CS 466, Fall 2018  
Assignment #4  
Distributed Programming with Chocolate Coins!  
Due 12/7/18, in class

Introduction
Centralized algorithms are easy to write and understand, but have the drawback that they have a single point of failure. Truly distributed algorithms are harder to write and understand, but lack this “central” drawback.

Problem Statement
Given a set of houses each having received a different number of chocolate coins from Santa, write a program for truly distributed chocolate coin balancing in a neighborhood of five houses. A correct outcome is that each house has the same number of chocolate coins plus or minus one. Note that it is very hard for a distributed algorithm to verify that it has achieved any global state goal such as this one.

Analysis
After looking at the repo, think about exactly what functionality is missing.

Design
Unless you receive permission, you must use C and the socket code found in the repo.

What to hand in
(1) A 2-up listing of your well-formatted program. (You need not include socket.*.)
(2) Some 2-up annotated (with a highlighter and pen) sample output. Prefix each output line with the current time in milliseconds and house name. Doing so will enable you to easily sequence (chronologically) each program’s output.
(3) An updated repo.

Notes
(1) Each house starts with a random number of chocolate coins.
(2) Test your program using five different unix machines.
(3) Use the command line arguments “my_number[0...n-1] host0 host1 host3 host...host_n-1”.
(4) You might try running your program on a(n architecturally) heterogeneous collection of machines (e.g., some different laptops). What effect do you expect this to have? What effect did it have?

Hints
(1) You might want to write the centralized version first.
(2) Output sufficient information about each transaction to successfully reconstruct each.
(3) Giving someone negative chocolate coins is both impossible and rude!