**SOCKET PROGRAMMING ASSIGNMENT**

**DESIGN**

**DNS SERVER-CLIENT**

**NETWORK:**

Our proposed system consists of multiple clients and a single server which acts as the DNS server, simulated on the same machine with the use of different virtual machines. The request from different clients will be handled by the server with the use of threads.

**DATA STRUCTURES USED:**

**Server:** The server will maintain a file which consists of the domain names and their respective IP addresses. Upon starting the system, the data stored in the file will be loaded into a hash table, which will be used for IP address resolution when the request comes from the client. In those cases where server does not have the address for the requested domain name, it gets the IP address from the global DNS server. It makes an entry into its data structure and then passes it on to the client.

**Client:** The client also maintains a hash table which serves as a local cache. In this data structure the most frequently used addresses by the client would be stored and the entries are updated or deleted on most frequently used basis, by the help of a timer.

**ALGORITHM:**

1. The clients and server are simulated using different virtual machines and are connected using UDP sockets.
2. The client checks for a match in its local cache for the domain name. If it is a hit, it uses the mapped IP address. If it is a miss, it goes to step 3.
3. The client sends a request to the server asking for the IP address. If the server finds it in its data structure, the IP address is sent to the client in the form of a reply. If the server does not find the mapping, go to step 4.
4. The server uses the function gethostbyname(), which in turn asks the global DNS server for address resolution. On receiving a reply, the server adds the new mapping to its data structure and sends the response to the client.
5. The client cache is updated based on the most frequently used addresses. This can be implemented by using a count variable and a timer.

**DESIGN DIAGRAM:**

