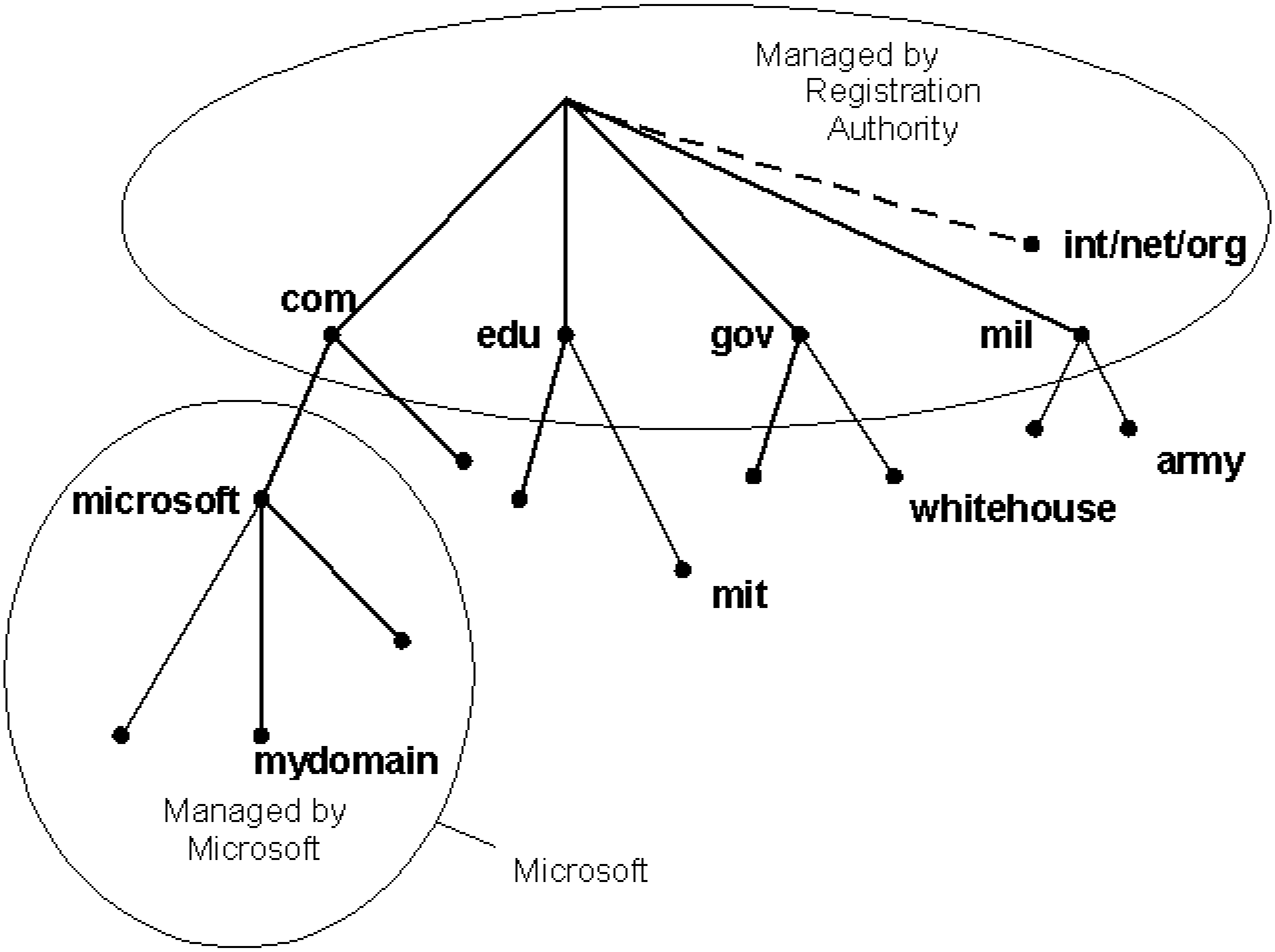
DNS (Domain Name Service)

The internet primarily uses IP addresses for locating nodes. However, its humanly not possible for us to keep track of the many important nodes as numbers. Alphabetical names as we see would be more convenient to remember than the numbers as we are more familiar with words. Hence, in the chaotic organization of numbers (IP addresses) we would be much relieved if we can use familiar sounding names for nodes on the network.

There is also another motivation for DNS. All the related information about a particular network (generally maintained by an organization, firm or university) should be available at one place. The organization should have complete control over what it includes in its network and how does it "organize" its network. Meanwhile, all this information should be available transparently to the outside world.

Conceptually, the internet is divide into several hundred top level domains where each domain covers many hosts. Each domain is partitioned in subdomains which may be further partitioned into subsubdomains and so on... So the domain space is partitioned in a tree like structure as shown below. It should be noted that this tree hierarchy has nothing in common with the IP address hierarchy or organization.

The internet uses a hierarchical tree structure of Domain Name Servers for IP address resolution of a host name.



The top level domains are either generic or names of countries. eg of generic top level domains are .edu .mil .gov .org .net .com .int etc. For countries we have one entry for each country as defined in ISO3166. eg. .in (India) .uk (United Kingdom).

The leaf nodes of this tree are target machines. Obviously we would have to ensure that the names in a row in a subdomain are unique. The max length of any name between two dots can be 63 characters. The absolute address should not be more than 255 characters. Domain names are case insensitive. Also in a name only letters, digits and hyphen are allowed. For eg. www.iitk.ac.in is a domain name corresponding to a machine named www under the subsubdomain iitk.ac.in.

Resource Records:

Every domain whether it is a single host or a top level domain can have a set of resource records associated with it. Whenever a resolver (this will be explained later) gives the domain name to DNS it gets the resource record associated with it. So DNS can be looked upon as a service which maps domain names to resource records. Each resource record has five fields and looks as below:

Domain Name	Class	Type	Time to Live	Value

- Domain name: the domain to which this record applies.
- Class: set to IN for internet information. For other information other codes may be specified.
- Type: tells what kind of record it is.
- Time to live: Upper Limit on the time to reach the destination
- Value: can be an IP address, a string or a number depending on the record type

Resource Record

A **Resource Record** (RR) has the following:

- **owner** which is the domain name where the RR is found.
- **type** which is an encoded 16 bit value that specifies the type of the resource in this resource record. It can be one of the following:
    - **A** a host address
    - **CNAME** identifies the canonical name of an alias
    - **HINFO** identifies the CPU and OS used by a host
    - **MX** identifies a mail exchange for the domain.
    - **NS** the authoritative name server for the domain
    - **PTR** a pointer to another part of the domain name space
    - **SOA** identifies the start of a zone of authority class which is an encoded 16 bit value which identifies a protocol family or instance of a protocol.
- **class** One of: **IN** the Internet system or **CH** the Chaos system
- **TTL** which is the time to live of the RR. This field is a 32 bit integer in units of seconds, an is primarily used by resolvers when they cache RRs. The TTL describes how long a RR can be cached before it should be discarded.
- **RDATA** Data in this field depends on the values of the type and class of the RR and a description for each is as follows:
    - for A: For the IN class, a 32 bit IP address For the CH class, a domain name followed by a 16 bit octal Chaos address.
    - for CNAME: a domain name.
    - for MX: a 16 bit preference value (lower is better) followed by a host name willing to act as a mail exchange for the owner domain.
    - for NS: a host name.
    - for PTR: a domain name.
    - for SOA: several fields.

Note: While short TTLs can be used to minimize caching, and a zero TTL prohibits caching, the realities of Internet performance suggest that these times should be on the order of days for the typical host. If a change can be anticipated, the TTL can be reduced prior to the change to minimize inconsistency during the change, and then increased back to its former value following the change. The data in the RDATA section of RRs is carried as a combination of binary strings and domain names. The domain names are frequently used as "pointers" to other data in the DNS.

Aliases and Cannonical Names

Some servers typically have multiple names for convenience. For example www.iitk.ac.in & yamuna.iitk.ernet.in identify the same server. In addition multiple mailboxes might be provided by some organizations. Most of these systems have a notion that one of the equivalent set of names is the canonical or primary name and all others are aliases.

When a name server fails to find a desired RR in the resource set associated with the domain name, it checks to see if the resource set consists of a CNAME record with a matching class. If so, the name server includes the CNAME record in the response and restarts the query at the domain name specified in the data field of the CNAME record.

Name Servers

Name servers are the repositories of information that make up the domain database. The database is divided up into sections called zones, which are distributed among the name servers. Name servers can answer queries in a simple manner; the response can always be generated using only local data, and either contains the answer to the question or a referral to other name servers "closer" to the desired information. The way that the name server answers the query depends upon whether it is operating in recursive mode or iterative mode:

- The simplest mode *for the server* is non-recursive, since it can answer queries using only local information: the response contains an error, the answer, or a referral to some other server "closer" to the answer. All name servers must implement non-recursive queries.
- The simplest mode *for the client* is recursive, since in this mode the name server acts in the role of a resolver and returns either an error or the answer, but never referrals. This service is optional in a name server, and the name server may also choose to restrict the clients which can use recursive mode.

Recursive Query vs Iterative Query

If the server is supposed to answer a recursive quesry then the response is either the reource record data or a error code. A server operating in this mode will never return the name of any forwarding name server but will contact the appropiate name server itself and try to get the information.

In iterative mode, on the other hand, if the server does not have the information requested locally then it return the address of some name server who might have the information about the query. It is then the responsibility of the contacting application to contact the next name server to resolve its query and do this iteratively until gets an answer or and error.

Relative Names

In place of giving full DNS names like cu2.cse.iitk.ac.in or bhaskar.cc.iitk.ac.in one can give just cu2 or bhaskar.This can be used by the server side as well as the client side.But for this one has to manually specify these extensions in the database of the servers holding the resource records.

BOOTP

The BOOTP uses UDP/IP. It is run when the machine boots. The protocol allows diskless machines to discover their IP address and the address of the server host. Additionally name of the file to be loaded from memory and executed is also supplied to the machine. This protocol is an improvement over RARP which has the follwing limitations:

1. Networks which do not have a broadcast method can't support RARP as it uses the broadcast method of the MAC layer underneath the IP layer.
2. RARP is heavily dependent on the MAC protocol.
3. RARP just supplies the IP address corresponding to a MAC address It doesn't support respond with any more data.
4. RARP uses the computer hardware's address to identify the machine and hence cannot be used in networks that dynamically assign hardware addresses.

Events in BOOTP

1. The Client broadcasts its MAC address (or other unique hardware identity number) asking for help in booting.
2. The BOOTP Server responds with the data that specifies how the Client should be configured (pre-configured for the specific client)

Note: BOOTP doesn't use the MAC layer broadcast but uses UDP/IP.

Configuration Information

The important informations provided are:

- IP address
- IP address of the default router for that particular subnet
- Subnet mask
- IP addresses of the primary and secondary nameservers

Additionaly it may also provide:

- Time offset from GMT
- The IP address of a time server
- The IP address of a boot server
- The name of a boot file (e.g. boot image for X terminals)
- The IP domain name for the client

But the problem with BOOTP is that it again can't be used for the dynamic IP's as in RARP servers.For getting dynamic IP's we use DHCP.

DHCP (Dynamic Host Configuration Protocol)

DHCP (Dynamic Host Configuration Protocol) is a protocol that lets network administrators manage centrally and automate the assignment of Internet Protocol (IP) addresses in an organization's network. If a machine uses Internet's set of protocol (TCP/IP), each machine that can connect to the Internet needs a unique IP address. When an organization sets up its computer users with a connection to the Internet, an IP address must be assigned to each machine. Without DHCP, the IP address must be entered manually at each computer and, if computers move to another location in another part of the network, a new IP address must be entered. DHCP lets a network administrator supervise and distribute IP addresses from a central point and automatically sends a new IP address when a computer is plugged into a different place in the network.

IP Address Allocation Mechanism

DHCP supports three mechanisms for IP address allocation.

- **Automatic allocation:** DHCP assigns a permanent IP address to a host.
- **Dynamic allocation:** DHCP assigns an IP address to a host for a limited period of time (or until the host explicitly relinquishes the address).
- **Manual allocation:** Host's IP address is assigned by the network administrator, and DHCP is used simply to convey the assigned address to the host. A particular network will use one or more of these mechanisms, depending on the policies of the network administrator.

Messages Used by DHCP

- **DHCP Discover** - Client broadcast to locate available servers. It is assumed atleast one of the servers will have resources to fulfill the request.(may include additional pointers to specific services required eg. particular subnet, minimum time limit etc).
- **DHCP Offer -** Server to client in response to DHCP Discover with offer of configration parameters.
- **DHCP Request -** Client broadcast to servers requesting offered parameters from one server and implicitly declining offers from all others.(also important in case of lease renewal if the alloted time is about to expire).
- **DHCP Decline -** Client to server indicating configration parameters invalid.
- **DHCP Release -** Client to server relinquishing network address and cancelling current lease.(in case of a graceful shut down DHCP server is sent a DHCP Release by the host machine).
- **DHCP Ack -** Server to client with configration parameters, including committed Network address.
- **DHCP Nack -** Server to client refusing request for configratin parameters (eg. requested network address already allocated).

Timers Used

Note that lease time is the time specified by the server for which the services have been provided to the client.

- **Lease Renewal Timer -** When this timer expires machine will ask the server for more time sending a DHCP Request.

- **Lease Rebinding Timer -** Whenever this timer expires, we have not been receiving any response from the server and so we can assume the server is down. Thus send a DHCP Request to all the servers using IP Broadcast facility. This is only point of difference between Lease renewal and rebinding.
- **Lease Expiry Timer -** Whenever this timer expires, the system will have to start crashing as the host does not have a valid IP address in the network.

Timer Configuration Policy

The timers have this usual setting which can be configured depending upon the usage pattern of the network. An example setting has been discussed below.

Lease Renewal = 50 % Lease time
Lease Rebinding = 87.5 % Lease time
Lease Expiry = 100 % Lease time