UTSA
The University of Texas at San Antonio™

CS 3873: Computer Networks, Fall 2019 – Project 5

Assign Date: Wednesday, October 23rd

Due Date: Wednesday, November 6th

Project Description and Instructions:

The purposes of this project is:

- To measure the average Round Trip Time (RTT) between your UDP client and my UDP server.
- To estimate the amount of segment loss over some testing interval.

Your UDP client program (which you coded in project 3) will have to be modified to do these things:

- Enable it to asynchronously send and receive datagrams by using **threads** (remember, you implemented a multi-threaded program in project 4). For details on how to make this modification, please refer to the flowchart below indicating the suggested flow of your client code.
- It should measure the time interval between sending a datagram and receiving the datagram back (if it comes back).
- It should compare the number of datagrams sent with the number of datagrams received in order to estimate the total percentage of datagram loss.

I will run two instances of my "echo" server. The one on port 7851 will always send back what it receives. The second one running on port 8591 will randomly drop a specified percentage of received segments. The target address of my UDP server is the same as before: 69.61.103.44. Your client will not know this datagram drop percentage (of the second server) and, to get full credit, you should estimate it within 10% of the actual value.

Here are some things to consider that will make your job easier. We will discuss these in class.

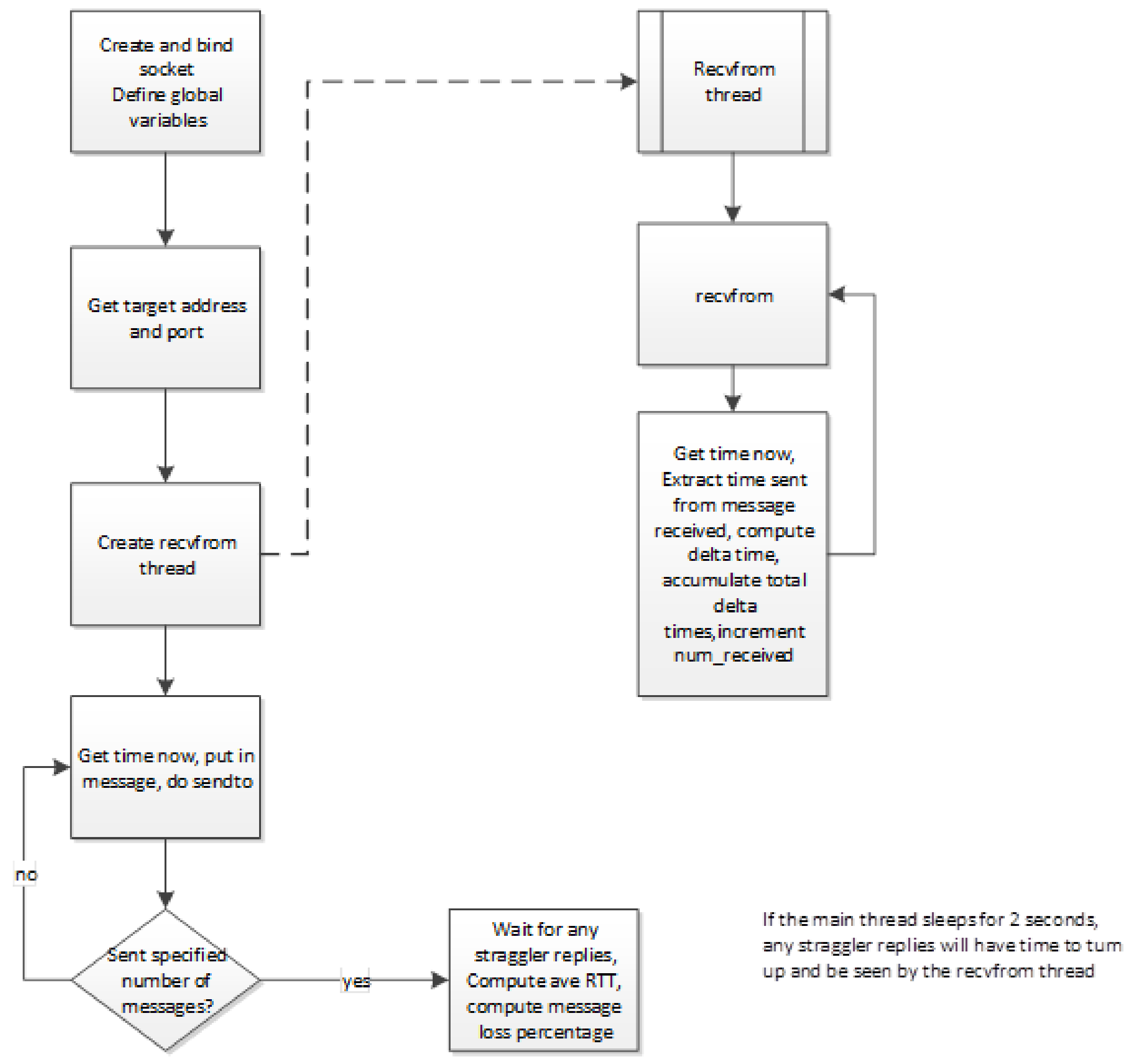
- What is the best way to measure RTTs? How do we keep from getting false measurements when a segment does not come back?
- Which units should be used for our time measurements?
- Is the average RTT a good measure or would some other statistical quantity make it easier to interpret?
- Does the RTT say anything about the One-way Trip Times (OWTT)?
- Can you think of key network parameter you could estimate by measuring the change in the spacing between the sent segments and the received segments?
- What happens if you send datagrams as fast as possible? Can you create loss even using the lossless server? How should you modify your code in this situation?

The no-loss server running on port 7851 is for testing your program. The measured average RTT should be approximately the same as when you run with the lossy server on port 8591 as your target.

Guidelines and Deliverables:
This programming assignment can be either done individually or in groups of maximum TWO (2) students, formed at the beginning of the semester. As a deliverable for this project, you need to submit a single .zip file containing all the source code (.c files only) and a single README file (can be text file). The README file should contain results of your program's computation of RTT and loss percentage and any specific instructions for compiling and executing your code. Make sure that you submit on time as no submissions after the due date will be allowed. All source code files should clearly contain the names and UTSAIDs of all students in the group. All students must individually submit the zip file on Blackboard, even if the project is done in a group. Also, all students should individually execute their client (even if the project is done in a group) to interact with my server and include their individual responses in their README file that is included in the submission.

Grading guideline:
The total points for project is 100. 50 points is for your modified UDP client source code. The rubric for the source code grading is as follows: 50% logic, 25% comments and coding style and 25% for following good programming practices (suitable variable name usage, checking for errors and returning appropriately, proper memory management and function use). The remaining 50 points will be earned for your computations (25 points for RTT computation and 25 points for datagram loss computation). As mentioned before, you will get full credit for your datagram loss computation only if you are able to estimate the datagram loss of my server within 10% of its actual value.

Suggested Flow for your UDP Client Application


Create and bind socket Define global variables
Get target address and port
Create recvfrom thread
Get time now, put in message, do sendto
Sent specified number of messages?
no
yes
Wait for any straggler replies, Compute ave RTT, compute message loss percentage
Recvfrom thread
recvfrom
Get time now, Extract time sent from message received, compute delta time, accumulate total delta times, increment num_received
If the main thread sleeps for 2 seconds, any straggler replies will have time to turn up and be seen by the recvfrom thread