WIP: Integration of Hardware and Software Environments to Study Wireless Sensor Networks

Ramakrishnan Sundaram
Gannon University, sundaram@gannon.edu

Abstract - **This work-in-progress research-to-practice paper explores the integration of hardware and software environments to facilitate the study of wireless sensor networks. The hardware environment is the physical wireless sensor network which comprises WiFi modules, configured in a grid, to transmit and receive radio frequency signal data. The software environment is the virtual wireless sensor network comprising virtual sensor nodes arranged in a grid within the chosen simulator environment. CupCarbon is the wireless sensor network design and simulation tool used to interface with the WiFi modules used in the physical wireless sensor network. The interface between the physical network and the virtual network enables the signal data from the physical wireless sensor network to be assigned and monitored by the virtual sensor nodes in the simulation environment. Interfaces are designed and tested to compare physical and virtual network performance. The platform is being developed to test and model wireless sensor networks at different geometric scales and distinct spatial topologies. The unified platform will be useful in introducing actual and virtual sensor networks to students ranging from K12 STEM schools to graduate students in engineering programs through (a) the setup and testing of actual sensor networks (b) studies on modeling of sensor networks through simulation tools (c) real-time integration of the joint operation of both environments.**

Index Terms – wireless sensor network, radio frequency

INTRODUCTION

The ubiquitous nature of the internet-of-things, or IoT, as well as the emergent advanced technologies in sensor design, wireless data communication and computing are rapidly altering the engineering landscape. Wireless sensor networks (WSN) form the crucial backbone for the IoT to expand and flourish [1]-[3]. While the physical sensor network, configured at manageable scales, and can be tested and monitored, it is vital to understand the operation of the network through modeling and analysis in a virtual or simulation environment. This paper focuses on creating a unified platform by integrating the hardware environment with the simulation environment to study wireless networks and their performance. Specifically, the hardware environment comprises the setup of the wireless sensor network grid using an organized collection of sensor nodes which transmit and receive radio frequency (RF) signal data. The command center comprises the computing station (laptop/PC) to collect and display the received signal strength (RSS) data from each transmitting node in the network.

The integrated environment facilitates project activities which can be incorporated within the electrical and cyber engineering curriculum. Courses related to wireless communications are supported by laboratory experiments using the hardware environment and simulation exercises using the software environment. In addition, STEM outreach activities can be developed from the components of the integrated environment by defining the activities based on the STEM preparation of the target group of students [4], [5]. The combined environment enables integration of concepts learned across the gamut of STEM education – from the pK-12 framework through undergraduate and graduate studies.

In this paper, the components of the setup of the integrated environment is described. The outcomes of the testing of the integrated environment are presented and evaluated. Section 2 overviews the hardware environment and its operation. Section 3 summarizes the software environment developed using CupCarbon [6]. Section 4 discusses the operation of the integrated environment and summarizes the test outcomes. Conclusions and next steps appear in Section 5.

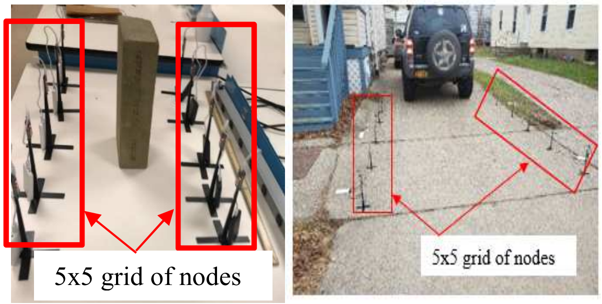
SECTION 2: HARDWARE ENVIRONMENT

Figure I illustrates two implementations of the hardware environment. Figure I(a) displays the setup of five transmitter nodes and five receiver nodes, together with the Wi-Fi router, in the laboratory or indoor setting. Figure I(b) shows the configuration in an outdoor setting. The hardware environment provides RSS data in the two stages identified as follows: (a) calibration stage (b) obstruction test stage. The calibration stage consists of RSS data in the absence of any obstruction within the WSN.

For the 5x5 grid of nodes, the 25 RSS values represent the base or reference set of the WSN. Each frame comprises a snapshot of these 25 values at the specific time instant. The calibration data is obtained by averaging the RSS values across five to ten frames. During the obstruction test stage, changes in the RSS values are captured as follows:

First frame:
Difference w.r.t. averaged data from the calibration stage

Subsequent frames:
Difference between the current frame and the previous frame



(a) (b)

FIGURE I
WIRELESS SENSOR NETWORK (a) INDOOR (b) OUTDOOR

Typical frames of RSS data are displayed in Figure II. Python script retrieves the RSS values at each receiver node using the Wi-Fi router. There are five sets of five RSS values, each representing the RSS at the chosen receiver from each transmitter, each timestamped at the start of the first reading. The RSS values are in negative dB values, indicating attenuation of the RSS levels within the grid enclosed by the WSN.

```
12:18:55
data: ['12:18:55', '1', '38', '64', '41', '55', '67']
data: ['12:18:55', '2', '43', '59', '46', '68', '61']
data: ['12:18:55', '3', '47', '70', '46', '67', '59']
data: ['12:18:55', '4', '63', '73', '51', '66', '67']
data: ['12:18:55', '5', '54', '67', '40', '62', '56']
12:19:23
data: ['12:19:23', '1', '38', '64', '41', '57', '64']
data: ['12:19:23', '2', '43', '59', '46', '68', '61']
data: ['12:19:23', '3', '46', '69', '46', '67', '58']
data: ['12:19:23', '4', '63', '73', '50', '65', '68']
data: ['12:19:23', '5', '54', '65', '38', '63', '55']
12:19:51
data: ['12:19:51', '1', '40', '65', '43', '55', '64']
data: ['12:19:51', '2', '41', '60', '47', '66', '57']
data: ['12:19:51', '3', '47', '70', '45', '68', '55']
data: ['12:19:51', '4', '63', '70', '48', '66', '67']
data: ['12:19:51', '5', '55', '63', '38', '63', '57']
12:20:19
data: ['12:20:19', '1', '39', '59', '41', '55', '63']
data: ['12:20:19', '2', '43', '58', '47', '68', '61']
data: ['12:20:19', '3', '49', '70', '47', '67', '60']
data: ['12:20:19', '4', '68', '72', '49', '66', '67']
data: ['12:20:19', '5', '52', '63', '39', '63', '57']
12:20:47
data: ['12:20:47', '1', '36', '65', '42', '56', '65']
data: ['12:20:47', '2', '40', '59', '46', '66', '60']
data: ['12:20:47', '3', '47', '70', '46', '68', '59']
data: ['12:20:47', '4', '65', '72', '50', '65', '66']
data: ['12:20:47', '5', '62', '65', '40', '63', '55']
```

FIGURE II
TIMESTAMPED RSS FRAMES – PYTHON OUTPUT

The tabulated data is captured in Excel as well, as shown in Table I.

TABLE I
RSS VALUES RECORDED – RECTANGULAR 5X5 GRID

Timestamp	Rx_ID	Tx_1	Tx_2	Tx_3	Tx_4	Tx_5
11:33:33	1	-57	-69	-66	-51	-56
	2	-45	-55	-44	-52	-52
	3	-70	-61	-54	-66	-62
	4	-67	-59	-51	-55	-61
	5	-62	-59	-54	-57	-52
11:33:45	1	-56	-71	-63	-50	-56
	2	-43	-55	-44	-49	-53
	3	-69	-58	-54	-67	-62
	4	-66	-58	-53	-54	-61
	5	-62	-59	-54	-58	-52
11:33:56	1	-56	-66	-69	-52	-55
	2	-44	-57	-43	-52	-54
	3	-68	-58	-54	-62	-63
	4	-67	-58	-52	-55	-59
	5	-65	-57	-54	-58	-51

SECTION 3: SOFTWARE ENVIRONMENT

CupCarbon is a versatile wireless and IoT simulation tool which not only enables the design and analysis of standalone virtual wireless sensor networks but also supports the interface of the simulation platform with the actual hardware platform presented in Section 2. Figure III depicts the software environment in CupCarbon with five virtual wireless sensor nodes configured as transmitters and five virtual wireless sensor nodes configured as receivers. The red arrows indicate transmission from each of the five transmitter nodes to the first receiver node (identified from left to right).

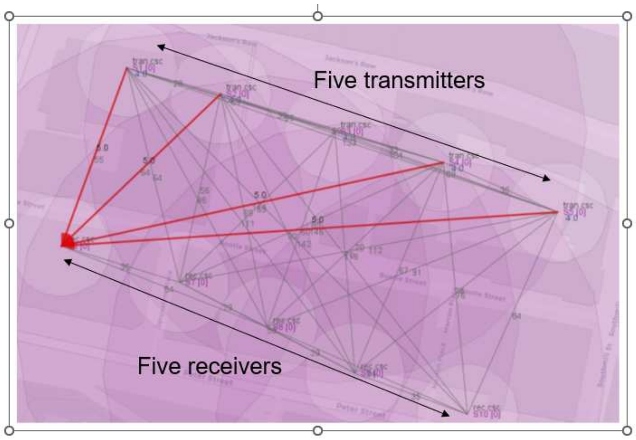


FIGURE III
SOFTWARE ENVIRONMENT IN CUPCARBON

Testing the virtual 5x5 WSN
Figure IV shows the capture of one frame of simulated RSS values at each receiver in the virtual 5x5 WSN. The received data, highlighted in the red box at each receiver, comprises a vector of five elements where each element represents the RSS value from the corresponding transmitter.

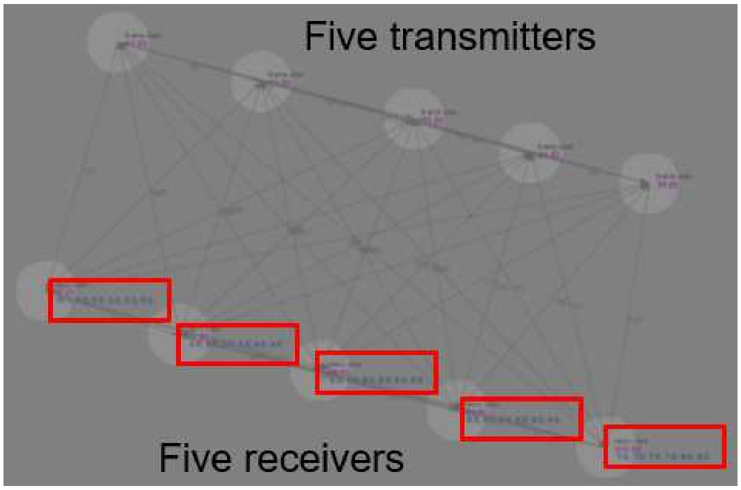


FIGURE IV
RECEIVED RSS VECTORS IN ONE FRAME OF SIMULATION

Testing the virtual 10x10 WSN
Studies with 10x10 virtual grids have been conducted and the outcomes during the second frame, after transmission to the second of the ten receivers, is displayed in Figure V.

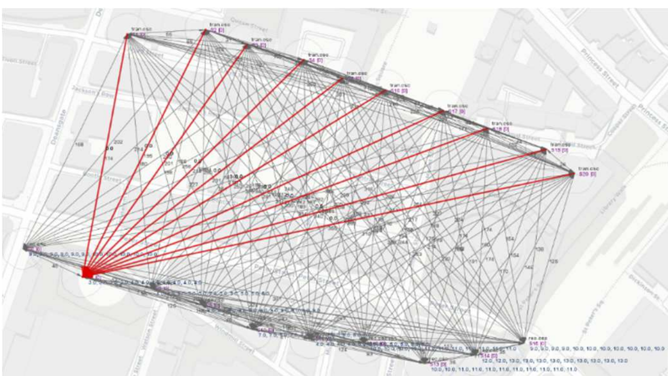


FIGURE V
RECEIVED RSS VECTORS DURING THE FRAME – EIGHT OF TEN RECEIVERS

SECTION 4: INTEGRATED ENVIRONMENT

The integrated environment comprising the hardware setup and the simulator platform is shown in Figure VI. Tests have been conducted on the 5x5 WSN based on ESP32 WiFi modules and the virtual 5x5 WSN configured in CupCarbon. The communication between the ESP32 and the CupCarbon simulator is achieved by the step-by-step procedure described in [7].

Test outcomes – calibration stage
The calibration stage comprises the capture of RSS data in the absence of any obstruction within the actual grid as shown in Figure VII. Figure VIII displays the RSS data captured at each of the five receiver nodes.

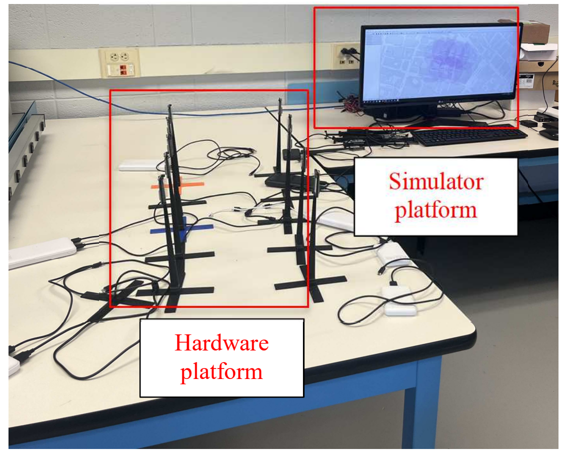


FIGURE VI
INTEGRATED ENVIRONMENT

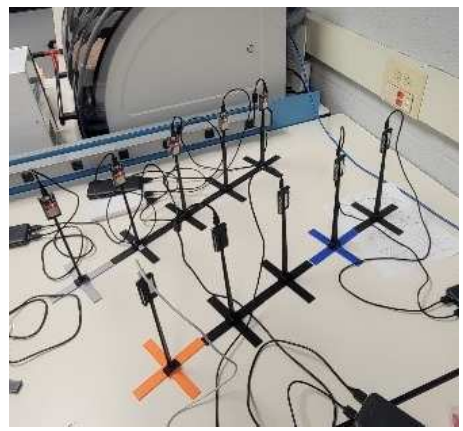


FIGURE VII
HARDWARE ENVIRONMENT – CALIBRATION

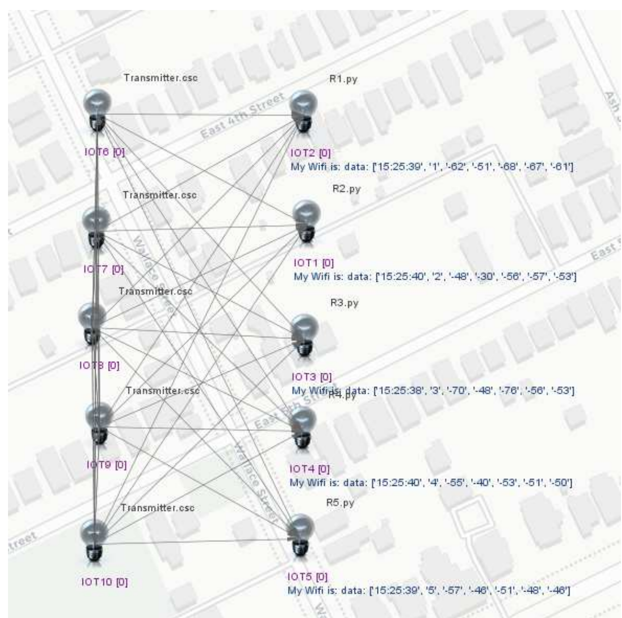


FIGURE VIII
SOFTWARE ENVIRONMENT – CALIBRATION

Test outcomes – detection stage

The detection stage follows the calibration stage and records changes in RSS data due to obstruction(s) within the actual grid. Figure IX shows the case with a single stationary obstruction within the actual 5x5 WSN. Figure X shows the RSS data at each of the virtual receiver nodes as recorded at the hardware setup.

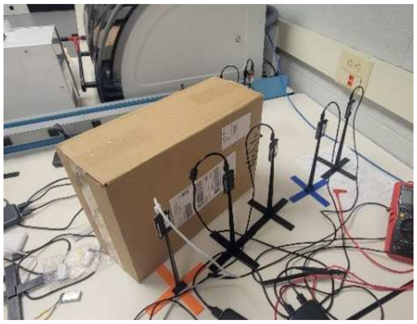


FIGURE IX
HARDWARE ENVIRONMENT – DETECTION

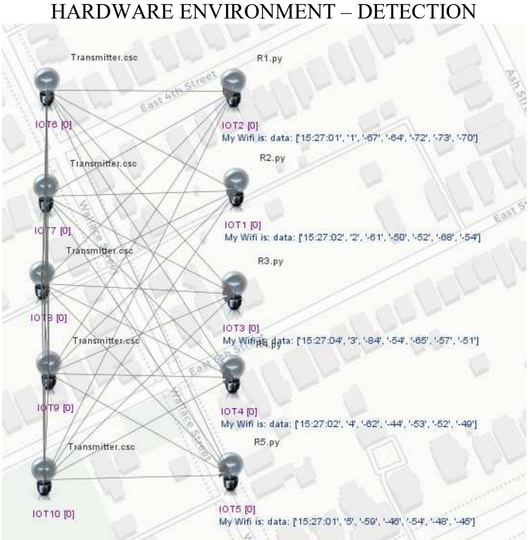


FIGURE X
SOFTWARE ENVIRONMENT – DETECTION

Table II displays one frame of RSS data values obtained during the calibration and detection stages respectively.

TABLE II
RSS DATA: CALIBRATION vs. DETECTION

Calibration	Detection
-62, -51. -68, -67, -61	-67, -64, -72, -73, -70
-48, -30, -56, -57, -53	-61, -50, -52, -68, -54
-70, -48, -76, -56, -53	-84, -54, -65, -57, -51
-55, -40, -53, -51, -50	-62, -44, -53, -52, -49
-57, -46, -51, -48, -46	-59, -46, -54, -48, -45

Note that the lower RSS values in either the calibration or detection stage, which are the larger negative entries (measured in dB) in the element-to-element comparison of the values between the two arrays are indicated in red. The entries in green represent the corresponding smaller negative entries i.e., the lower attenuation in the RSS level. The values that are unchanged are indicated in black. Clearly, the presence of the obstruction increases the attenuation at a majority of the twenty-five entries in the 5x5 grid.

SECTION 5: CONCLUSIONS AND NEXT STEPS

The proposed engineering research platform, comprising wireless sensor nodes in the hardware environment integrated with the virtual wireless sensor nodes in the software environment, promotes the participation and advancement of faculty and student research and curriculum development 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. Work is ongoing to study the effect of regularization on the recovery of the image representing the entity or obstruction within the network. The goal is to track the entities within the network by solving the inverse reconstruction problem using RSS attenuation data on each wireless link within the network.

REFERENCES

[1] H. Kim, "Low power routing and channel allocation method of wireless video sensor networks for Internet of Things (IoT)," *Proceedings of the 2014 IEEE 2nd World Forum on Internet of Things (WF-IoT)*, Seoul, South Korea, Mar. 2014; DOI: 10.1109/WF-IoT.2014.6803208.

[2] S. Shankar *et al*., "Development of Efficient Wireless Sensor Network for IoT Applications," *Proceedings of the 2023 3rd International Conference on Pervasive Computing and Social Networking (ICPCSN),* Salem, India, June 2023; DOI: 10.1109/ICPCSN58827.2023.00237.

[3] E. Pantoya *et al*., "Advancing Wireless IoT Sensor Networks With Edge-Powered RFID Devices," *Proceedings of the 2023 IEEE 14th Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON)*, New York, NY, Oct. 2023; DOI: 10.1109/UEMCON59035.2023.10316020.

[4] Zachary Dickinson, Tyler Seelnacht, Ram Sundaram, "Engineering Project Activities Designed to Promote STEM Engagement," *Proceedings of the. IEEE ISEC 2022 conference*, Princeton, NJ, March 2022; DOI: 10.1109/ISEC54952.2022.10025178.

[5] Ram Sundaram, "Research Platform to Promote the Evolution of STEM Project Experiences," *Proceedings of the. IEEE ISEC 2023 conference*, Laurel, MD, March 2023; DOI: 10.1109/ISEC57711.2023.10402314

[6] CupCarbon® User Guide Version U-One 5.1.

[7] C. Maier and R. Sundaram, "Unify and Broaden STEM Outreach Activities through Simulator Interfaces for Wireless Sensor Networks," *Proceedings of the 2024 IEEE-ISEC conference*, Princeton, NJ, Mar. 2024.