

#### To compile

1. Navigate to Project Directory
2. To Build Java Files
  - a. Enter “ant” in command line
3. To Run
  - a. Enter “python Project\_2\_GenerateInput.py [number of nodes]”
    - i. This will generate input files for the simulation.
  - b. Enter “java -jar wfong\_p2.jar [number of nodes]”
    - i. Will accept any number between 2 and 254.

#### List of Files

- bin file directory
  - Contains the binary for the Java Code.
- src file directory
  - Contains the source code for the project.
- README.md
  - A readme file for the git-repository for the project.
- Grade.py
  - A supplied python script for checking the output of the program.
- Project\_2\_GenerateInput.py
  - A supplied python script for checking the output of the program.
- test.sh
  - A bash script for running multiple iterations of the Project to compare THT sizes against simulation completion time.
- build.xml
  - The XML build file for the project.
- wfong\_p2.jar
  - The project as a runnable jar file (created when build.xml is ran).

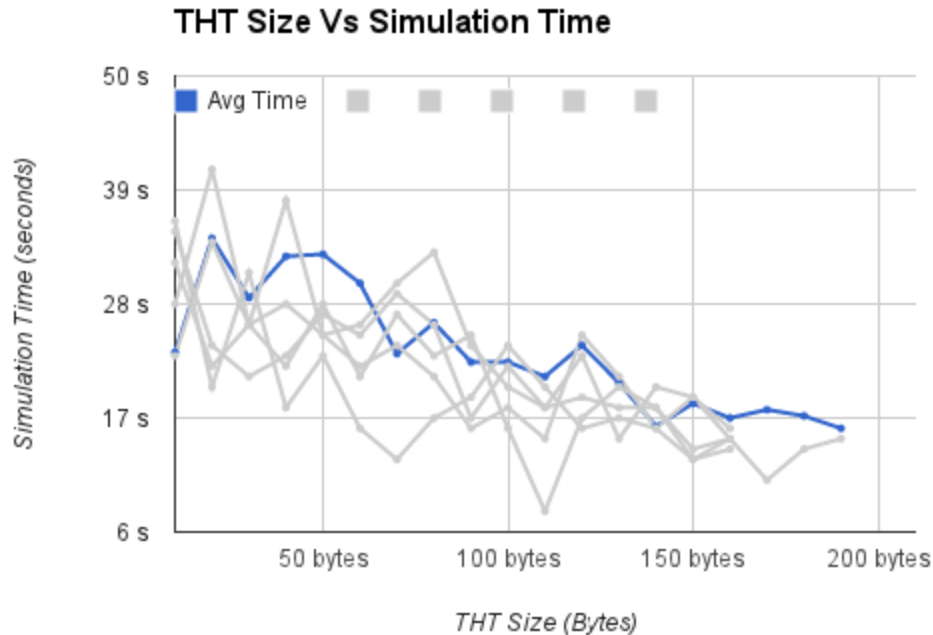
#### Checklist

Item	Status/Description
Load Input Item	Complete
Parse Input Item	Complete
Generate Token	Complete
Generate Frame	Complete
Node ran in separate thread	Complete
Project can be ran in Terminal with args	Complete
Randomly Reject Frame (20%)	Complete

Extract all fields from the frame	Complete
Switch between transit and receive states	Complete
Pass tokens and frames to neighbor	Complete
Print to output-file-n	Complete
Documentation	Complete
Executable	Complete
Monitor detects lost token	Complete
Monitor cleans up garbled frame	Missing
Monitor cleans up orphan frame	Complete
No frames are lost	Complete
<b>EC:</b> Frames are transmitted in as a Binary Frame	Complete: Frame class can encode input file lines to a byte array to transmit. It also contains methods to decode incoming frames into readable data.
<b>EC:</b> Network shuts down using interprocess communication	Complete: Used unused bytes in Frame Status to send kill signal to network

#### Token holding time

To choose what Token Holding time to use I simulated various THT sizes and compared them to the simulation time. This process took into account the 20% frame rejection rate, the 5% token loss rate, and the 2% orphan generation rate. There were a total of 5 trial runs that ran simulations of 10 to 200 byte THT at 10 byte increments. This simulation had a total of 10 nodes. Larger networks would have yielded more usable data; however my computer could not do the task in a reasonable time frame and the TCC administrators weren't too keen on me hogging computer resources. So the resulting 'optimal' THT holding time is only applicable for networks with 10 nodes. I do not know if it maintains this trend for larger networks (probably not).



THT sizes ranging around the 50 byte mark yield the longest transmission time. With this low size, usually only a few frames can be transmitted at a time. The transmission time seems to level off beyond the 150 byte mark, however this is an assumption my data ends there as getting averages for them would take too long. But around this mark, the eccentricity of the data is low in comparison to smaller THT sizes. Based upon this data, the simulation, by default, runs with a 150 byte THT.

### Garbled Frames

I was unable to compensate for the garbled frames. To simulate them I randomly omitted one of the frame format bytes. The monitor node was able to detect these garbled frames. The regular nodes however could not. Once they received a garbled frame, if the data size was incorrect, had no way of knowing where the next frame began. On the next read instruction, the node may get stuck and attempt to read a very large data size which will never arrive. There is a check to see if the last byte is a valid frame status byte, however once a frame incorrectly transmitted a frame, it could not differentiate between a frame format sequence and the actual data.

### Bugs

Garbled frames were simulated, but nodes easily lost track (after transmitted a garbled frame) of where legitimate nodes began.

William Fong

CSE 353

Project 2

11/4/14

Credit

StackOverflow for a socket timeout method:

<http://stackoverflow.com/questions/7360520/connectiontimeout-versus-sockettimeout>

To Skyler Manzanares for helping me with a problem where the networked dropped a single frame very rarely.