

For Project 1, you will implement a Simplified DNS consisting of a client program and two servers: RS and TS. In HW1, you have already implemented a client server program with one socket in each program. In this project, you will extend the HW1 implementation to have two sockets in the client program. One socket will communicate with the RS server and the other with the TS server.

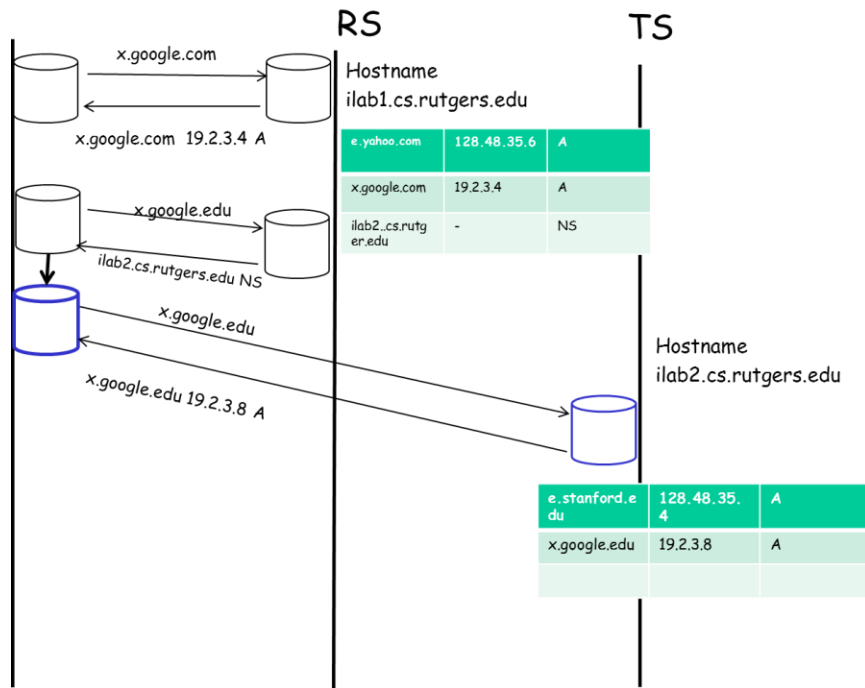
The RS server and the TS server each maintain a `DNS_table` consisting of three fields: Hostname, IP address, Flag (A or NS) (you need to choose the appropriate data structure to store the values for each entry). The client always connects first to the RS server and sends the hostname as a string. The RS server does a look up in the `DNS_table` and if there is a match, sends the entry as a string "Hostname IPaddress A" else sends the name of the TS server as a string "TSHostname - NS". Here, TSHostname is the name of the server on which the TS server program is running. If the client receives, a string with "A" field, it outputs the string consisting of "Hostname IPaddress A".

On the other hand, if the client receives a string with "NS" field, it uses the hostname part in the string to determine the IP address of TS server and connects to the TS server using the second socket. The client then sends the hostname as a string to the TS server. The TS server looks up in its `DNS_table` and if there is a match sends the entry as a string "Hostname IP address A" else sends an error string "Hostname - Error:HOST NOT FOUND". The client outputs the received string from the TS server.

The hostname strings will be given one per line in a file (PROJ1-HNS.txt). The DNS tables entries, one for each server, will also be strings (fields separated by space) one per line and will be in (PROJ1-DNSRS.txt and PROJ1-DNSTS.txt). Your server programs should populate the DNS table by reading the entries from the corresponding files. Your client program should output the results to a file RESOLVED.txt. As part of your submission, you need to submit three program files as well as the output file.

As always, first start the TS server, RS server and then the client program. Figure out how you will communicate the port number and hostnames of TS and RS servers to the client program. Choose a suitable data structure for the DNS table.

A brief sketch of the interaction among the programs as well as a high level sketch and pseudo code is provided for the three programs.



#TS server

```
try:
    tsdd =
mysoc.socket(mysoc.AF_INET,
mysoc.SOCK_STREAM)

except mysoc.error as err:
    print('[TS]: {}
\n'.format("socket open error ",
err))
determine local hostname, IP
address , select a port number

tsdd.bind(...)
ctsd,addr=tsdd.accept()
hnstring=ctsd.recv(...)
if hnstring in TS_table:
    entry=TS_table(hnstring)
else:
    entry= "hname" + "Error:
Host not found"
ctsd.send(entry)

.....
```

#RS server

```
try:
    rssd =
mysoc.socket(mysoc.AF_INET,
mysoc.SOCK_STREAM)

except mysoc.error as err:
    print('[RS]{} \n'.format("RS
server socket open error ", err))

determine local hostname, IP
address , select a port number

rssd.bind(...)
crsd,addr=rssd.accept()
hnstring=crsd.recv(...)
if hnstring in RS_table:
    entry=TS_table(hnstring)
else:
    entry= TS-table-
row(flag='NS')
crsd.send(entry)

....
```

#Client

```
try:
    [ first socket]
ctots=mysoc.socket(mysoc.AF_INET,
mysoc.SOCK_STREAM)

except mysoc.error as err:
    print('{} \n'.format("socket
open error ",err))

[second socket]
try:
    ctots=mysoc.socket(mysoc.AF_INET,
mysoc.SOCK_STREAM)

except mysoc.error as err:
    print('{} \n'.format("socket
open error ", err))

[determine hostname of RS server
and port ]
[bind ctots socket to RS address,
rport]
#First Connect to RS server
ctots.send("hostname",RSserver )
dr=ctots.recv(...)
if flag field(dr) == 'A'
output(dr)
else
    if flag-field (dr) == 'NS'
[connect to TS server]
    TSname= hostname-field(dr)
[Determine IP address of TSname
    bind ctots socket to TS address,
    tport]
    ctots.send(hostname)
[Connect and send hostname string ]
dr=ctots.recv(...)
output (dr)
```