P1 - RTT via I/O multiplexing

Submitted By : Abdul Kadir Khimani, Ayush Singh, Nayan Khanna
Submitted for : IS F462, Network Programming.
IC : Dr. Hari Babu

Design Decisions :

We implement a program to find the round trip times for multiple IPs using a single program. It works for IPv4 and IPv6 addresses.

Assumptions :

1. IPv4 and IPv6 supported.
2. Instead of a numeric string IP address, website URLs can also be used.
3. We send 3 ICMP packets and do not retransmit in case of packet loss.

Implementation :

We create a new raw socket for each new IP address. ICMP packet for a particular IP is identified by a combination of PID and socket descriptor value, plugged in the ID field of ICMP request packets. We use epoll API for I/O multiplexing, due to its performance over select and poll for N > 10000 file descriptors. A socket is closed once we receive 3 replies for that particular IP address.

Execution flow :

Our program has two threads of execution. The main thread and the receiving thread. Both threads take care of IPv4 and IPv6 packets using protocol-specific functions.

Main Thread :

1. It creates an epoll instance.
2. It loops line by line, reads the IP address on each line from a file.
3. For each IP, it creates a new raw socket, adds this socket to the interest list of epoll API, saves details for this remote host in a structure, sends 3 ICMP packets in succession, and continues to the next IP address.

Receiving Thread :

1. Another thread takes care of receiving replies from corresponding sockets for each IP address and calculates the RTT and stores it in a buffer.
2. It uses I/O multiplexing using epoll on the socket fds.
3. On receiving 3 replies, it prints the RTT values and closes the socket.

Usage :

Change to the directory in which the source code is present, then

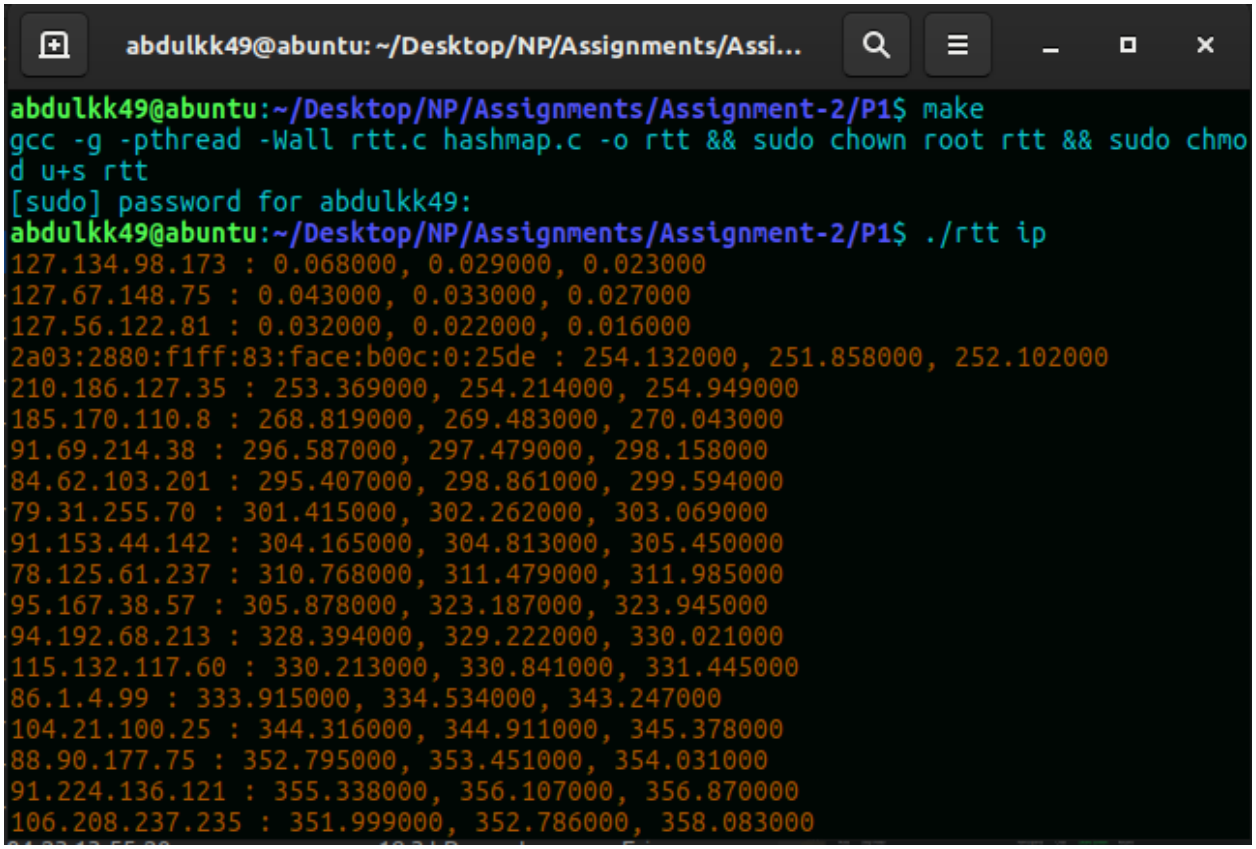
Compile :

In the bash terminal, issue command: **make** or **make rtt**

Run :

In the bash terminal, issue command: **./rtt <filename>**

Sample :


abdulkk49@abuntu: ~/Desktop/NP/Assignments/Assi...
abdulkk49@abuntu:~/Desktop/NP/Assignments/Assignment-2/P1$ make
gcc -g -pthread -Wall rtt.c hashmap.c -o rtt && sudo chown root rtt && sudo chmo
d u+s rtt
[sudo] password for abdulkk49:
abdulkk49@abuntu:~/Desktop/NP/Assignments/Assignment-2/P1$ ./rtt ip
127.134.98.173 : 0.068000, 0.029000, 0.023000
127.67.148.75 : 0.043000, 0.033000, 0.027000
127.56.122.81 : 0.032000, 0.022000, 0.016000
2a03:2880:f1ff:83:face:b00c:0:25de : 254.132000, 251.858000, 252.102000
210.186.127.35 : 253.369000, 254.214000, 254.949000
185.170.110.8 : 268.819000, 269.483000, 270.043000
91.69.214.38 : 296.587000, 297.479000, 298.158000
84.62.103.201 : 295.407000, 298.861000, 299.594000
79.31.255.70 : 301.415000, 302.262000, 303.069000
91.153.44.142 : 304.165000, 304.813000, 305.450000
78.125.61.237 : 310.768000, 311.479000, 311.985000
95.167.38.57 : 305.878000, 323.187000, 323.945000
94.192.68.213 : 328.394000, 329.222000, 330.021000
115.132.117.60 : 330.213000, 330.841000, 331.445000
86.1.4.99 : 333.915000, 334.534000, 343.247000
104.21.100.25 : 344.316000, 344.911000, 345.378000
88.90.177.75 : 352.795000, 353.451000, 354.031000
91.224.136.121 : 355.338000, 356.107000, 356.870000
106.208.237.235 : 351.999000, 352.786000, 358.083000