CS5331: Mobile Data Management and Privacy
Spring 2023

Project #2: Heartbeat

- Name only: _____
- Release date: Feb 17th, 2023 (Friday)
- Due date: **Mar 10th, 2023 (Friday), submit through the Blackboard before 1:00 PM**
- It should be done individually; No handwritten code or report; No email submission; No late project will be accepted.
- Total 20 points

0. Access the Linux server (willow.cs.ttu.edu):
    - Please refer to "Accessing Willow and Troubleshooting Tips" uploaded in the Blackboard, download and install all the necessary programs and follow all the instructions.
    - The credentials you use to log into the server are your eRaider username and password.
    - You will need to download and install GlobalProtect for TTUnet VPN (vpn.ttu.edu) before logging into Willow from off-campus. You can refer the link (askit.ttu.edu) to find related instructions.
    - You will be given a temporary account and thus, your account will be removed at the end of semester. You have a limited amount of quota and please do not store anything except the project related files.

1. Implement a simple inter-process communication protocol that exchanges two control packets, *Hello* and *Hello_Ack*. Refer to the example code shown in lecture note #5.
    - Suppose there are five nodes in a network, where each node is fully connected with others. Each node generates a *Hello* packet periodically by following inter-arrival time, i.e., `exponential (5.0)`.
    - A sender node (in short, sender) transmits a *Hello* packet to a randomly chosen receiver node (in short, receiver). Upon receiving, the receiver replies to a *Hello_Ack* packet. If the sender receives the *Hello_Ack* packet, then the transmission is successful. Suppose it takes 0.2 seconds to transmit a packet over the network. A local processing delay before replying *Hello_Ack* is 0.1 second.
    - A packet can be lost during the transmission because of an unreliable link quality. Suppose a set of packet loss probabilities is 0.1, 0.2, 0.3, 0.4, and 0.5. If the sender does not receive the *Hello_Ack* packet within a timeout period (2 seconds), it retransmits a *Hello* packet. If the sender still does not receive the *Hello_Ack* packet, then the transmission is failed.
    - Dump a snapshot of events (i.e., a packet loss probability is 0.3). Only one-page would be enough. For example,

```
...
node.0 sends a Hello to node.3 at 100.2 seconds.
node.2 replies a Hello_Ack to node.1 at 100.5 seconds
...
node.4 sends a Hello to node.3 at 110.5 seconds
node.1 receives a Hello_Ack from node.2 at 110.6 seconds
node.4 re-sends a Hello to node.3 at 112.5 seconds
...
```

    - The simulation ends when the simulation time reaches 1000 seconds.
    - Draw two result graphs in terms of following performance metrics against the packet loss probabilities (0.1, 0.2, 0.3, 0.4, or 0.5).

```
Average number of successful transmissions
Average number of failed transmissions
```

2. Type the project number and your name (Last, First) at the top in your source code.

- `/* Project #2; Bond, James */`

3. Refer to CSIM manual uploaded in the class homepage:
   - CSIM20 Quick Start Guide for C
   - CSIM20 User's Guide for C
   - http://www.mesquite.com/documentation/

4. Compilation & Run:
   - `csim64.gcc proj2_bond_james.c -o proj2_bond_james`
   - `./proj2_bond_james`

5. What you turn-in?
   - Source code(s)
     - Your source code name should be "project number" + "last name" + "first name", e.g., "`pro2_bond_james.c`".
   - Report (e.g., MS word file) including,
     - Instruction how to run your program.
     - Two result graphs
   - The grader will compile and run your source code and double check your result graphs.