**DNS Client Library and Server Simulations**

DNS Library (ClientDNS)

(DNS Resolver, DNS Controller)

Browser Client

DNS Server port B

Cache  
CaCa

Cache  
CaCa

DNS Server port A

DNS “Local” Server

UDP

UDP

UDP

Browser client calls ClientDNS.getAddressByName(String hostname)

DNS Server port C

Cache  
CaCa

DNS Controller calls the local server and coordinates further queries if the local server does not have the IP address mapping in its cache. If there are multiple IP address mappings to one hostname, it also determines how the resolver chooses which IP address to return.

Uses DNSResolver to read the mapping and return the IP address (16 byte character string) to the browser.

Cache  
CaCa

Fixed format for getAddressByName requests:

Assume that each hostname always consists of 3 segments separated by periods: the name of the host, the second-level domain name, and the top-level domain name. The host name and the second-level and top-level domain names are strings of 3 letters. We also assume a flat file system, i.e., no directories. Each file name contains only a string of 4 letters followed by “.htm” extension.

For the DNS request from the client, the Browser needs to extract the full hostname + second-level and top-level domain names into a 12 character string (i.e., 12 bytes, last character is blank) in the format “ABC.CDE.COM” where each segment listed above can be represented by any letter.

ClientDNS only considers three levels in the domain hierarchy, including the root, top level, and second level. ClientDNS communicates with the DNS servers over UDP. UDP protocol should be used for DNS communication. The DNS server at different levels simply extracts the partial string that represents the domain name it can serve and finds the mapping.

The ClientDNS response to the Browser returns a 16 byte character string, padded if the actual IP address string is shorter than 16 characters.