

Assignment 8

Problem Statement: Write a program for DNS lookup. Given an IP address input, it should return the URL and vice versa.

Objectives:

1. To get the hostname and IP address.
2. Map the host name with IP address and vice-versa

Learning outcomes:

Demonstrate DNS lookup by IP address or URL

Requirements:

1. Open source linux based OS
2. Python interpreter or Javac/OpenJDK

Theory:

Need for DNS

To identify an entity, TCP/IP protocols use the IP address, which uniquely identifies the connection of a host to the internet. However, people prefer to use names instead of numeric addresses. Therefore, we need a system that can map a name to an address of vice-versa.

This is achieved by dividing the huge amount of information into smaller parts and storing each part on a different computer. In this method, the host that needs mapping can contact the closest computer holding the needed information. This method is used by the Domain Name System (DNS).

The following six steps map the host name to an IP address:

1. The user passes the host name to the file transfer client.

2. The file transfer client passes the host name to the DNS client.
3. We know that each computer, after being booted, knows the address of one DNS server. The DNS client sends a message to a DNS server with a query that gives the file transfer server name using the known IP address of the DNS server.
4. The DNS server responds with the IP address of the desired file transfer server.
5. The DNS client passes the IP address to the file transfer server.
6. The file transfer client now uses the received IP address to access the file transfer server.

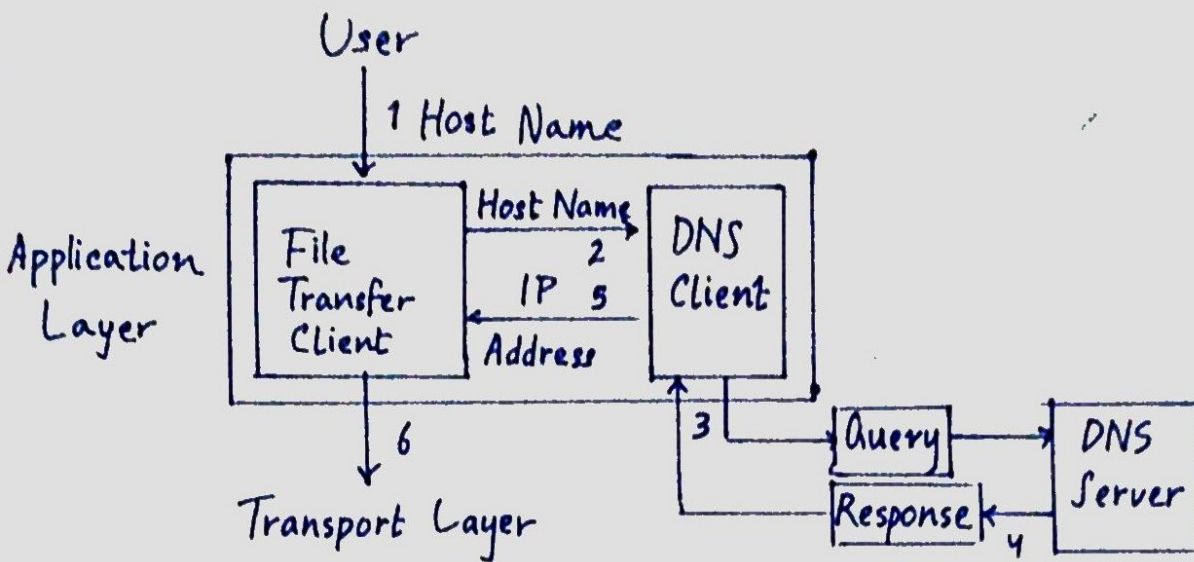
Name Spaces

To be unambiguous, the names assigned to machines must be carefully selected from a namespace with full control over the binding between the names and IP addresses. It can be organized in 2 ways:

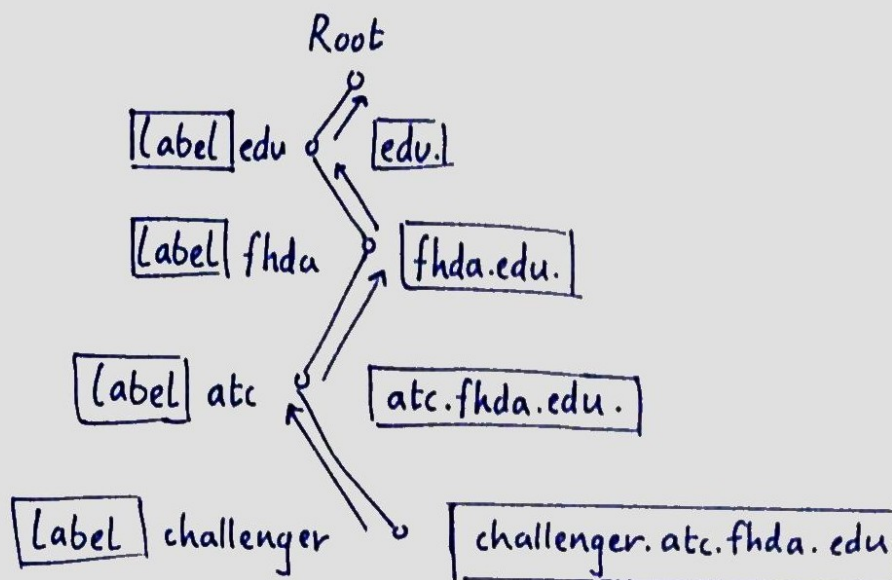
1. Flat Name Space: A name in this space is a sequence of characters without structure. Thus, it cannot be used in a large system such as the Internet.
2. Hierarchical Name Space: A name in this space is made up of several parts, like the nature, the name and the departments in the organization and so on. To have a hierarchical namespace, a Domain Name Space was designed. In this design, the names are defined in an inverted-tree structured with the root at the top. The tree can only have 128 levels.

Label: Each node in the tree has a label, which is a string with a maximum of 63 characters. The root label is a null string. DNS requires the children of a node to have different labels, which guarantees the uniqueness of the domain names.

Domain Name: Each node in the tree has a domain name. A full domain name is a sequence of labels separated by dots. The domain names are read from the node up to the root. The last label is the label of root (null).



Name-Address mapping by TCP/IP



Domain Name Tree

Domain: A domain is a subtree of the domain name space. The name of the domain is the name of the node at the top of the subtree. A domain may be divided into other domains, called subdomains.

Resolution:

Mapping a name to an address or vice-versa is called name-address resolution.

DNS is designed as a client-server application. A host that needs to map an address to a name or vice-versa is called a resolver. The resolver accesses the closest DNS server with a mapping request. If the server has the information, it satisfies the resolver. Otherwise, it either refers the resolver to other servers or asks other servers to provide the information. After the resolver receives the mapping, it interprets the response to see if it is a real resolution, or an error, and finally delivers the result to the process that requested it.

Conclusion:

We have learnt how DNS lookup works and have successfully implemented a program to perform DNS lookup.