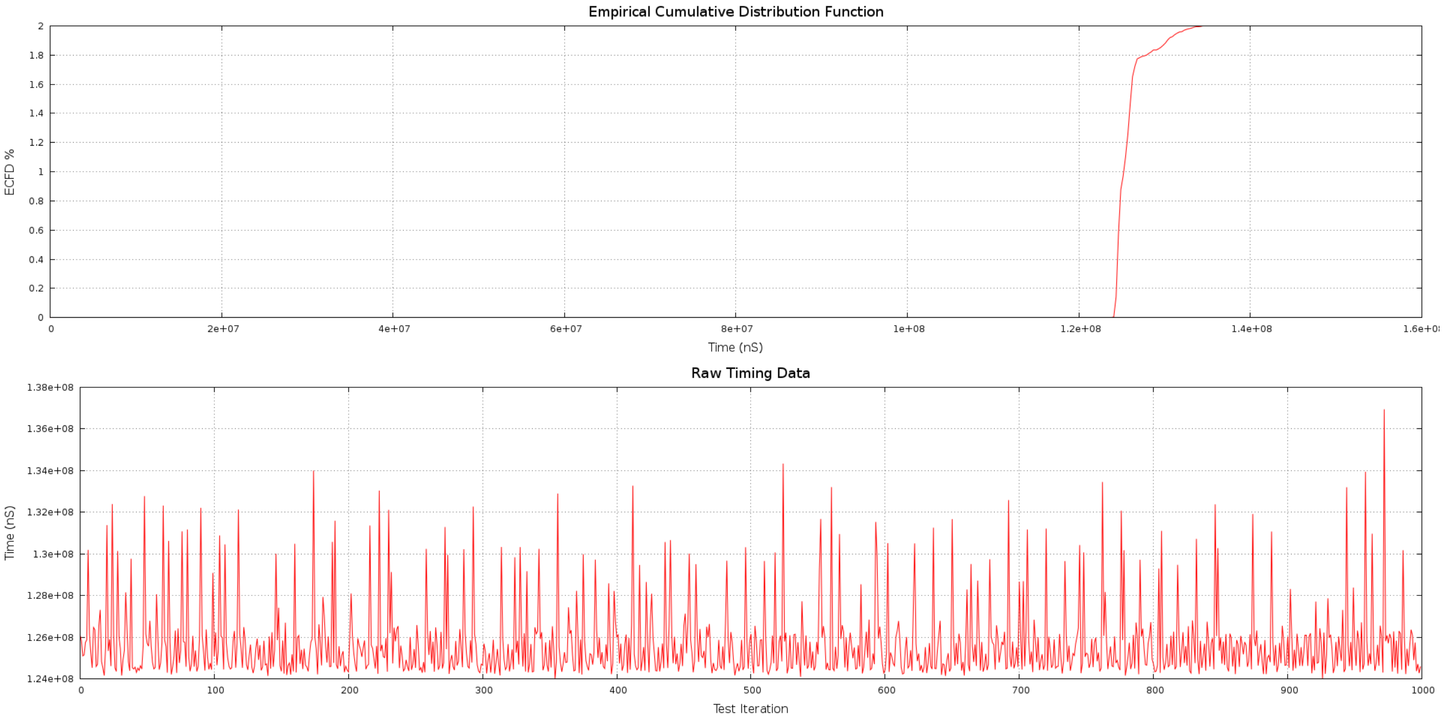
Eric Hair
Shakti Chaganty
Brent Mellor

ECE 6785 – Advanced Embedded Systems
Team Assignment 4

Timing Analysis

In order to test our ROT13 server we wrote a simple test harness which monitored the duration required to send and receive a simple string using blocking serial transfers. After collecting the raw timing data of 1,000 iterations, the test harness calculated the empirical cumulative distribution function. Both sets of data were plotted together using gnuplot. (Note: All CDF's are plotted on identical axis, the raw timing data is not.)

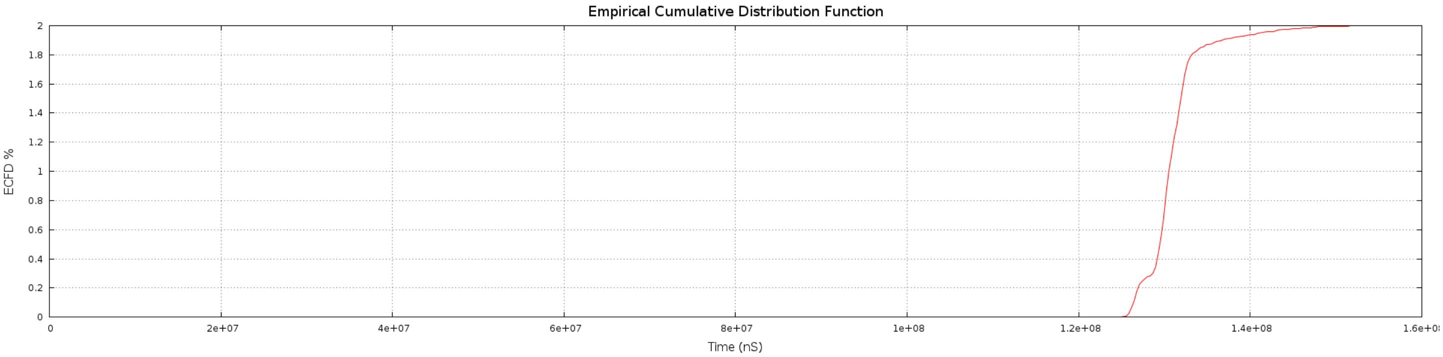
Unloaded Operation:

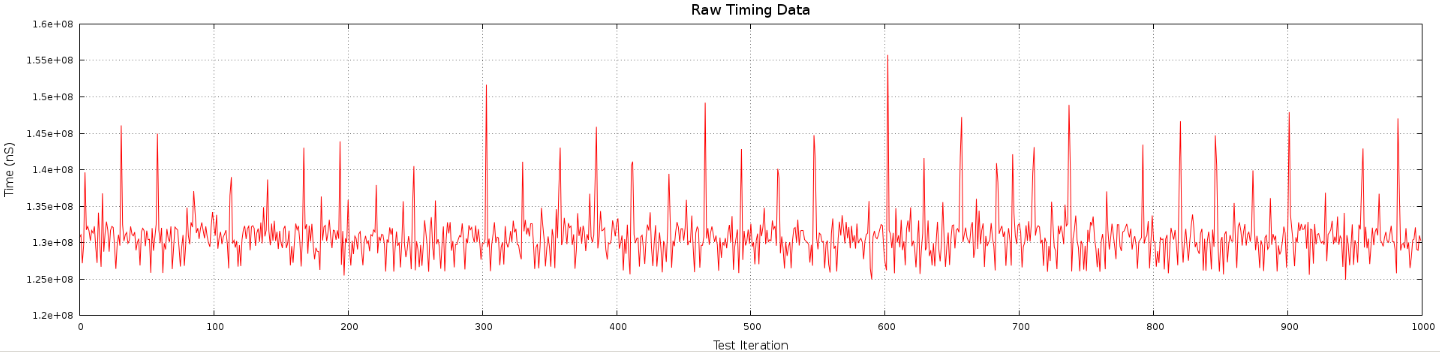


Loaded Operation #1:

Iperf reported speed = 80.5 Mbits/sec

ROT13 server run at standard priority.

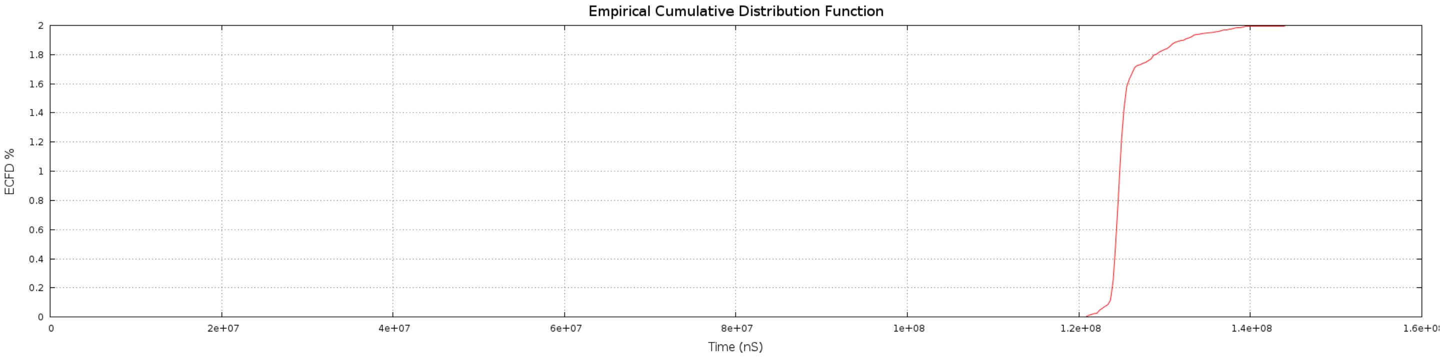

Empirical Cumulative Distribution Function

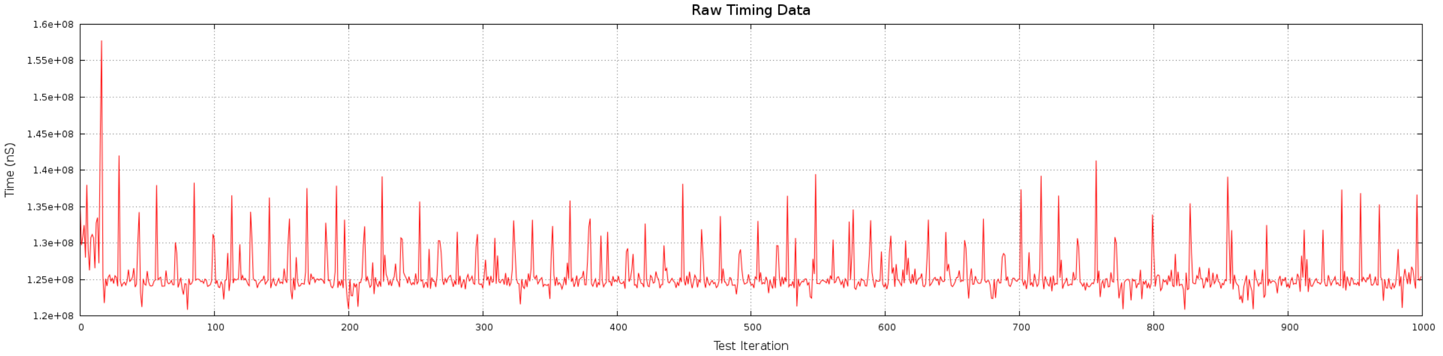

Raw Timing Data

Loaded Operation #2:

Iperf reported speed = 77.3 Mbits/sec

ROT13 server run using nice -18


Empirical Cumulative Distribution Function


Raw Timing Data

As seen by the difference between the last two CDF's we were able to improve the performance of the ROT13 server while the system was under load. For these tests we used the system "nice" utility to modify the importance of the ROT13 server in comparison to other system tasks and programs.

Initially we had wanted to utilize the PRU coprocessors to manage the ROT13 server. These devices would have given us excellent latency and would have been completely immune to the effects of heavy system load. Unfortunately although we were able to get code running on the PRU system, we were unable to get the UART's to connect and function properly. Because of this we chose to do the much easier approach and use the built-in Linux priority system to manage the increased load.

Surprisingly the priority modification had only a minor effect on the Ethernet transmit speed. This is likely because the serial transmissions were very infrequent in comparison. If the rate of serial communications was more rapid, the higher priority of the server process would have had a more drastic effect on the rest of the system.

As far as the improvement to the serial timing, there was a noticeable difference between the iterations where the priority was elevated. In fact there were iterations where the duration was smaller than on the standard unloaded test. (Seen by the small foot/base of the cliff on the last CDF.) The overall performance however, was slightly less than the unloaded test.

If we had more time we would have liked to continue to work on the PRU system.