**Assignment 2 - DNS**

The Domain Name System (DNS) is a hierarchical distributed naming system for computers, services, or any resource connected to the Internet or a private network. It associates various information with domain names assigned to each of the participating entities. Most prominently, it translates more readily memorized domain names to the numerical IP addresses needed for the purpose of locating and identifying computer services and device with the underlying network protocols. By providing a worldwide, distributed directory service, the Domain Name System is an essential component of the functionality of the Internet.

**Resolver operation**

When a computer on the Internet needs to resolve a domain name, it uses resolver software to perform the lookup. A resolver breaks the name up into its labels from right to left. The first component (TLD) is queried using a root server to obtain the responsible authoritative server. Queries for each label return more specific name servers until a name server returns the answer of the original query. Messages are sent over UDP; so you will be using datagram sockets for passing the messages.

In this assignment, you are implementing a hierarchical DNS system, which contains a local server, Root DNS servers, TLD servers and authoritative servers. The address resolution mechanism can be either through iterative query or through recursive query.

<u>**Iterative query**</u>
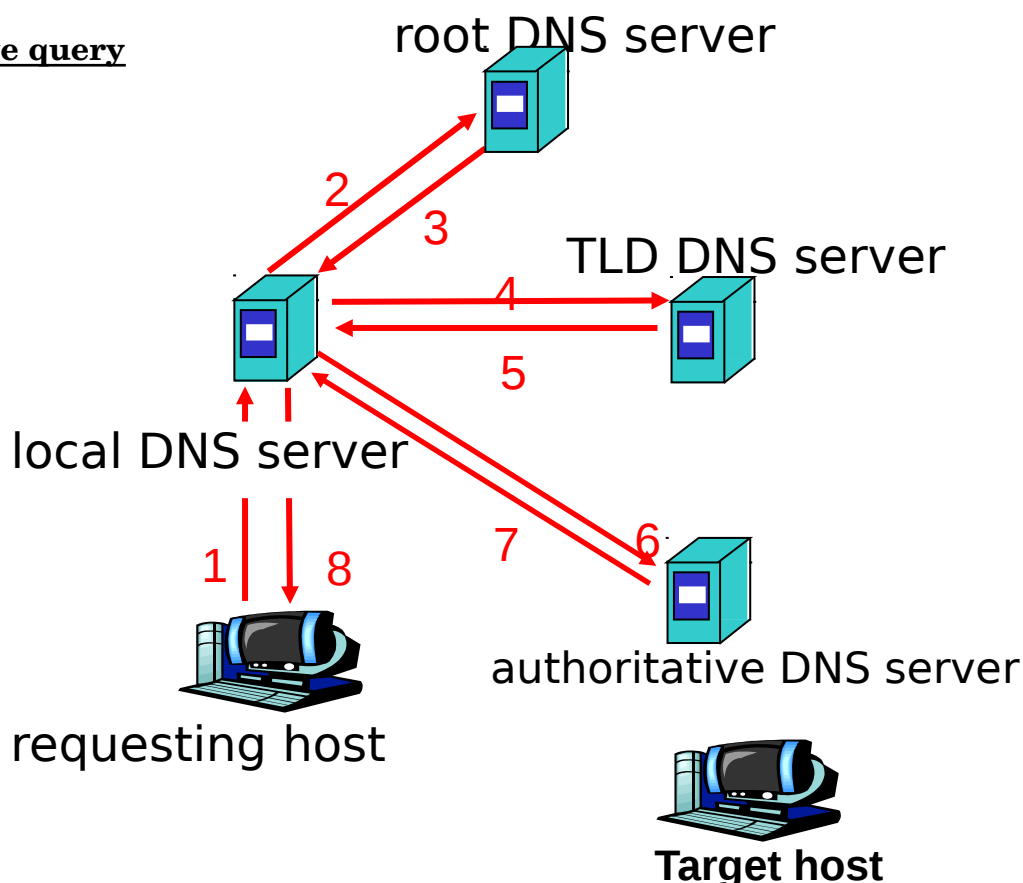
Your solution will have a client program and a server program for each of the DNS servers. Your resource records should conform to the standard format as mentioned below:

A resource record is a four-tuple that contains the following fields:
(Name, Value, Type, TTL)

TTL is the time to live of the resource record; it determines when a resource should be removed from a cache. In the example records given below, we ignore the TTL field. The meaning of Name and Value depend on Type:

• If Type=A, then Name is a host name and Value is the IP address for the hostname. Thus, a Type A record provides the standard hostname-to-IP address mapping. As an example, (relay1.bar.foo.com, 145.37.93.126, A) is a Type A record.

• If Type=NS, then Name is a domain (such as foo.com) and Value is the host-name of an authoritative DNS server that knows how to obtain the IP addresses for hosts in the domain. This record is used to route DNS queries further along in the query chain. As an example, (foo.com, dns.foo.com, NS) is a Type NS record.

• If Type=CNAME, then Value is a canonical hostname for the alias hostname Name. This record can provide querying hosts the canonical name for a host-name. As an example, (foo.com, relay1.bar.foo.com, CNAME) is a CNAME record.

• If Type=MX, then Value is the canonical name of a mail server that has an alias hostname Name. As an example, (foo.com, mail.bar.foo.com, MX) is an MX record. MX records allow the hostnames of mail servers to have simple aliases.

**Once the client and the server parts are set up, the client executes a command of the following form to resolve IP address of a host name:**
        *nslookup abc.com*

*Use sendto and recvfrom commands for the implementation. You should simulate the iterative querying mechanism.*

Make a design document listing out the design decisions and functions used. Please make sure that your code is conforming to the design.

*[Note: 1. nslookup is a network administration command-line tool available for many computer operating systems for querying the Domain Name System (DNS) to obtain domain name or IP address mapping or for any other specific DNS record.*

*2. Read the DNS portion in the text book thoroughly since it might help to do modifications part].*