**Assignment No.08 (Group A-11)**

**Title:** Write a program for DNS lookup. Given an IP address input, it should return URL and vice-versa.

**Theory:**

The Domain Name Systems (DNS) is the phonebook of the Internet. Humans access information online through domain names, like nytimes.com or espn.com. Web browsers interact through Internet Protocol (IP) addresses. DNS translates domain names to [IP addresses](https://www.cloudflare.com/learning/dns/glossary/what-is-my-ip-address/) so browsers can load Internet resources.

Each device connected to the Internet has a unique IP address which other machines use to find the device. DNS servers eliminate the need for humans to memorize IP addresses such as 192.168.1.1 (in IPv4), or more complex newer alphanumeric IP addresses such as 2400:cb00:2048:1::c629:d7a2 (in IPv6).

The process of DNS resolution involves converting a hostname (such as www.example.com) into a computer-friendly IP address (such as 192.168.1.1). An IP address is given to each device on the Internet, and that address is necessary to find the appropriate Internet device - like a street address is used to find a particular home. When a user wants to load a webpage, a translation must occur between what a user types into their web browser (example.com) and the machine-friendly address necessary to locate the example.com webpage.

In order to understand the process behind the DNS resolution, it’s important to learn about the different hardware components a DNS query must pass between. For the web browser, the DNS lookup occurs “ behind the scenes” and requires no interaction from the user’s computer apart from the initial request.

DNS is what translates your familiar domain name (www.google.com) into an IP address your browser can use (173.194.33.174). Before the page and any resource on the page is loaded, the DNS must be resolved so the browser can establish a TCP connection to make the HTTP request. In addition, for every external resource referenced by a URL, the DNS resolution must complete the same steps (per unique domain) before the request is made over HTTP. The DNS Resolution process starts when the user types a URL address on the browser and hits Enter. At this point, the browser asks the operating system for a specific page, in this case google.com.

**Step 1: OS Recursive Query to DNS Resolver**

Since the operating system doesn’t know where “www.google.com” is, it queries a DNS resolver. The query the OS sends to the DNS Resolver has a special flag that tells it is a “recursive query.” This means that the resolver must complete the recursion and the response must be either an IP address or an error.

For most users, their DNS resolver is provided by their Internet Service Provider (ISP), or they are using an open source alternative such as Google DNS (8.8.8.8) or OpenDNS (208.67.222.222). This can be viewed or changed in your network or router settings. At this point, the resolver goes through a process called recursion to convert the domain name into an IP address.

DNS Settings on a Mac (left) and Windows Settings for IPv4 Protocol of the network connection (right).

**Step 2: DNS Resolver Iterative Query to the Root Server**

The resolver starts by querying one of [the root DNS servers](http://www.iana.org/domains/root/servers) for the IP of “www.google.com.” This query does not have the recursive flag and therefore is an “iterative query,” meaning its response must be an address, the location of an authoritative name server, or an error. The root is represented in the hidden trailing “.” at the end of the domain name. Typing this extra “.” is not necessary as your browser automatically adds it.

There are 13 root server clusters named [A-M with servers in over 380 locations](http://www.root-servers.org/). They are managed by 12 different organizations that report to the Internet Assigned Numbers Authority (IANA), such as Verisign, who controls the A and J clusters. All of the servers are copies of one master server run by IANA.

**Step 3: Root Server Response**

These root servers hold the locations of [all of the top level domains](http://data.iana.org/TLD/tlds-alpha-by-domain.txt) (TLDs) such as .com, .de, .io, and newer generic TLDs such as .camera.

The root doesn’t have the IP info for “www.google.com,” but it knows that .com might know, so it returns the location of the .com servers. The root responds with a list of the 13 locations of the .com gTLD servers, listed as NS or “name server” records.

**Step 4:  DNS Resolver Iterative Query to the TLD Server**

Next the resolver queries one of the .com name servers for the location of google.com. Like the Root Servers, each of the TLDs have 4-13 clustered name servers existing in many locations. There are two types of TLDs: country codes (ccTLDs) run by government organizations, and generic (gTLDs). Every gTLD has a different commercial entity responsible for running these servers. In this case, we will be using the gTLD servers controlled by Verisign, who run the .com, .net, .edu, and .gov among gTLDs.

**Step 5: TLD Server Response**

Each TLD server holds a list of all of the authoritative name servers for each domain in the TLD. For example, each of the 13 .com gTLD servers has a list with all of the name servers for every single .com domain. The .com gTLD server does not have the IP addresses for google.com, but it knows the location of google.com’s name servers. The .com gTLD server responds with a list of all of google.com’s NS records. In this case Google has four name servers, “ns1.google.com” to “ns4.google.com.”

**Step 6: DNS Resolver Iterative Query to the Google.com NS**

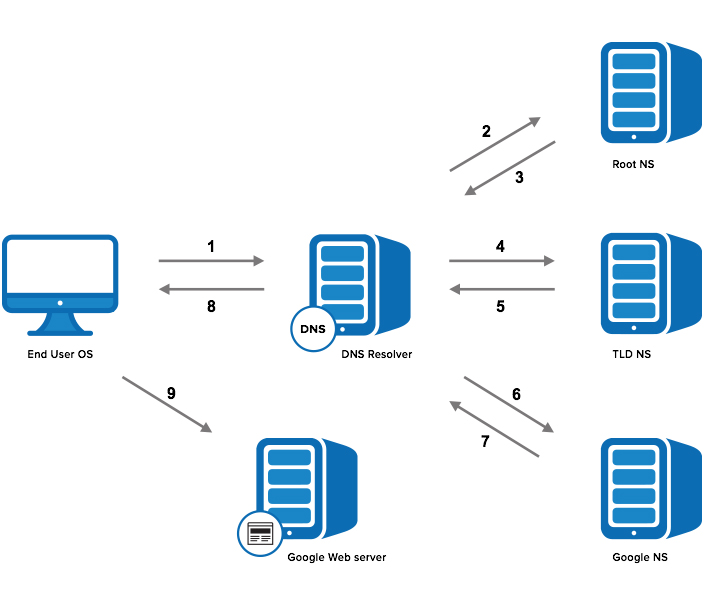
Finally, the DNS resolver queries one of Google’s name server for the IP of “www.google.com.”

**Step 7: Google.com NS Response**

This time the queried Name Server knows the IPs and responds with an A or AAAA address record (depending on the query type) for IPv4 and IPv6, respectively.

**Step 8: DNS Resolver Response to OS**

At this point the resolver has finished the recursion process and is able to respond to the end user’s operating system with an IP address.

[](http://blog.catchpoint.com/wp-content/uploads/2014/06/dns101.jpg)

**Step 9: Browser Starts TCP Handshake**

At this point the operating system, now in possession of www.google.com’s IP address, provides the IP to the Application (browser), which initiates the TCP connection to start loading the page. For more information of this process, we wrote a [blog post on the anatomy of HTTP](http://blog.catchpoint.com/2010/09/17/anatomyhttp/).

**Conclusion:**

Thus ,We implemented program for DNS lookup