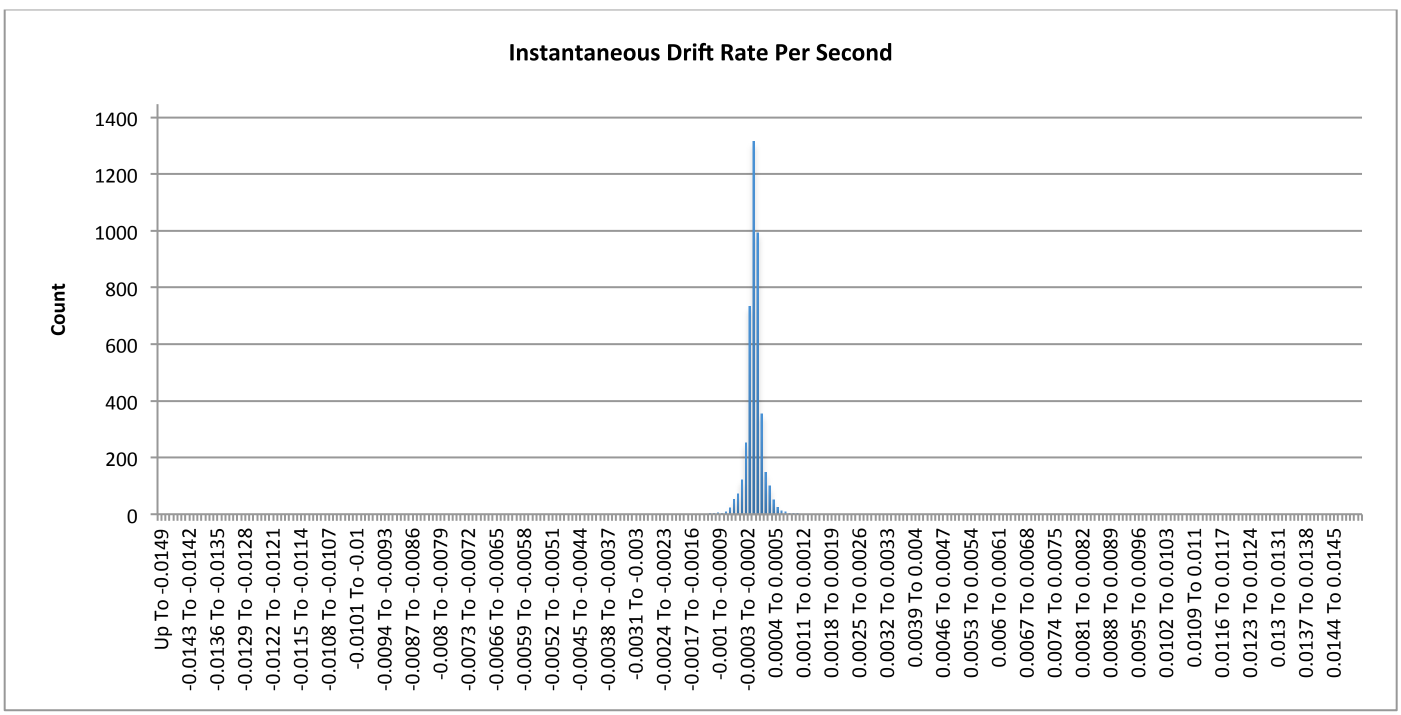
Marc Beitchman
CSE P 552
Assignment 1
4/12/2013

I used Python 2.7.4 to implement the time synchronization client. I chose Python because it is familiar and has good libraries for communicating over UDP. To run my code, run the following command line: *python.exe TimeSyncClient.py*. My program will log the current sequence number to standard output and log the required data data to a file named log.txt.

I calculated average clock drift rate by calculating the difference between the current smoothed instantaneous drift rate and the smoothed previous instantaneous drift rate value. I then averaged these values and divided by 10 to get the per-second rate.

I chose a timeout of 8 seconds to detect a failed interaction with the idea that a timeout could at most cause the client to miss one interaction waiting to receive data. I drop the packet if it is received out of order.

Below is the histogram for the “instantaneous” clock drift rate values for a 12-hour run with NTP disabled. I used the unsmoothed drift rate values and subtracted the current value from the previous to and then divided by 10 seconds to get the instantaneous unsmoothed drift rate per second values.



The reason the histogram seems to be shaped this way is due to the fact that clock drift rate appears to be mostly constant. I believe this is due to slight differences in the frequency of my local machines clock verses the server’s clock.

The following table shows average RTT, drop rate, and average drift rate for a 12-hour run with NTP disabled.

12-hour run with NTP disabled	
Average Round Trip Time	17.04 milliseconds
Packet Loss Rate	0%
Average Clock Drift Rate	-33.6 microseconds per second

I found it interesting that no packets were lost during this run. For a 33-minute test run, I had a 5.10% packet loss rate and for a 12-hour run with NTP on I had a 0.16% packet loss rate.