Assignment 1

COL 334/672

Due date: August 21, 2016, 23:55 hours (Sunday)

Note: *Solve all problems on your own.* Approach the instructor for clarifications. Your solutions must be submitted as a pdf report (you can choose to use Word, Latex etc. to create the report). Upload the report along with your socket programming code to moodle according to the instructions given at the end of the assignment. Note that you are expected to use a tool called `netem` which comes installed on Linux boxes by default.

1. From a machine in your hostel or favourite lab check your download and upload speed using `http://www.testmyspeed.com/`. What are the download and upload speeds you get? Compare these to the results you get from `www.speedtest.net`. Are the results more or less the same? If not, suggest reasons for why they are different.

2. Learn about the networking utility `ping` from the WWW.

    (a) Write a one-paragraph description of it in your own words. You should mention what protocol it uses and the method used to generate echo packets from the remote host.

    (b) Use the `ping` command to determine the RTT to www.google.com, www.rice.edu, and www.iitd.ac.in. Mention the IP addresses of these web servers along with the RTT to them in milli-seconds. Which of the three servers is closest in terms of RTT. Which is farthest in terms of RTT? Why do you think is the reason for this?

3. Run the `ifconfig` (on Windows use `ipconfig` or `netsh`) command to get information about various interfaces on your machine.

    (a) List the IP addresses (IPv4) of all interfaces. These are of the type `a.b.c.d`.

    (b) A few interfaces might have Ethernet addresses (`ether ..`). List the names of these interfaces, their Ethernet addresses, along with the `mtu` for those interfaces. What does MTU refer to?

    (c) Some interfaces may have IPv6 addresses listed against them? List these out, if any. How many bits does an IPv6 address have?

4. `traceroute` is a networking diagnostic tool which gives the IP addresses of routers on a path to a given destination and the RTTs to each router.

    (a) Run tracroute to `www.iitd.ac.in` and `www.cse.iitd.ac.in`. Discuss the differences in the list of routers (how many are common and how many different) on both paths. In theory the RTT to routers further along the path should be larger than for those closer to the source (your machine running tracroute). Is this always the case in your experiments? If not, suggest why.

5. You can learn socket programming basics and use C source code from "Beej's Guide to Network Programming" http://beej.us/guide/bgnet/ to implement this assignment. Write two socket programs, client.c and echo.c, that together communicate using

"Datagram Sockets". Each of these when run with the "-h" option should show their usage (example: `client -h`).

Each datagram generated by client.c must contain (a) a sequence number which identifies the packet, (b) a timestamp with microsecond level precision which indicates the time at which the packet is first transmitted, and (c) an even non-negative integer called the *reflection count* (RC) field with initial value T. Let us assume that the packet is of size P bytes.

When echo.c receives a datagram from client.c, it immediately decrements the RC value in the datagram and sends the same datagram (with the new RC) back to client.c. The client.c program on receiving a datagram from echo.c, decrements the RC value, and checks if this new value is zero. If the new RC is greater than zero, then client.c sends the datagram (with the new RC) back to echo.c.

However, if RC is zero, then client.c prints to a file (on a new line) the difference between the current time and the timestamp field in the datagram. Call this time the "cumulative RTT". A new datagram is then generated by client.c with RC set to T. Your program should take care of *packet loss*: if a timer expires and no echo packet has reached the client, then it assumes that packet loss has occurred.

The value of P and T, and the output file name for storing the "cumulative RTT" should be entered on the command line when executing client.c. P should be within the range 100 to 1300 bytes, and T between 2 and 20000 (and must be even).

The client.c program totally sends out 50 datagrams and then quits. Run client.c and echo.c on two different machines.

(a) For $T = 2000$ and different values of $P = 100, 200, \ldots, 1000$, run client.c. Plot a scatterplot (using any suitable software, such as matlab, gnuplot etc.) of "cumulative RTT for all 50 datagrams" vs. P for the different values of P when $T = 2000$. What do you observe? What information does the slope of the graph contain? Repeat when $T = 8000$. Add the plots, and your observations for each plot, to your report.

(b) Repeat the same as above but now using `netem` to introduce a random "normal" distribution of delay with mean 5ms and 2ms standard deviation, as well as random packet loss of 1%. Add the plots, and your observations for each plot, to your report. How is the scatterplot qualitatively and quantitatively differenent from the case with no netem delay/loss added?

Comment your socket programming code well. Upload a single zip file with name ENTRYNUMBER_HW1.zip (e.g.: 2014CS10100_HW1.zip) to moodle containing (i) client.c, (ii) echo.c, (iii) Makefile, (iv) README, and (iv) the pdf report.