CPR E / SE 492 BIWEEKLY STATUS REPORT 2

February 9 - February 22

Senior Design Team 15

Debugger and Visualizer for a Shared Sense of Time
on Batteryless Sensor Networks

Client/Advisor
Dr. Henry Duwe

Team Members
Adam Ford - Report Manager
Allan Juarez - Scribe
Maksym Nakonechnyy - Design Lead
Anthony Rosenhamer - Facilitator
Quentin Urbanowicz - Test Engineer
Riley Thoma - Project Manager

Biweekly Summary

We have been working on building off of our progress from the last two-week period and integrating our client feedback as we continue to build the simulator and visualizer. On the simulator side, we expanded the event handling and made it more configurable. We also began exploring the interfacing options that our clients will be using to interact with the tool. The frontend team decided on a graphing library, GoJS, and worked to implement it into our demo code from last semester. The backend team continued exploration and began writing code for the actual project use cases.

Accomplishments from the Past Two Weeks

- Backend Team (Adam and Allan)
    - Pushed ExpressJS code further to start reflecting an actual needed function for persisting simulator data.
    - Completed the Socket connection to the simulator.
- Frontend Team (Maksym and Riley)
    - Rewrote the project to use a new graphing library.
    - Changed the styling of sensor nodes.
    - Added a real time panel.
    - Refactored the code.
- Simulator Team (Anthony and Quentin)
    - Improved event message formatting by removing delimiter bytes from data fields
    - Implemented an interface for configurable sensor node event handling capable of:
        - Logging to a trace file
        - Sending messages to the backend
        - Printing plain-text event messages

Pending Issues

We need to check with our client to see if the format for our trace file is appropriate and contains all the relevant information about sensor node events.

We need to figure out if the nodes being able to communicate with one another is visualized on the frontend or if it’s just displayed during communication.

What types of graphs need to be displayed on the visualizer?

Individual Contributions

Name	Individual Contributions	New Hours (last 2 wks)	Total Hours
Adam Ford	Further explored the ExpressJS connections to MongoDB, wrote a sample function to iterate on as the storage function. Starting writing the db setup script, currently incomplete.	12	25
Allan Juarez	Completed the connection via websocket from backend to frontend. Also checked up the rest api from frontend to backend and made sure it was working.	12	24.5
Maksym Nakonechnyy	Examined GoJS documentation. Experimented with a sample GoJS project. Rewrote our last semester's demo to use GoJS. Refactored the code. Started implementing network configuration file support.	14	26
Anthony Rosenhamer	Implemented event handling for the simulator events, including logging to a trace file, sending data to the backend via the sockets, and human-readable logging Improved message formatting to reduce size of trace files and event messages	12	25
Quentin Urbanowicz	Continued exploring ways to introduce pseudorandom variability into node behavior. Continued work on abstracting node lifecycle logic and refactoring changes into existing code.	12	24
Riley Thoma	Created an experimental project for testing GoJS graphing library. Also tried out ReactFlow. Learned how to use the zoom functions in GoJS, worked on styling of nodes and time slider paired with the real time clock.	12	24

Plans for the Next Two Weeks

- Adam Ford - backend development
    - Complete database setup script, ideally as a part of Express, otherwise ask the team/advisor for alternative ideas
    - Trace file consumption
    - Work on tree generation based on trace file
- Allan Juarez - backend development
    - Will work on taking in the proper data that the simulator sends and format it
    - Work on tree generation based on trace files.
- Maksym Nakonechnyy - frontend development
    - Finish implementing configuration file support.
    - Start implementing a node communication history panel.
    - Look into libraries to display network statistics.
- Anthony Rosenhamer - simulator development
    - Continue to develop the interface for adding node communication protocols
    - Adjust event message formatting to make it configurable
- Quentin Urbanowicz - simulator development
    - Continue refactoring existing node simulation code to be more abstracted in order to accommodate easier changes
    - Implement a framework for providing pseudo-random variation in node lifecycle behaviors (local and simulation-wide)
- Riley Thoma - frontend development
    - Finish developing slider to real time clock component.

Summary of Advisor Meetings

February 12th:

We discussed the time synchronization algorithm, the sniffer node on the simulator, and details about the interface that our clients will use with the tools. We also discussed some of the basic communication protocols that we will begin testing our system with.