NetProg Assignment-2 Q3

1 fastertraceroute

1.1 Usage

make fastertraceroute
sudo ./fastertraceroute <url>

Sample use $ sudo ./fastertraceroute www.google.com

Parameters:
<url> - The url for which you want to traceroute to (must be able to look it up with DNS).

1.2 Implementation

After running the command, the program gets the actual IP address of the url using DNS. After receiving the actual IP address, the program splits up into one thread for each TTL from 1 to 10 and 1 thread that reads all the ICMP packets that have been sent back.

Whenever an ICMP reply is read by the icmp thread, the thread signals the corresponding UDP thread for that TTL (the TTL is found by analyzing the port number of the ICMP reply). It places the receiving address in a global variable and signals the thread, which causes it to process the address and print the information.

2 findLongestCommonPath

2.1 Usage

make findLongestCommonPath
sudo ./findLongestCommmonPath <domains.txt>

Sample use $ findLongestCommonPath domains.txt

Parameters:
<domains.txt> - The txt file that contains the urls of all the websites for which you would like to find the longest common path for.

2.2 Implementation

When run, the process creates 1 socket for each file in the domains.txt file. It also creates an icmp socket to read all the replies. It adds all the sockets into a select call and waits for the ICMP socket to be readable and the UDP socket to be writable

After receiving the message on the ICMP socket, it processes the message using the sending port (each sending port is mapped to a domain and TTL) and stores the information in an array. Later, after the select() times out, we process all the messages and find the longest common path for all the domains.