CPR E / SE 492 BIWEEKLY STATUS REPORT 5

March 15 - March 29

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

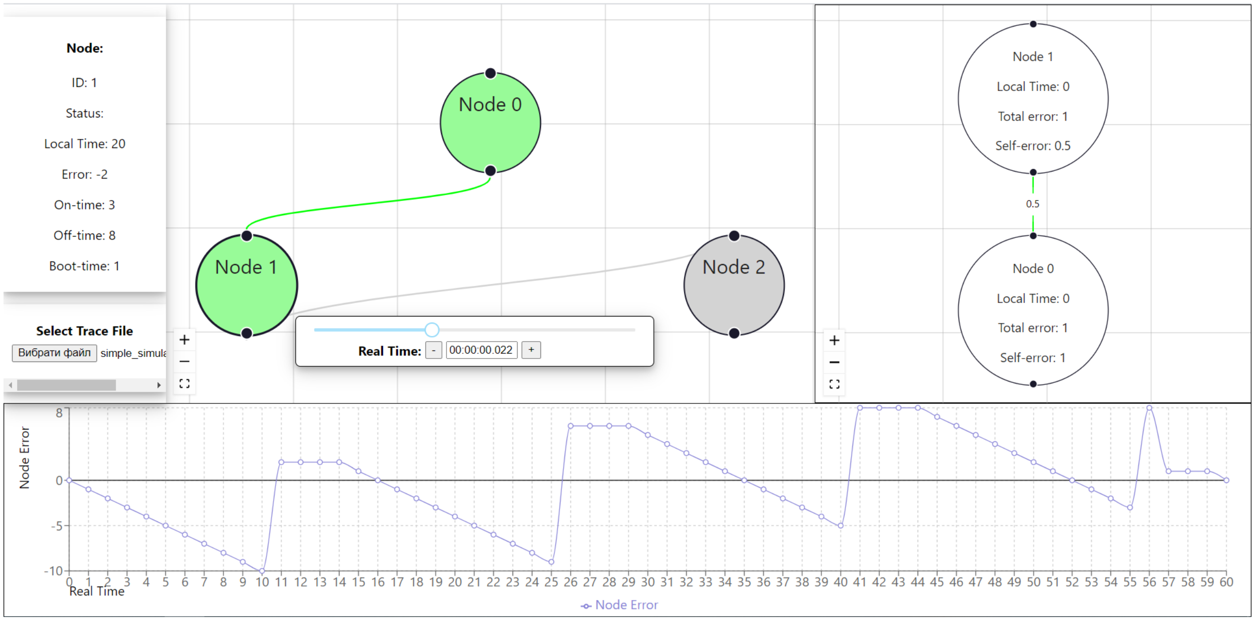
Biweekly Summary

Our team has continued to integrate the entire system. On the simulator side, we have expanded the energy model for nodes in the system. We have also been working on providing more abstraction for interfacing with the simulator (e.g. adding communication protocols for research). The backend team has pushed deeper into making sense of the simulator events to create a summary of the system at a given time to the frontend. Some event types, but not all have been used to update the system status. On the frontend side, we continued work on the Error Graph to make it display actual simulation results with more details. We also updated the general system UI, updated the Network Graph UI, implemented the Error Tree component, and refactored the code.

Accomplishments from the Past Two Weeks

- Backend Team (Adam and Allan)
  - Parsed simulator events are put through a processing function to update the current system status.
  - System statuses are stored and fetched from the database where they are constantly updated with each new event (irrelevant with trace files, but will be necessary for any live functionality).
  - Some simulator events meanings are used, some are yet to be used for their full meaning.
- Frontend Team (Maksym and Riley)
  - The “Error Tree” component is now functioning and displays the history of communications.
  - Updated the Network Graph functionality:
    - Reachability edges (which nodes can communicate with one another) are displayed.
    - Communication status (successful/unsuccessful) displayed.
  - Refactored the code that handles network communication with the backend application.
  - Refactored the code that updates the UI.
  - Restyled the frontend UI panel sizings
  - The Error Graph has demo data and styling is completed for future data

Updated UI:



Node:

ID: 1

Status:

Local Time: 20

Error: -2

On-time: 3

Off-time: 8

Boot-time: 1

Select Trace File

Вибрати файл simple_simula

Real Time: - 00:00:00.022 +

Node 1

Local Time: 0

Total error: 1

Self-error: 0.5

Node 0

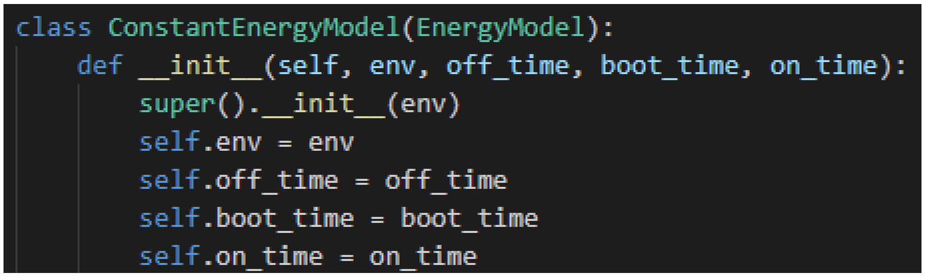
Local Time: 0

Total error: 1

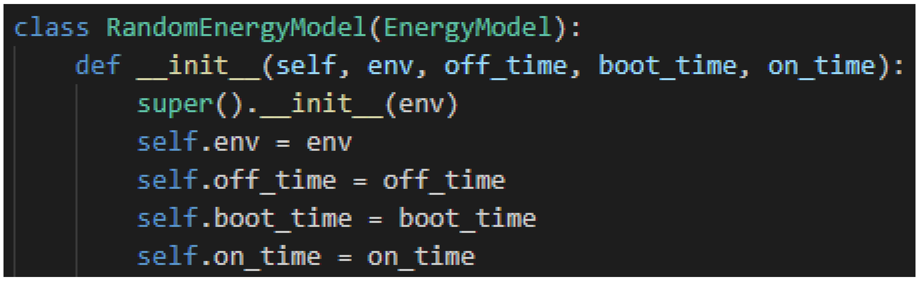
Self-error: 1

- Simulator Team (Anthony and Quentin)
    - Implemented the energy model
        - Added a model with constant behavior
        - Added a model with random variation
    - Abstracted the node interface
        - Broke up functionality within the node into compact functions
        - Expanded node communication interfacing

  Energy Models in the Simulator Code:



```python
class ConstantEnergyModel(EnergyModel):
    def __init__(self, env, off_time, boot_time, on_time):
        super().__init__(env)
        self.env = env
        self.off_time = off_time
        self.boot_time = boot_time
        self.on_time = on_time
```



```python
class RandomEnergyModel(EnergyModel):
    def __init__(self, env, off_time, boot_time, on_time):
        super().__init__(env)
        self.env = env
        self.off_time = off_time
        self.boot_time = boot_time
        self.on_time = on_time
```

Pending Issues

- We need to discuss with our client to see how to handle node communication. We have to check when we should show communication as successful vs failed and how to handle the time delay between a message and its reply.

Individual Contributions

Name	Individual Contributions	New Hours	Total Hours
Adam Ford	Created function framework for saving and fetching system statuses from database synchronously. Retrieved initial meaning out of some events.	14	58
Allan Juarez	Created a V3 parsing to grab the new items in the trace files. Also completed the parsing and formatting for the configurations of the simulator.	13	57.5
Maksym Nakonechnyy	Implemented the error tree component. Updated the network graph to display reachability edges and communication status. Implemented a class to handle all network communication. Added an event bus to update the UI after receiving a response from the backend.	15	60
Anthony Rosenhamer	Expanded node communication interfacing Reworked node behavior logic Implemented logic for basic energy models	13	58
Quentin Urbanowicz	Created an abstract energy model for use in abstracting node energy and state behavior logic. Revised node class API to better accommodate modularization.	12	54
Riley Thoma	Modified the frontend styling to give more room for the network graph. Created a more accurate Error graph using constant data from the network demo Styled the Error Graph to have it ready for future data	8	48

Plans for the Next Two Weeks

- Adam Ford - backend development
    - Complete system status objects and create an endpoint to serve them to the frontend
    - Attempt integration with frontend
    - Begin working on error graph endpoint
        - Create error graph from node status in system
        - Serve it via endpoint
- Allan Juarez - backend development
    - Work with adam and see if he needs help with completing the system objects
    - Work on creating endpoints so frontend can call and receive some events from the trace files.
- Maksym Nakonechnyy - frontend development
    - Figure out how to graph the error tree more nicely (node coordinates).
    - Refactor the code to limit components' access to unnecessary information.
    - Look into how to have multiple tabs with different simulations open at the same time.
- Anthony Rosenhamer - simulator development
    - Continue to expand the inputting for different communication interfaces
    - Add failed communication event
    - Document the simulator code and its usage for our client
- Quentin Urbanowicz - simulator development
    - Continue refining node API based on client feedback
    - Finish working on implementing time delay and random variation
- Riley Thoma - frontend development
    - Work on the Error Graph to add mean and median lines
    - Add a vertical line to track the global time for the simulation
    - Implement requests for error graph lines
    - Create input simulation trace file start page

Summary of Advisor Meetings

3/19/2021 - Meeting with Dr. Duwe and Vishal
We walked through the updated frontend. We showed our clients the new views added, including a node error tree and an error graph. Dr. Duwe made several requests for the Error Graph like a Median and Mean line and also for a separate window for the user to add a trace file. We also showed them the new network structure view that uses an open source library as they requested.