



Domain Name System

(RFC 1034,1035,2181)

Mark Sathish Pairedha
16CS01032



History

Way back in the **arpanet** days, there was a text file called **hosts.txt** that listed all the computer names and IP addresses (~100). Every night all hosts fetch it from <https://www.internic.net/> where it was maintained.

Since the #internet users increased rapidly, the size of file became large thereby hostname conflicts occurred & maintaining a single, centralized host table had become slow and unwieldy.

This led to the invention of Domain Name System.

The hosts.txt file is stored in **C:/Windows/system32/drivers/etc/hosts.txt** (Windows) and **/etc/resolv.conf** (Linux).



What is DNS ?

- The domain name system is mechanism that maps host names to an IP addresses. (ex : www.iitbbs.ac.in to 14.139.204.218).
- It is hierarchical and has distributed domain based naming scheme & a distributed database system.
- Domain names comprise of hierarchy so that names are unique,yet easy to remember.
- The DNS delegates the responsibility of assigning domain names and mapping those to IP address by designating authoritative name server for each domain.
- Network administrators may delegate authority over sub-domains of their allocated name space to other name servers. This mechanism provides distributed and fault tolerant service and was designed to avoid a single large central database (ex : www.iitbbs.ac.in domain admin has authority over sub domain www.erp.iitbbs.ac.in).



DNS protocol transport

DNS primarily uses the User Datagram Protocol (UDP) on port number 53 to serve requests.

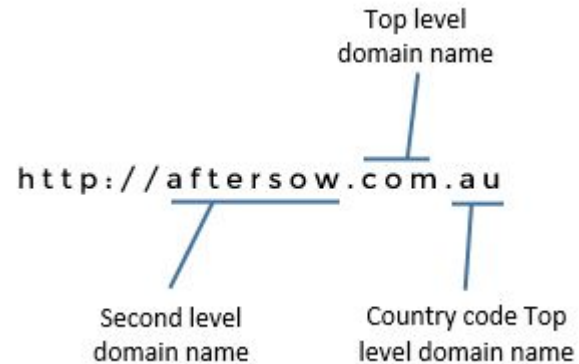
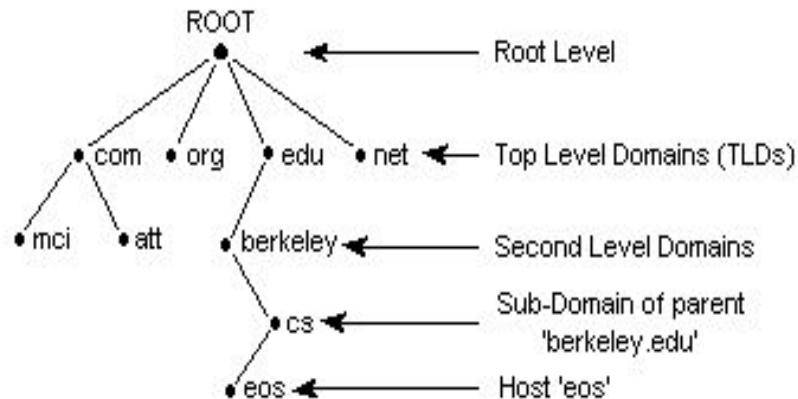
DNS queries consist of a single UDP request from the client followed by a single UDP reply from the server.

When the length of the answer exceeds 512 bytes and both client and server support EDNS, larger UDP packets are used.

Otherwise, the query is sent again using the Transmission Control Protocol (TCP). TCP is also used for tasks such as zone transfers. Some resolver implementations use TCP for all queries.

Domain Name Space & Hierarchy

DNS Hierarchy






DNS Name Servers

A **name server** implements a network service for providing responses to queries against a directory service.

Root name server : Its the first step in resolving host names into IP address. It directly answers requests for records in the root zone and answers other requests by returning a list of the Authoritative name servers for the appropriate Top Level Domains.

Top Level Domain name server : These servers are responsible for top-level domains such as com, org, net, edu, and gov, and all of the country top-level domains such as in,us,uk, fr, ca, and jp.It answers the requests for records in TLD zone and answers other request by returning the address of respective authoritative name servers.



Authoritative name server : It only gives answers to DNS queries from data that has been configured by an original source. An authoritative-only name server returns answers only to queries about domain names that have been specifically configured by the administrator. Name servers can also be configured to give authoritative answers to queries in some zones, while acting as a caching name server for all other zones.

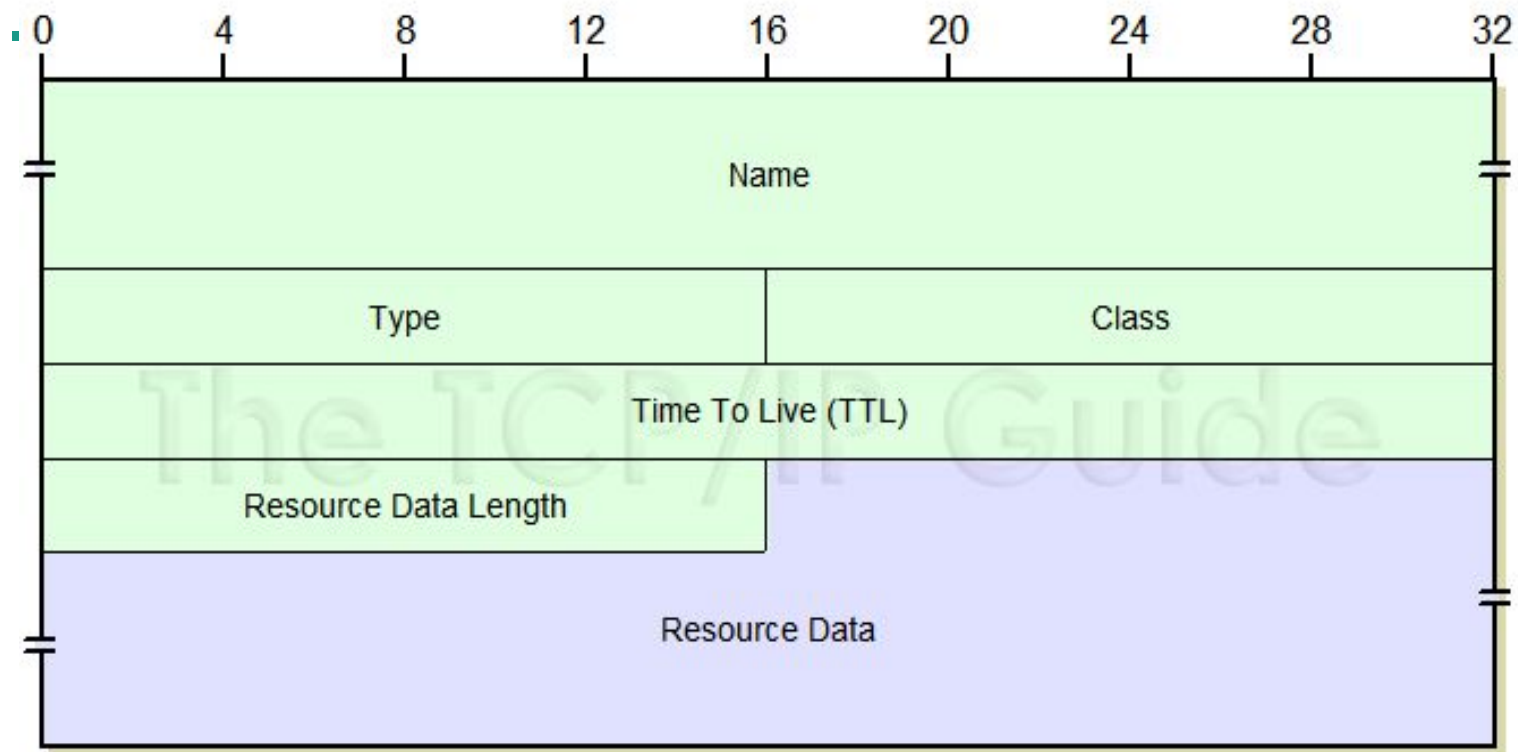
An authoritative name server can either be a *primary* server (master) or a *secondary* server (slave). A primary server for a zone is the server that stores the definitive versions of all records in that zone. It is identified by start-of-authority (SOA) resource record. A secondary server for a zone uses an automatic updating mechanism to maintain an identical copy of the primary server's database for a zone. Examples of such mechanisms include DNS zone transfers and file transfer protocols. DNS provides a mechanism whereby the primary for a zone can notify all the known secondaries for that zone when the contents of the zone have changed. The contents of a zone are either manually configured by an administrator, or managed using Dynamic DNS.



Domain Resource Records

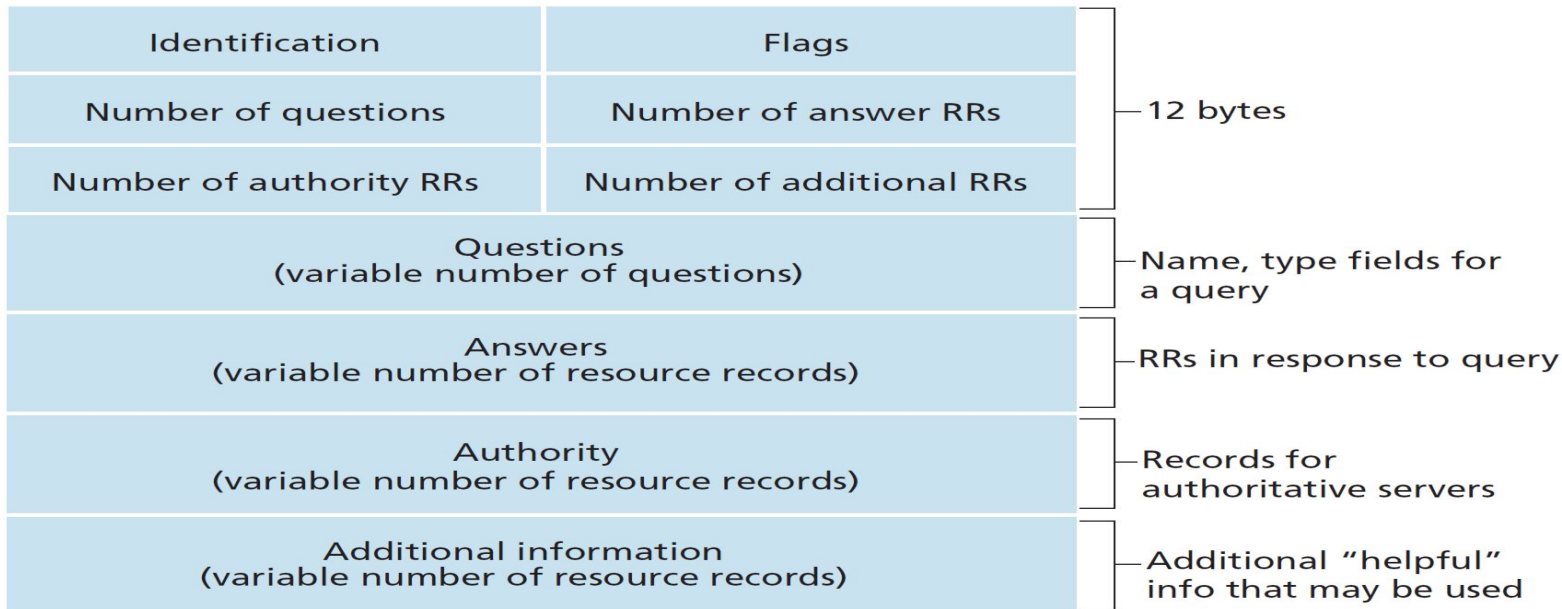
The Domain Name System specifies a database of information elements for network resources. The types of information elements are categorized and organized with a list of DNS record types, the resource records (RRs). Each record has a type (name and number), an expiration time (time to live), a class, and type-specific data. Resource records of the same type are described as a resource record set (RRset), having no special ordering. DNS resolvers return the entire set upon query, but servers may implement round-robin ordering to achieve load balancing.


When sent over an Internet Protocol network, all records use the common format specified in RFC 1035.






DNS Header Format





The first 12 bytes is the header section, which has a number of fields. The first field is a 16-bit number that identifies the query. This identifier is copied into the reply message to a query, allowing the client to match received replies with sent queries. There are a number of flags in the flag field, 1-bit query/reply flag indicates whether the message is a query (0) or a reply (1). A 1-bit authoritative flag is set in a reply message when a DNS server is an authoritative server for a queried name. A 1-bit recursion-desired flag is set when a client (host or DNS server) desires that the DNS server perform recursion when it doesn't have the record. A 1-bit recursion available field is set in a reply if the DNS server supports recursion. In the header, there are also four number-of fields. These fields indicate the number of occurrences of the four types of data sections that follow the header.

The question section contains information about the query that is being made. This section includes (1) a name field that contains the name that is being queried, and (2) a type field that indicates the type of question being asked about the name—for example, a host address associated with a name (Type A) or the mail server for a name (Type MX).



In a reply from a DNS server, the answer section contains the resource records for the name that was originally queried. Recall that in each resource record there is the Type (for example, A, NS, CNAME, and MX), the Value, and the TTL. A reply can return multiple RRs in the answer, since a hostname can have multiple IP addresses.

The authority section contains records of other authoritative servers.

The additional section contains other helpful records. For example, the answer field in a reply to an MX query contains a resource record providing the canonical hostname of a mail server. The additional section contains a Type A record providing the IP address for the canonical hostname of the mail server

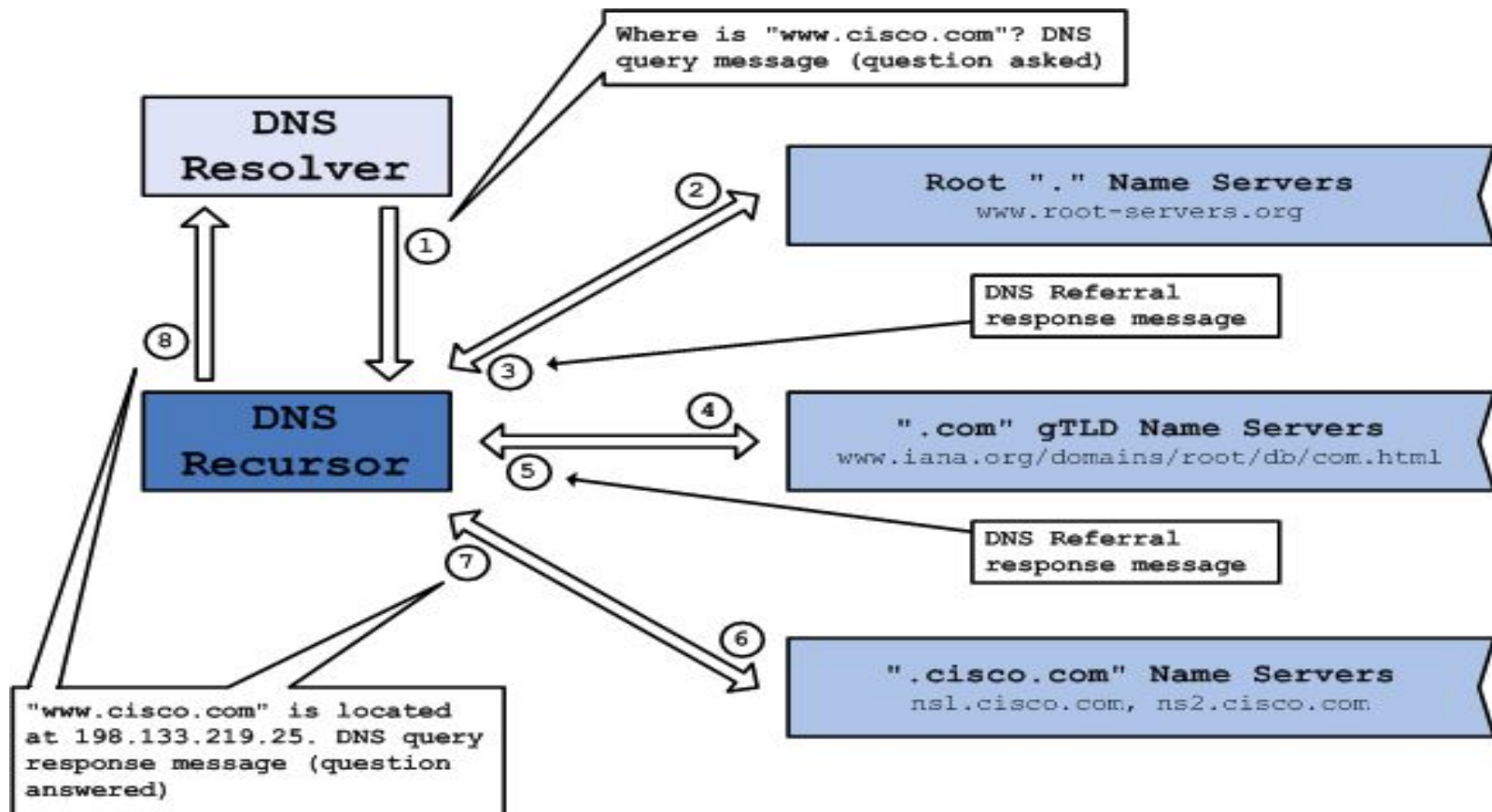


DNS Query

Recursive DNS Query : The DNS resolver sends a Query to a DNS Server for name resolution. The reply to the DNS Query can be an answer to the query or an error message. If the DNS Server doesn't know the answer to provide accurate answer to the DNS resolver, DNS Server may iterative query other DNS Servers on behalf of the DNS resolver.

Iterative DNS Query : When a DNS resolver asks the DNS root server for name resolution, the DNS Server provides the best answer it has. If the DNS Server doesn't know the answer, the server responds with an address reference to a TLD/authoritative server. This lower level DNS Server is delegated at the higher level DNS Server to be Authoritative for the DNS namespace which the DNS Query is related with. Once the DNS resolver get the referral from higher level DNS Server, it can then send a DNS Query to the lower level DNS server, got as referral.

Inverse DNS Query : Inverse DNS Queries (Reverse DNS Queries) are used when the user wants to resolve the IP Address to a Fully qualified domain name. Pointer (PTR) records are added to the in-addr.arpa domain. PTR Resource Records must be added in local DNS Server for Inverse Name Resolution to work properly.





References

- [Wikipedia : Domain-name-system](#)
- [Indian Institute of Technology,Kharagpur NPTEL](#)
- [Github DNS Code](#)
- [Computer Networks Tanenbaum PDF](#)
- [The TCP/IP Guide PDF](#)