cache for future use. Say if the hostname corresponding to the IP address are cached in the local DNS server, the local DNS server can provide the IP address for that hostname incase if a query for that hostname arrives in future, although it is not authoritative for that hostname. A local DNS server can store or cache an IP address of the TLD server so as to skip the time asking the root DNS server asking for the IP address of the TLD server.

As mapping of hostname to IP address may change, therefore the DNS server discards its cache after a certain amount of time.

An example of what happens when a query is made to the local DNS server and the mapping is found in its cache.

```
User's computer: "What is the IP address of mail.yahoo.com"
Local DNS server: "I know that. The IP address is 209.73.168.74.
```

Note: The iterative or recursive query was already executed once before we get such a quick response.

DNS dynamic update / notify

The update/notify mechanism is design under RFC 2136. Many companies and usually all ISP use DHCP (DHCP is covered in section 5.1 in more detail) to assign IP address to the hosts which are connected to them (server). For this DNS is needed to support dynamic addition and deletion of resource records (RR - Resource Records are covered in further sections). This mechanism is called DNS dynamic update.

The dynamic update facility allows authorized updaters to add or delete a RR from a zone where a name server is authoritative. With the help of NS record the authorized update message is sent to the primary node of that zone. If a name server receives any update message and if that name server is not a primary node of that zone then that message is forwarded upstream to its master server. If the server who receives it is also slave then it is again forwarded upstream. This process is called “update forwarding” and it continues till the update message is received to the primary node of that zone.

The primary master name server holds a writable copy of zone data. The slave nodes are notified when an update is performed either directly or indirectly.

DNS Resource Records

Every domain, whether it is a Top Level Domain, or an Authoritative server, or simply a single host have a set of resource records associated with it in the DNS distributed database. These RR provide the mapping of hostname to IP address. The RR is stored in binary format for internal use, but when a RR is transmitted in cyberspace it is text format.

When a query is made to the DNS server, the querier (host/server who sends that query) receives a response which is nothing but the resource record associated with it.

The Resource Records is a 5 tuple that contains the following

```
( Name,  Time to live, Class, Type, Value)
```

1. Name: It is the domain name to which this resource record belongs to. More than one resource records may exists for the same domain.

2. Time to live: This is a 32 bit integer. The TTL is measured in seconds. The value zero indicates the data should not be cached.

3. Class: The field usually contains the value ‘IN’ it represents if this record is to be used by internet.

4. Type: The type field defines the type of resource record – Address, Name Service, Mail Exchange, Canonical Name.

5. Value: This field can be a number, ASCII strings or any domain. The semantics of Name and Value depends on the type field.

Various type fields along with the details are mentioned below.

1. Type = ‘A’

```
    ‘A’ stands for address where
    Name = Hostname (e.g. yahoo.com)
    Value = IP address (e.g. 216.109.112.135)
```

Thus it can provide mapping of hostname to IP address.

2. Type = ‘NS’

```
    ‘NS’ stands for Name Service
    Name = Domain name (e.g. yahoo.com)
    Value = Host name of Authoritative DNS server (e.g. dns.yahoo.com)
```

3. Type = ‘CNAME’

```
    ‘CNAME’ refers to canonical name. It is used to define alias hostname
    Name = Alias name (www.ibm.com)
    Value = Real name of that host (e.g. servereast.backup2.ibm.com)
```

4. Type = ‘MX’

```
     'MX’ stands for Mail Exchange.
     Name = Domain name (eg.yahoo.com)
     Value = Name of mail server associated with that name. (e.g.
mx.mail.yahoo.com)
```

DNS messages

As we have finished with most of the parts of DNS, now lets look at how a DNS message looks like. The figure below provides a DNS message format. The query and response, both, are within the same message format.

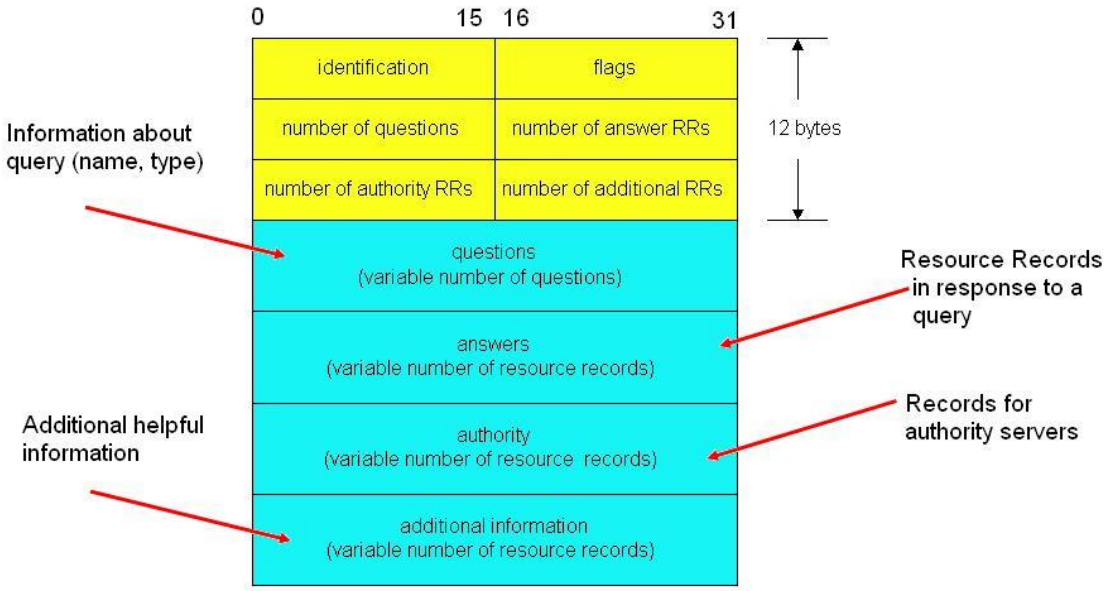

Fig. DNS message

1. Identification – This is a 16 bit number through which a query is identified. This number is set by client and when a response is send back to the client, the same identification number is used.

2. The Flag consists of 16-bit parameter:

a) The first (0th)bit indicates query(0) or response(1)

b) Next three bits (1-4) indicates 'Standard Query (0)', 'Inverse Query (1)' and 'Server Status Request (2)'.

c) The 5th bit field indicates Authoritative answer. The name server is authoritative for the domain in the question section.

d) The 6th bit field is set if message was truncated. With UDP this means that the total size of the reply exceeded 512 bytes and only the first 512 bytes of reply were returned.

e) The 7th bit field indicates Recursion Desired .This bit can be set in a query and is returned in the response.

f) The 8th bit field indicates Recursion Available or not.

g) The next 3 bits (9-11) has to be 0.

h) The Next 4 bits (12-15) give a return code where 0 signifies No Error and 3 signifies Name Error.

3. The fields labeled Number of... give each a count of entries in the corresponding sections in the message.

4. The Question section is filled by the client and contains information about the query that is being made. Each question has a name and type associated with it.

5. The Answer, Authority, and Additional Information sections consist of a set of resource records that describe the domain names and mappings.

Insert Resource Record in DNS Database

Until now we have seen how DNS is used to find IP address of any host in the cyberspace and the use of it. Here the most important question arise is: How these resource records are inserted into DNS database. For this we consider a real time example, Lets say a company name INetwork is established and want to publish a website on internet.

The company INetwork approaches the registrar to register its domain name 'inetwork.com'. The registrar is responsible to maintain uniqueness of domain.

The company INetwork provides the registrar with the names and IP address of their primary and secondary authority DNS server. The registrar enters the information in form of resource record and stores it into DNS database

```
(   inetwork.com,          NS    ,     dns.inetwork.com   )
(   dns.inetwork.com,      A     ,     203.166.178.34     )
```

The company needs to make sure that the Type 'A' resource record for the their web server inetwork.com and Type 'MX' resource record for the company's mail server are entered into our authority DNS server to make sure that others can access your website and employees can use the system to send mails.

References

1. Jump up↑ Load_balance_dns.html
2. Jump up↑ DNS and BIND – Paul Albitz & Cricket Liu, Published by O'Reilly publicactions. Ch.2 Pg13, 14.

- Microsoft Technet, Chapter 16 from Microsoft Windows 2000 TCP/IP Protocols and Services Technical Reference, published by Microsoft Press, By Thomas Lee and Joseph Davies
- Computer Networking – A top down approach featuring the Internet. J Kurose & K Ross
- Computer Networks – Andrew Tanenbaum
- http://www.zytrax.com/books/dns/ch8/
- Materials from Cisco Systems http://www.cisco.com/en/US/tech/tk648/tk362/technologies_tech_note09186a0080094727.shtml#t6
- http://www.verisign.com.au/dns/fyi.shtml