

Exercise #1: Benchmarking Throughput

Goal

The goal of this exercise is to learn how to measure point-to-point (unidirectional) throughput, by writing a C/C++ benchmark application on top of TCP sockets.

What do you need to measure?

You need to measure throughput between two machines, for exponential series of message sizes, ranging from 1 byte to 1MB. Throughput is the highest possible transmission rate with that message size.

How to measure throughput?

In order to measure throughput you need code both for a server and a client (try to make them share as much code as possible). You run your server first and the client second, the client connects to the server and sends X messages (you decide how many, and explain your decision in a comment inside the code), the server replies after all X have arrived, and the client calculates the throughput based on the time it all took (you can ignore the reply in your calculation).

Items to pay attention to:

1. The client output should have exactly three columns, delimited by a tab (a script will read them): the message size in bytes, the number (integer or float) and the unit of measurement. For example (not actual results):

Client machine	Server machine
#> client <server-ip>	#> server
1 2 lightyears	#>
2 11 MHz	
4 1.3 Volt	
...	
512 5 celsius	
1024 1 Kg/cm^2	
#>	

2. Use “warm-up cycles”. Document how you choose their amount (a couple of lines of comment in the code are enough).
3. There will be multiple students running on the same machine at the same time. Don’t be alarmed if the result is not exactly consistent, as long as you measure it correctly.
4. Please pay attention to the command line (will be auto-tested).

Bonus for best result: +5 points

The student submission reaching the best result in the class on the course hardware setup (and satisfies requirements) will get additional points, limited by 100 for the entire exercise.

Setup & Submission

For this exercise you may use any lab machine (subject to change in the future).

You should submit an archive named “<id1>_<id2>.tgz”, containing a Makefile (calling “make” will build the code), building “server” and “client” executables. Submission should include a report with short description of the implementation and the performance results.