ASEE AMERICAN SOCIETY FOR ENGINEERING EDUCATION

Paper ID #41170

Design of Simulator Test Interfaces for Wireless Sensor Networks

Hunter Dawson Yaple, Gannon University
Dr. Ramakrishnan Sundaram, Gannon University

Dr. Sundaram is a Professor in the Electrical and Computer Engineering Department at Gannon University. His areas of research include computational architectures for signal and image processing as well as novel methods to improve/enhance engineering educa

Charles julius Maier, Gannon University

Work-in-Progress: Design of Simulator Test Interfaces for Wireless Sensor Networks

Hunter Yaple, Charles Maier, Dr. Ramakrishnan Sundaram
Department of Electrical and Cyber Engineering
Gannon University, Erie, PA 16541
yaple007@gannon.edu, maier002@gannon.edu, sundaram001@gannon.edu

Abstract
This paper describes the setup of the interface between the physical nodes of the wireless sensor network and the virtual sensors in the simulation environment. The wireless sensor network comprises WiFi modules, configured in a grid, to transmit and receive radio frequency signal data. The interface between the actual network and the simulator enables the signal data from the actual wireless sensor network to be assigned to virtual sensor nodes in the simulation environment. In this manner, the actual network is replicated on the simulator. The creation of the virtual sensor network facilitates the virtual modeling of the actual sensor grid. Test interfaces are designed to compare actual and virtual network performance. CupCarbon is the wireless sensor network design and simulation tool used to interface with the WiFi modules used in the actual wireless sensor network. In addition to displaying key information from the actual network, one of the functions of the virtual sensor network is to compute the change in the received signal strength at each node. The grid configuration, both actual and simulated, yields received signal strengths arranged as a matrix of values from each transmitter to each receiver. Each matrix constitutes a frame with a specific time stamp. The change in the received signal strengths at each receiver of the network is observed in the actual network and processed in the virtual environment. The change, an attenuation or an accentuation, is used to detect the obstruction within the network by solving the inverse problem. One of the virtual models will assign binary weights to the cells in the space occupied by the grid and solves for the shade of gray to be assigned to each cell. It is expected that as the obstruction passes through the real sensor network, the virtual sensor network determines the location and shape of the obstruction. The platform is being developed to test and model wireless sensor network grids at different geometric scales and with distinct spatial topologies. Its effectiveness to deliver STEM components across K-12 and advanced degree programs is summarized. Although platforms based on wireless sensors such as the TelosB modules have been designed, these cannot be easily integrated with the virtual environment for modeling and simulation studies. The platform is cost effective (approximately $200 for a 10x10 grid) and the CupCarbon software is accessible as a free download.

Section 1: Introduction

Rapid technological advances are constantly impacting the engineering landscape. Consequently, educational institutions must restructure the content and delivery of engineering courses to prepare the student for the vagaries of the work environment. Project-based activities which instill self-driven and just-in-time learning experiences[1]-[3] are incorporated to promote system-level thinking and reinforce the technical, communication, and leadership skills of each student. Engineering project platforms[4] offer an effective approach to integrate the curriculum across courses and academic level of the student. This paper provides the framework of an engineering project which integrates the hardware environment with the simulation environment. Each

environment can function independently and involve different aspects of system design and development. The integrated environment focuses on the design and implementation of interfaces which couple the operation of the hardware environment with the simulation environment.

The hardware environment comprises wireless sensor networks (WSN) which are configured to surveil and monitor entities/obstructions within the network. The sensors comprise radio frequency (RF) and/or infra-red (IR) modules which transmit and receive signals (RF or IR) within the confines of the network. The attenuation of the received signal strength (RSS) is used to detect the presence of and track the motion of entities/obstructions within the network [5]-[11]. The simulation environment comprises wireless sensors configured in the CupCarbon simulator [12]. The sensors are arranged in scaled topologies like those used in the hardware environment. The sensors are programmed to (a) function in stand-alone or virtual mode (b) respond to the corresponding sensor in the hardware environment.

Section 2 identifies the hardware environment. Section 3 describes the simulator environment. Section 4 discusses the integration of the hardware and software environments. Section 5 outlines conclusions and future work.

Section 2: Hardware environment

Figure 1 illustrates one of the WSN configurations on a rectangular grid comprising five transmitting nodes and five receiving nodes. Each configuration comprises sensor nodes mounted on 3-D printed stands. The sensor nodes are Wi-Fi modules[13] capable of transmitting and receiving RF signals. The Wi-Fi router interfaces the WSN to the user interface (UI) on the desktop or laptop. The interface between the UI and the WSN captures the RSS data for display and analysis[14],[15].

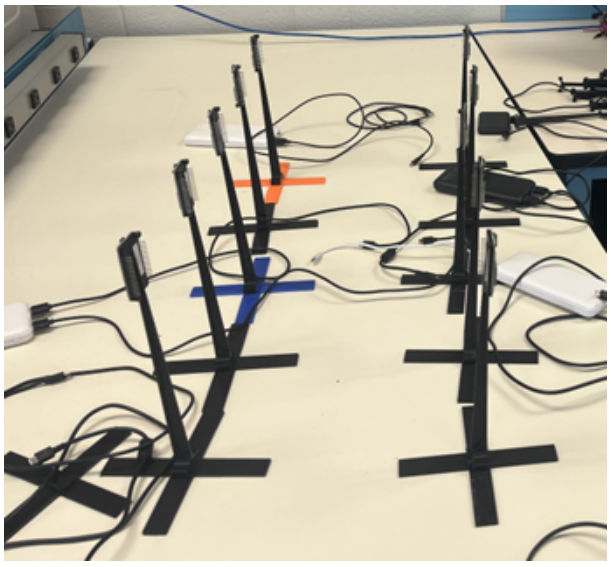


Figure 1: WSN – rectangular grid

Figure 2 shows the screenshot of the RSS link values captured by the UI. The RSS link values are recorded in decibel units (dB). Table 1 displays the record of RSS link values in Excel.

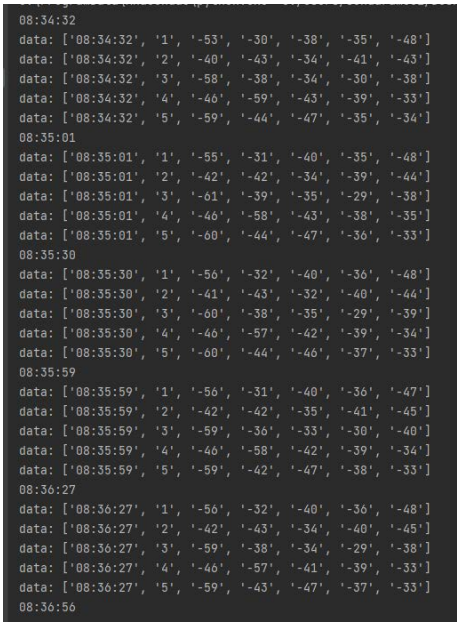


Figure 2: Snapshots of RSS data as seen on the UI

Table 1 displays the record of RSS link values in Excel.

Table 1: Table of RSS values in Excel

EXCEL TABLE OF RSS DATA: CALIBRATION PHASE: 4/14/22						
8:34:32	1	-53	-30	-38	-35	-48
8:34:32	2	-40	-43	-34	-41	-43
8:34:32	3	-58	-38	-34	-30	-38
8:34:32	4	-46	-59	-43	-39	-33
8:34:32	5	-59	-44	-47	-35	-34
8:35:01	1	-55	-31	-40	-35	-48
8:35:01	2	-42	-42	-34	-39	-44
8:35:01	3	-61	-39	-35	-29	-38
8:35:01	4	-46	-58	-43	-38	-35
8:35:01	5	-60	-44	-47	-36	-33
8:35:30	1	-56	-32	-40	-36	-48
8:35:30	2	-41	-43	-32	-40	-44
8:35:30	3	-60	-38	-35	-29	-39
8:35:30	4	-46	-57	-42	-39	-34
8:35:30	5	-60	-44	-46	-37	-33
8:35:59	1	-56	-31	-40	-36	-47
8:35:59	2	-42	-42	-35	-41	-45
8:35:59	3	-59	-36	-33	-30	-40
8:35:59	4	-46	-58	-42	-39	-34
8:35:59	5	-59	-42	-47	-38	-33
8:36:27	1	-56	-32	-40	-36	-48
8:36:27	2	-42	-43	-34	-40	-45
8:36:27	3	-59	-38	-34	-29	-38
8:36:27	4	-46	-57	-41	-39	-33
8:36:27	5	-59	-43	-47	-37	-33

Section 3: Simulator environment

Figure 3 displays the simulator environment in CupCarbon. The wireless sensor network comprising virtual sensor nodes are configured as devices in software with the appropriate sensor parameters.

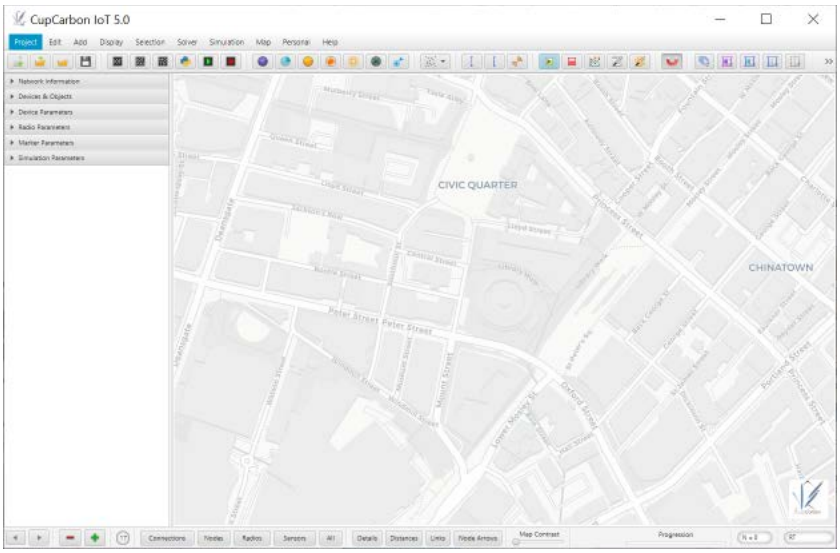


Figure 3: Simulator environment in CupCarbon

For example, the virtual sensor grid comprising five transmitters and five receivers in a rectangular arrangement is seen in Figure 4. The data, alongside each red line, is the transmitted value for each link as a test case prior to RSS simulation and integration with the hardware environment.

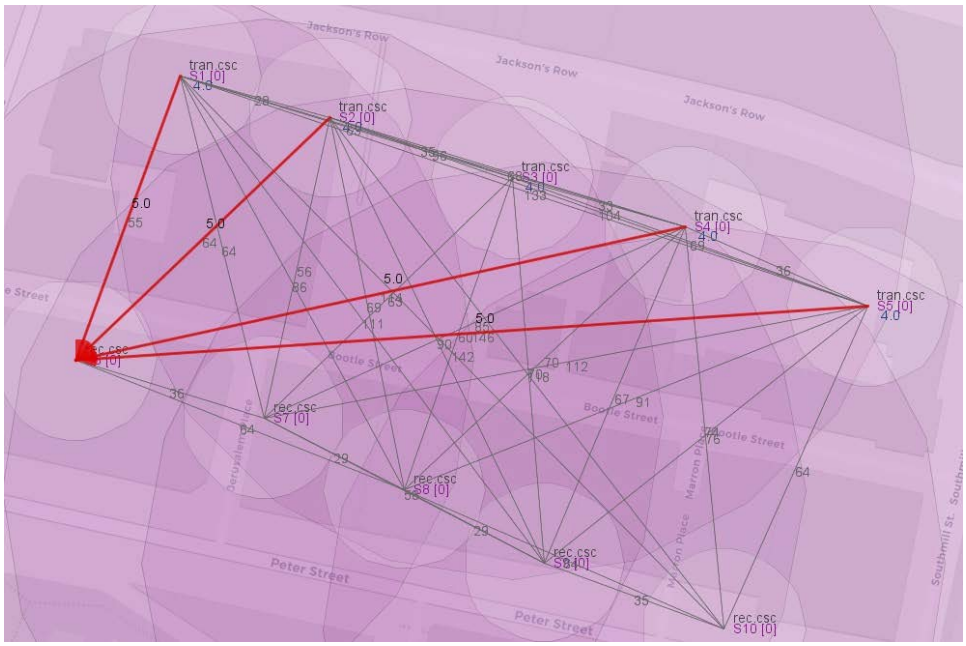


Figure 4: Rectangular grid (5x5) of virtual sensors in CupCarbon

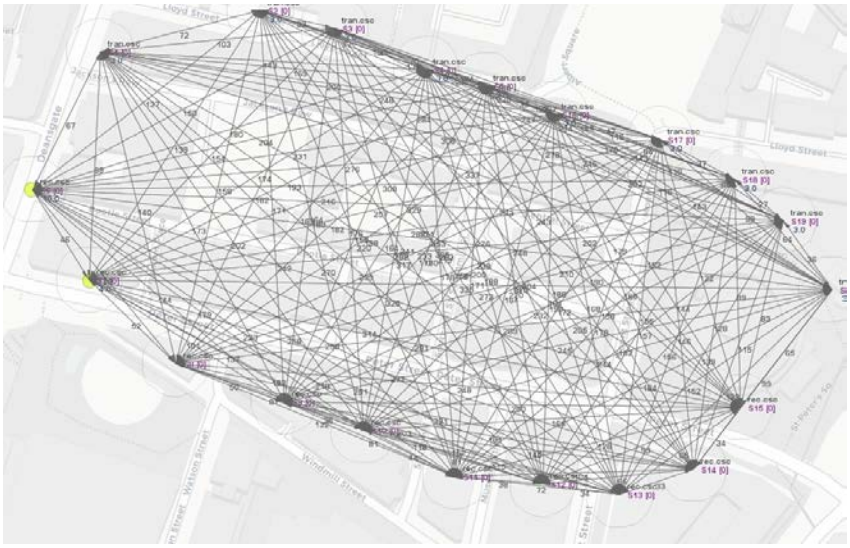
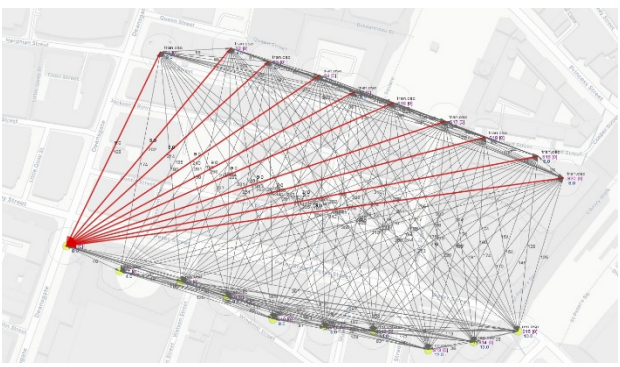
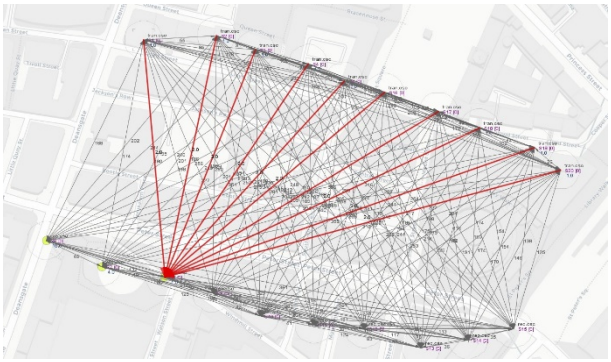


Figure 5: Elliptic grid (10x10) of virtual sensors in CupCarbon

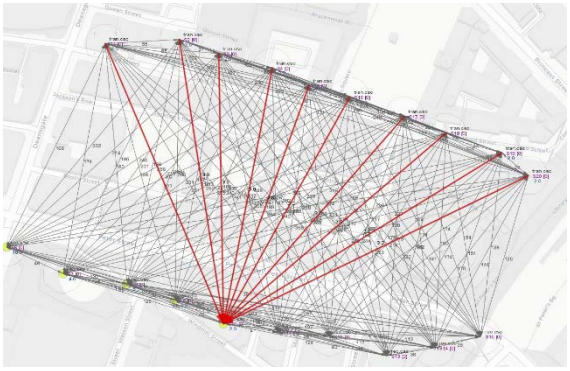
Figure 5 extends the grid to have ten transmitting sensors and ten receiving sensors in an elliptic configuration. Figure 6 displays four instances of data transmission, indicated by the red lines, from each transmitter to a single receiver. The data, alongside each red line, is a random number used to test each link prior to RSS simulation and integration with the hardware environment.



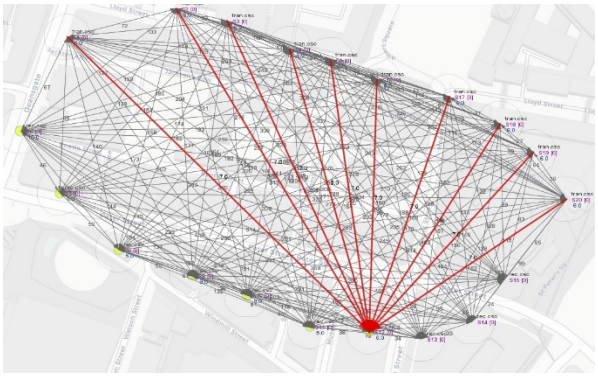
(a)



(b)



(c)



(d)

Figure 6: Data transmission to (a) receiver #1 (b) receiver #3 (c) receiver # 5 (d) receiver #7

Figure 7 displays the data vector at each receiver of the 5x5 sensor grid after all transmitters have communicated with each receiver.

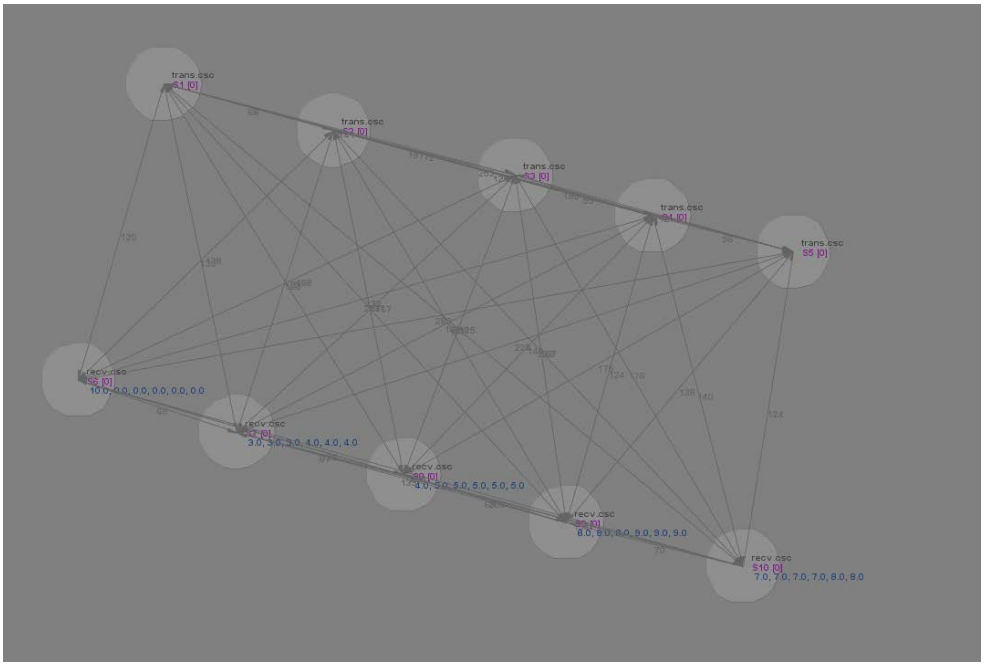


Figure 7: Data vector at each receiver

Figure 8 shows the result of data vectors received at eight of the ten receivers in the 10x10 sensor grid during the simulation. The red lines indicate the transmissions to the eighth receiver.

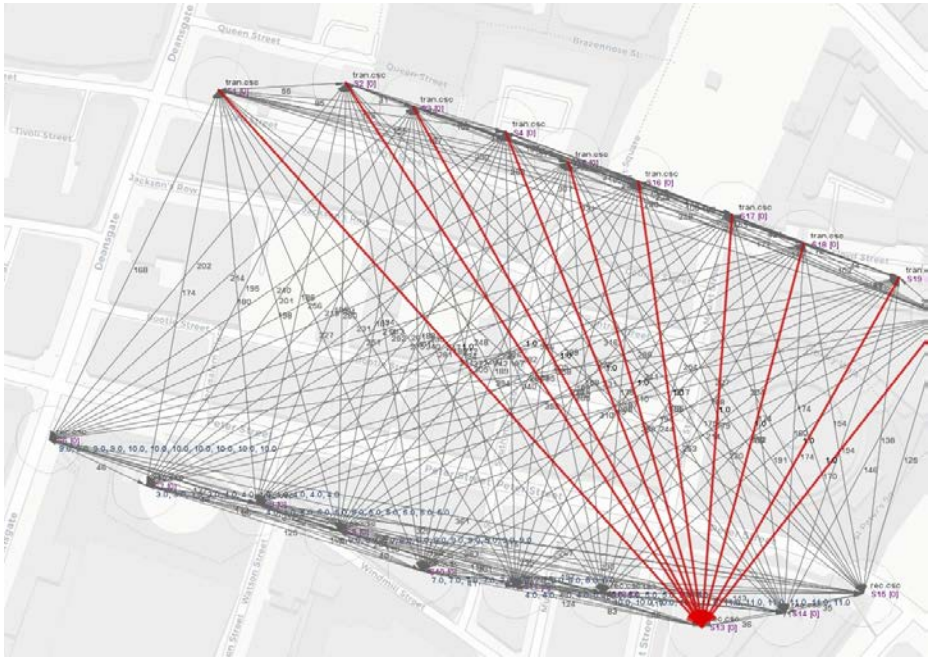


Figure 8: Data vector at each receiver during simulation

Section 4: Integrated platform – set up and test

The interface between the ESP32 WiFi module uses as the wireless sensor node and the CupCarbon simulator is created using the following stepwise procedure. Local administrative privileges on the computing platform (laptop or workstation) are required to successfully complete the steps.

1. To download CupCarbon V6 go to this site and input the necessary info for download: https://cupcarbon.com/download.php
2. Install python in the Program Files directory of the C: drive using this site: https://www.python.org/downloads/

Ensure, going forward, that there are no spaces in the file path as Python sometimes fails to understand spaces as given by CupCarbon.

3. Open CupCarbon V6 and start a new project, ensuring a clean file path as detailed above.
4. Create wireless sensor nodes as necessary for the project. Review CupCarbon SenScript syntax to understand which functions are needed to realize your design.
5. Install the pyserial, numpy, and scipy python libraries using the command line. For the pyserial library used below navigate to the directory of the python executable and run the command 'pip install pyserial'.
6. Using an external code/text editor, create .py scripts and save them in the 'scripts' folder of your CupCarbon project. The internal editor in CupCarbon does not follow the spacing requirements needed for most python scripts. Follow the COM port reading script below for syntax, noting that CupCarbon executes the .py script externally and takes the printouts of that script as input commands for the node in CupCarbon.

```
import time
import serial
ser = serial.Serial('COM5',9600,serial.EIGHTBITS,timeout=5)
while True:
        data =ser.readline()
        full=data.decode()
        val=int(full)
        if val == 1:
                print("print 1", flush=True)
                print("mark", flush=True)
        else:
                print("print 0", flush=True)
        print("unmark", flush=True)
```

7. Test the code in Python to ensure the script runs before proceeding to CupCarbon. Note that the outputs are in the proper SenScript syntax to be delivered back to intended CupCarbon node.
8. In CupCarbon, assign the script to the intended wireless sensor node.

9. In the CupCarbon interface (Figure 9), set the executor path to ‘py’ or ‘python’ based off the Python configuration you have downloaded so that the program will execute .py scripts using the Python program.

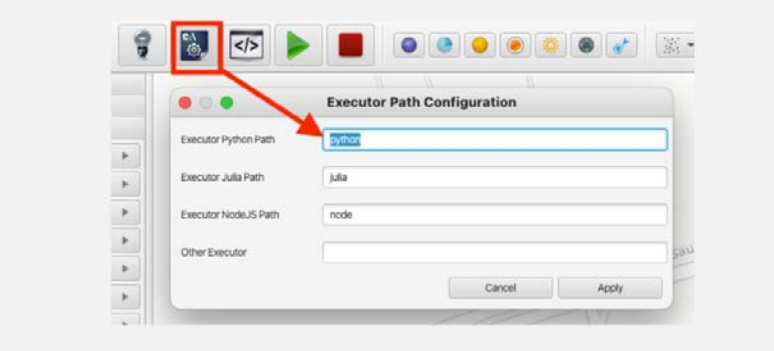


Figure 9: CupCarbon interface setting

10. Run the simulation using the ‘Run IoT Simulation’ (green block/arrow). If successful, there will be a red outline around the simulation window.

Figure 10 displays the test case of the integrated set up with a single transmitter-receiver pair and an obstruction (water bottle) interfaced to the CupCarbon environment.

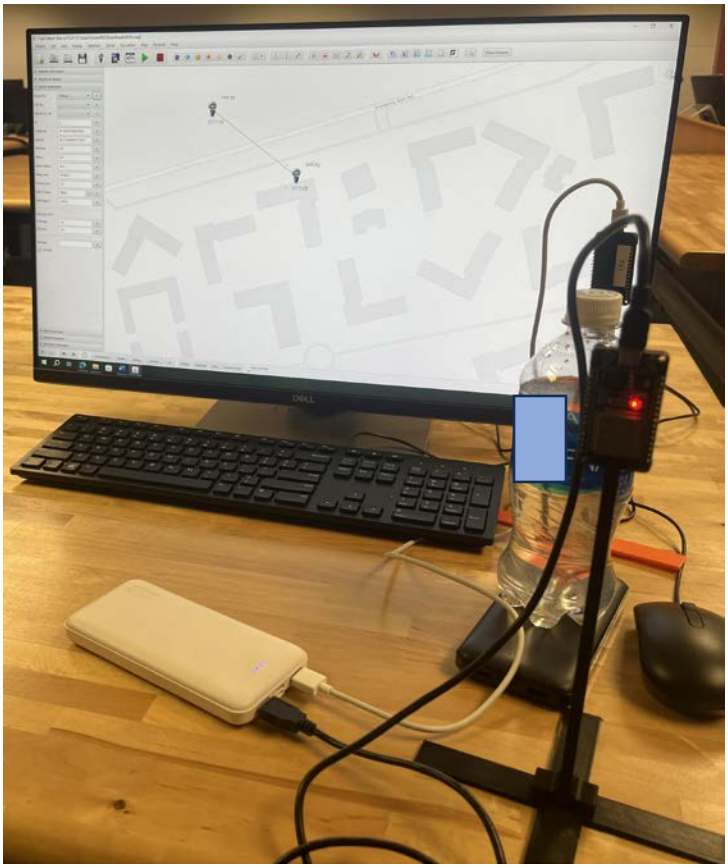


Figure 10: Integrated system – test case

Figure 11(a) illustrates the outcomes of the test case using the integrated platform without an obstruction between the transmitter and receiver. The virtual sensor node in CupCarbon is an IoT node and is identified by the light bulb. This is the single available CupCarbon device with the capability to interface with the communication port. The time-stamped RSS link value of -56 dB as obtained from the hardware set up is displayed at the receiver node. This value attenuates to -73 dB when the obstruction is introduced in the space between the two nodes (Figure 10) as shown in Figure 11(b).

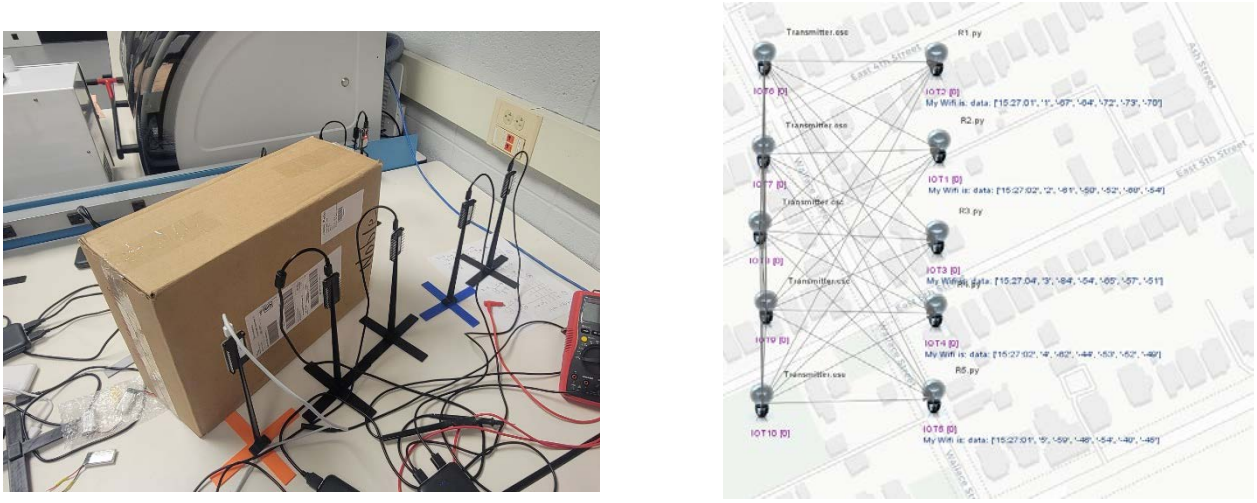
(a) (b)

Figure 11: Time-stamped RSS value (a) no obstruction (b) with obstruction

Having established the proof of principle, the next step is extending the configuration to larger grid sizes in both environments and recording the RSS values at each virtual receiver node from all the transmitters. The platform is expected to be fully functional in the summer of 2024 and will be incorporated in courses on wireless communications starting the fall of 2024.

Preliminary results with 5x5 grid

Figure 12 illustrates the extension of the test case to the 5x5 grid of actual and virtual nodes. Figure 12(a) is the set up in the hardware environment with the obstruction within the WSN. Figure 12(b) shows the capture of the timestamped RSS data at each virtual receiver in the simulator environment.



(a) (b)

Figure 12: (a) Hardware setup with obstruction (b) Time-stamped RSS data in simulator

Associated learning outcomes for undergraduate ECE students
The undergraduate ECE students are expected to design and test the hardware environment, configure the standalone simulator environment, and finally, integrate the two environments. The following ABET student outcomes are mapped to laboratory activities based on the integrated WSN.

abet_SO_1: Ability to identify, formulate, and solve complex engineering problems by applying principles of engineering, science, and mathematics

abet_SO_2: Ability to apply engineering design to produce solutions that meet specified needs with consideration of public health, safety, and welfare, as well as global, cultural, social, environmental, and economic factors

abet_SO_6: Ability to develop and conduct appropriate experimentation, analyze, and interpret data, and use engineering judgment to draw conclusions

STEM activities for K-12 students
Typical application of the platform to deliver STEM activities for K-12 students are based on the hardware environment as follows:
Middle school students: assemble and test the hardware platform
High school students: record and interpret data using the hardware platform

Section 5: Conclusions and Future Work

The proposed engineering research platform, comprising wireless sensor nodes in the hardware environment integrated with the virtual wireless sensor nodes in the simulator environment, promotes the participation and advancement of faculty and student research, curriculum development through courses, laboratory experiments, course projects, as well as STEM outreach with pK-12 schools. The hardware and software requirements of the platform are inexpensive. The setup is portable as well as easily scaled up to grids of larger dimensions and topologies. The software platform of CupCarbon is free and easy to install on personal computing stations. At present, the platform is in the testing phase to ensure reliable operation of the configuration in each environment and across both environments.

Work is ongoing to study the effect of regularization on the recovery of the image representing the entity or obstruction within the network. This requires incorporating the results obtained using the data from the hardware environment [10], [11] to create a model for the virtual environment which studies and tracks entities within the wireless network in the hardware environment (real world) to capture and display the information in the simulator environment (virtual world).

Bibliography

[1] R. Sundaram, "Engineering Project Platform for Electrical and Computer Engineering Curriculum Integration," *Proceedings of the 2014 Annual ASEE conference & Exposition, Indianapolis, IN*, June 15-18, 2014. pp. 24.503.1-24.503.14, ISSN: 2153-5965; DOI: 10.18260/1-2-20394.

[2] D. Damian *et al*., "Teaching a globally distributed project course using Scrum practices," Collaborative Teaching of Globally Distributed Software Development Workshop (CTGDSD), Zurich, pp. 30-34, June 2012.

[3] D.F. Rico and H.H. Sayani, "Use of Agile Methods in Software Engineering Education," *Proceedings of the Agile Conference* (*AGILE '09*), pp. 174-179, August 2009.

[4] V. Mahnic, "A Capstone Course on Agile Software Development Using Scrum," *IEEE Transactions on Education*, Vol. 55, Issue: 1, pp. 99-106, February 2012.

[5] B. R. Hamilton, X. Ma, R. J. Baxley, and S. M. Matechik, "Radio Frequency Tomography in Mobile Networks, in *Proc. IEEE Workshop on Statistical Signal Processing*, Ann Arbor, MI, Aug. 2012.

[6] R.P.S. Inglis, R.P. Brenner, E.L. Puzo, T.O. Walker, III, C.R. Anderson, R.W. Thomas, and R.K. Martin, "A Secure Wireless Network for Roadside Surveillance using Radio Tomographic Imaging," *Proc. 6th Int'l. Conf. Signal Processing and Comm. Systems*, Gold Coast, Australia, Dec. 2012.

[7] C.R. Anderson et al., "Radio Tomography for Roadside Surveillance," *IEEE Journal of Selected Topics in Signal Processing*, Volume: 8, Issue: 1, pp. 66-79, Feb. 2014; DOI: 10.1109/JSTSP.2013.2286774.

[8] O. Kaltiokallio *et al.*, "A Fade Level-Based Spatial Model for Radio Tomographic Imaging," *IEEE Transactions on Mobile Computing*, Volume: 13, Issue: 6, pp. 1159-1172, June 2014. doi: 10.1109/TMC.2013.158.

[9] L. Heng *et al*., "Image reconstruction algorithms for radio tomographic imaging," *Proc. of the 2012 IEEE International Conference on Cyber Technology in Automation, Control, and Intelligent Systems* (*CYBER*), 27-31 May 2012; DOI: 10.1109/CYBER.2012.6392525.

[10] R. Sundaram *et al*., "Regularization in Radio Tomographic Imaging," *Proc. SPIE 8753, Wireless Sensing, Localization, and Processing VIII*, 87530O: DOI:10.1117/12.201216.

[11] T. Batjargal & R. Sundaram, "Tomographic imaging with wireless sensor networks," *Proc. SPIE 10752, Applications of Digital Image Processing XLI*, 1075202 (17 September 2018); doi: 10.1117/12.2314860.

[12] Website: https://cupcarbon.com.

[13] Website: https://github.com/espressif/arduino-esp32

[14] R. Sundaram, "Engineering Project Activities Designed to Promote STEM Engagement," *Proceedings of the 2022 IEEE-ISEC conference, Virtual Event*, Princeton, NJ, March 2022. doi: 10.1109/ISEC54952.2022.10025178

[15] Z. Dickinson *et al*., "STEM Project Experiences with Wireless Sensor Networks," *Proceedings of the 2022 ASEE-NCS conference*, Pittsburgh, PA, March 19, 2022. https://peer.asee.org/39261