# Experiences Teaching a Wireless for the Internet of Things Course Cooperatively at Multiple Universities

Nabeel Nasir
nabeeln@virginia.edu
University of Virginia
Charlottesville, USA

Viswajith Govinda Rajan
gyx4bw@virginia.edu
University of Virginia
Charlottesville, USA

Pat Pannuto
ppannuto@ucsd.edu
University of California, San Diego
La Jolla, USA

Branden Ghena
branden@northwestern.edu
Northwestern University
Evanston, USA

Bradford Campbell
bradjc@virginia.edu
University of Virginia
Charlottesville, USA

## ABSTRACT

Today's computational devices are overwhelmingly wireless. To realize wireless communication, today's devices use a grab bag of protocols (Bluetooth, WiFi, 4G/5G, LoRa, NFC, etc.) and no one universal standard has emerged. This diversity presents a ripe pedagogical opportunity to introduce students to the fundamental tradeoffs and design decisions inherent to wireless communication and networking. Furthermore, many wireless protocols are accessible to study in a classroom (in fact, many we all use daily), which lends to a very hands-on course.

We report on our experience teaching a new "Wireless Networking for the Internet of Things" course in three R1 universities across six offerings, with sections both in quarter and semester format and for undergraduate, graduate, and professional-master's levels. We share the scope of the covered topics, our approach for making the course interactive and hands-on, lessons learned from multiple iterations, adaptations to fit within different prerequisite chains, and different structures to adapt to different delivery formats.

## CCS CONCEPTS

• **Social and professional topics** → **Computing education**; • **Computer systems organization** → **Embedded and cyber-physical systems**.

## KEYWORDS

Internet of Things, Wireless Protocols, CS Education

**ACM Reference Format:**
Nabeel Nasir, Viswajith Govinda Rajan, Pat Pannuto, Branden Ghena, and Bradford Campbell. 2024. Experiences Teaching a Wireless for the Internet of Things Course Cooperatively at Multiple Universities. In *Proceedings of the 55th ACM Technical Symposium on Computer Science Education V. 1 (SIGCSE 2024), March 20–23, 2024, Portland, OR, USA.* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3626252.3630848

*SIGCSE 2024, March 20–23, 2024, Portland, OR, USA*
© 2024 Copyright held by the owner/author(s).
ACM ISBN 979-8-4007-0423-9/24/03
https://doi.org/10.1145/3626252.3630848

## 1 INTRODUCTION

The Internet of Things (IoT) is a paradigm for ubiquitous connected devices that is projected to reach 100s of billions connected devices by 2030 [1–3]. This growth will require a new cohort of trained engineers with the knowledge and experience to work across multiple (lower) layers of the computing stack. Whereas other computing platforms have robust abstractions with stricter layer boundaries, the low-resource nature of IoT devices necessitates thinner abstractions and a tighter coupling between layers. For IoT engineers to be successful, they need training and practice working with the software and hardware details, and constraints, of IoT-class computers.

To help meet this increasing demand, we developed a new IoT-centric course focused on the diverse set of wireless protocols designed for low-power IoT devices. We strategically focus on the wireless and networking aspects of IoT to provide a unique, hands-on experience for students, but also because wireless networks prompt rich design-space tradeoffs for students to grapple with as they learn about how IoT systems are built. Importantly, this course is not an embedded systems course, and is designed to appeal to computer scientists, computer engineers, and electrical engineers. In this paper we discuss how we designed this course, our experience offering this course over multiple terms, and our lessons learned from creating this new opportunity from students.

One unique aspect of our experience is that we have taught this new wireless IoT course at multiple R1 institutions at different levels and on different university schedules. This necessitated building a flexible and modular design for the course that can scale to different lengths and different focuses to meet the needs of a particular class of students, instructor preference, and university calendar schedule (i.e. quarters versus semesters). To date, we have offered the wireless IoT course six times over several years at three universities: at Northwestern University, University of California San Diego (UCSD), and University of Virginia (UVA).

To make the course accessible and of sufficient interest and pedagogical value to a diverse audience of upper-level undergraduates through early-career PhD students, we focus on the communication design decisions system designers must grapple with when creating a new IoT device. This enables the course to integrate well into the mesh of existing courses at our universities by building on concepts from computer networking, wireless modulation and communication, and low power systems design. This integration has catalyzed the growth of this course across universities.

This paper discusses the key lessons derived from teaching a highly multidisciplinary course with different formats, and how variations in university-specific factors went into designing the course syllabus and materials. This paper also offers a framework for faculty from other universities who are interested in taking the wireless IoT course and applying it to their academic context.

## 2 RELATED WORK

There has been a recent push towards utilizing embedded hardware in CS courses. Multiple panel discussions and workshops have been organized at SIGCSE with this focus [5, 14, 36]. There have also been efforts to outline how best to offer IoT courses and best practices to aid instructors in teaching IoT courses for the first time [12]. Our course also utilizes embedded hardware but specializes on teaching wireless protocols within the context of IoT. We were unable to find papers which describe best practices in setting up labs with this specialization, which we include in this work.

There are a few experience reports on IoT courses in the literature, notably by Ali [6, 7], Mäenpää et al. [23], and Förster et al. [18]. Both the courses by Ali [6, 7] and Mäenpää et al. [23] have their focus on embedded programming. In contrast, our labs focus on teaching wireless for IoT concepts and strive to minimize the difference from a ‘traditional’ execution environment to keep focus on communication-related concepts. The course offered by Förster et al. [18] has few wireless labs (WiFi, BLE, 802.15.4), but also explores embedded programming, embedded OSes/runtimes such as Contiki [16], wireless sensor networking (WSN) principles, etc.—the course introduces the breadth of IoT. Our course gives students a working experience on multiple wireless protocols to understand tradeoffs and to identify the best protocol for a given use case. Another key difference is that these prior experience reports are for a same type of offering (semester/quarter/summer) or target audience (undergraduate/graduate), whereas this paper encompasses experiences from multiple universities with different offering types and target audiences.

We found syllabi for four courses which explore wireless protocols for IoT, from CMU [37], UIUC [21], PennState [20], and Rensselaer [4]. We have referenced the first two courses [21, 37] while building our material as they cover some of the protocols used in IoT, but their focus is more on the wireless networking aspects like MAC concepts, localization etc., than on the protocols for IoT. The course from PennState [20] is a seminar which provides a good coverage of the current state-of-the-art for IoT with a focus on more advanced wireless networking and mobile sensing topics like GPS and localization, Batteryless networking etc. In our course, we look at more fundamental wireless protocols used in IoT and utilize labs to provide a hands-on learning experience. The one from Rensselaer [4] is very similar to our course in this regard, having labs on wireless protocols, but also covers additional topics like IoT transport protocols, IoT security etc. Unlike these courses, the gap we are trying to address with our course is to provide students with experience and skills to compare between various wireless IoT protocols, critically analyze design choices, and to become good IoT practitioners. In addition, there are no experience reports for these or other other wireless for IoT courses to the best of our knowledge, which we try to address with our work.

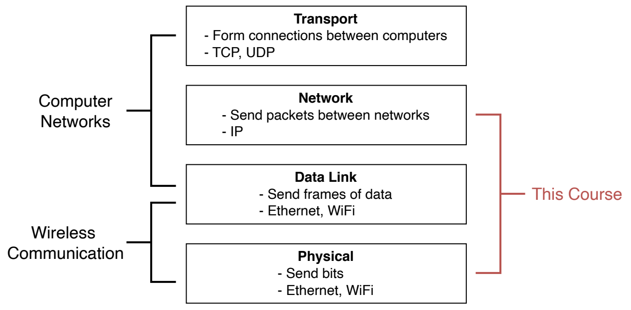

**Figure 1: How Wireless for IoT fits in the CS curriculum and complements other courses in terms of topics covered in the OSI model of communication.**

**Table 1: Distillation of Revised Bloom’s Taxonomy [9], reproduced from ISU CELT’s effective teaching practices site [17].**

| Knowledge → ↓ Cognitive Proc. | | **K1** Factual | **K2** Conceptual | **K3** Procedural | **K4** Metacognitive |
|---|---|---|---|---|---|
| **C1** | Remember | List | Recognize | Recall | Identify |
| **C2** | Understand | Summarize | Classify | Clarify | Predict |
| **C3** | Apply | Respond | Provide | Carry Out | Use |
| **C4** | Analyze | Select | Differentiate | Integrate | Deconstruct |
| **C5** | Evaluate | Check | Determine | Judge | Reflect |
| **C6** | Create | Generate | Assemble | Design | Create |

## 3 COURSE OVERVIEW

Our course centers on how wireless protocols interface with system design for Internet of Things devices. Fundamentally, this course asks “Why are there so many wireless protocols for IoT?” and “Why is there not one clear and obvious choice?”. This means we introduce the physical layer and modulation schemes to explain tradeoffs among bandwidth, transmission range, bitrate, and reliability, but do not cover the mathematical foundation for physical layer modulation schemes. We emphasize the medium access control (MAC) layer and different network topologies to highlight the strengths and weaknesses of wireless IoT protocols such as Bluetooth Low Energy, LoRa, and Thread. Finally, we introduce data models and upper network layers with hands-on programming assignments for students to gain first-hand experience with how to build IoT applications using IoT wireless protocols.

The course builds on and reinforces embedded programming, but does not focus on fundamental embedded programming or embedded systems concepts. The course couples well with a traditional course on physical layer and wireless modulation as students in Wireless IoT see how the theoretical techniques are used in real-world protocols. Computer networking courses typically focus on higher layers of the OSI model, and our course both illustrates how those layers can work for IoT and why many IoT protocols eschew them. Finally, capstone-level embedded systems courses sometimes introduce wireless technologies and their high-level APIs, and our course explains why those APIs are designed as they are. Figure 1 shows how the course fits in a CS curriculum alongside wireless communication courses and computer networking courses.

### 3.1 Learning Goals

To guide course design we developed a formal suite of learning goals. We iterated internally as a course development team on these goals and consulted other faculty at our institutions and researchers in the course topic areas for further feedback. With assistance from faculty more experienced in defining formal goals, we mapped our learning goals onto the Revised Bloom's Taxonomy, summarized in Table 1, and then iterated further to increase the number of higher-level learning outcomes achieved by the course.

(1) **C4: K2, K4** Analyze a new wireless technology or protocol to extract key technical features and limitations.
(2) **C4-5: K1-2** Assess how physical-world constraints of an application scenario map to the capabilities of communication technologies.
(3) **C2: K1** Explain the basic operating principles and performance of most-used technologies in mobile computing.
(4) **C2-3: K1** Explain what "{B,P,L,W,R}AN", "star", "mesh", and "cell" mean in wireless networking, and how topology influences system design and performance.
(5) **~C3: K3** Demonstrate basic self-sufficiency in the compilation, loading, and testing of previously-unseen software on previously-unseen hardware platforms.
(6Q) **C6: K3** Design a system architecture and estimate its performance given an application scenario.
(6S) **C6: K4** Create an original system that realizes performance requirements for a novel application scenario.

While most goals remain the same across course modality, the extra time afforded by a semester allows for more metacognitive work.

## 3.2 Course Topics

To explore the design space of wireless networks we strategically include course modules on point-to-point networks (Bluetooth Low Energy), mesh networks (IEEE 802.15.4 and Thread), hub-and-spoke networks (WiFi), and long-range networks (LoRa and 4G/5G cellular). These network types illustrate how underlying physical constraints affect protocol design and how protocol design affects use cases and software interfaces.

We introduce these topics in lecture and re-enforce learning with in-class and out-of-class hands-on assignments where students build devices to use each network. With this foundation, we introduce students to additional specialized wireless protocols which further expand the design space, including Visible Light Communication (VLC), Citizens Broadband Radio Service (CBRS), Sigfox, and Cellular IoT (LTE-M and NB-IoT). We also introduce non-communication-focused wireless-based techniques such as wake-up radios, ranging, indoor localization, and GPS.

## 3.3 Hands-on Hardware

To enable our learning objectives, we provide student groups with hardware development platforms. We use Nordic Semiconductor's nRF52840 development kit [34] and nRF52840 dongle [35] for teaching BLE, 802.15.4, and Thread. We have the Heltec Automation WiFi LoRa 32 v3 board [10] for teaching WiFi and LoRa, which uses the popular ESP32 microprocessor [38]. All three of these boards are reasonably priced (nRF52840 development kit: $50, nRF52840 dongle: $10, Heltec board: $20) and have decent documentation online. The devices are shown in Figure 2.

We use VSCode [26] for development and use the nRF Connect [32] and PlatformIO [22] extensions to use with these boards. We use standard software platforms including the Linux Foundation's Zephyr RTOS [39] and/or the Arduino interface on the nRF boards due to availability of sample code, good documentation, and widespread usage in the embedded community.

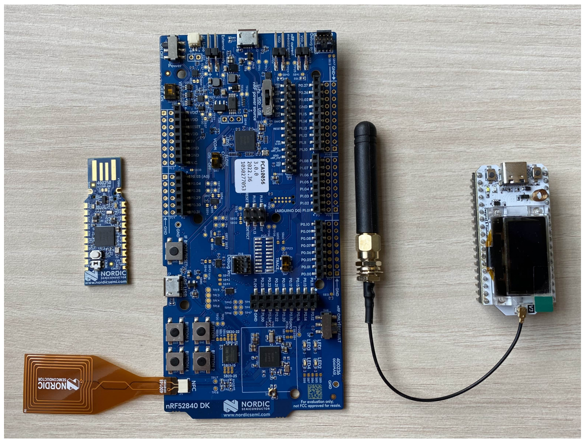

**Figure 2: Hardware platforms we use for the course. From left: nRF52840 Dongle, nRF52840 Development Kit, and Heltec WiFi LoRa 32 v3.**

**Table 2: Statistics and demographics for the course offerings across the three universities.**

| | **Northwestern** | | **UCSD** | | | **UVA** |
|---|---|---|---|---|---|---|
| **Duration** | 10-week quarter | | 10-week quarter | | | 16-week semester |
| **Session** | WI'23 [19] | WI'23 [19] | WI'22 [28] | FA'22 [27] | WI'23 [29] | SP'23 [13] |
| **Level** | UG | G | G | UG/G | PM | UG |
| **Cadence** | 80 m T/R | 80 m T/R | 50 m M/W/F | 50 m M/W/F | 8 h bi-wkly | 75 m M/W |
| **Enrolled** | 20 | 13 | 13 | 37 | 7 | 58 |
| **Major** | CS: 12 ECE: 6 Other: 2 | CS: 3 ECE: 3 Other: 7 | CS: 9 CE: 4 EE: 0 | CS: 9/13 CE: 10/3 EE: 0/2 | CS: 2 CE: 0 EE: 5 | CS: 53 CE: 2 EE: 3 |

**Table 3: Components for latest offering of course at each of the three universities.**

| | Northwestern | UCSD | UVA |
|---|---|---|---|
| Lectures | ✓ | ✓ | ✓ |
| Labs | ✓ | ✓ | ✓ |
| Programming Assignments | ✓ | | ✓ |
| Homework | ✓ | ✓ | ✓ |
| Mini Quizzes | | ✓ | ✓ |
| Final Exam | | | ✓ |
| Design Project | ✓ | ✓ | |
| Implementation Project | | | ✓ |
| Guest Lectures | | | ✓ |

## 3.4 Comparison of Course Offerings

There are some variations in how the course is offered at the three universities due to differences in time available for instruction, instructor preferences, and other parameters. To give better context about these differences, we provide the course statistics and target demographics of the latest offering at each university in Table 2. The course webpages for each offering, which include syllabus and publicly available course materials are also cited in Table 2. Northwestern and UCSD operate on a quarter system, whereas UVA has a semester system. The course is offered only at the undergrad level at UVA, whereas Northwestern and UCSD both offer it for undergrads (UG) and graduate students (G). UCSD also has a Professional Master's (PM) offering, which consists of 5 full-day sessions of 8 hours each, taught over 10 weeks.

There are variations in the course components at the three universities as well, and we list this in Table 3, along with the course website for the latest offerings. A repository of slides for the lectures, materials for the labs, homework, and mini quizzes, as well as a Github repository for labs and programming assignments are available upon request. As these course components are mostly self explanatory, we do not describe these in detail for brevity.

## 4 COURSE DESIGN INSIGHTS

Based on our collective experience teaching this course several times in various contexts, we share insights likely helpful for future instructors interested in designing a similar course.

### 4.1 Removing Pre-requisites for the Course

*Insight: focusing on wireless protocols and not embedded programming supports lecture-lab separation and attracts more CS students.* Realistic hands-on experimentation with wireless protocols requires using embedded devices. This comes at a risk of complexity, however, as not all students, particularly CS students, are familiar with embedded systems development. We learned that emphasizing wireless networking, even in the context of IoT devices, removes the need to emphasize embedded systems development and programming. This increases the number of students who are prepared to take the course without sacrificing content depth.

Not requiring embedded systems experience also removes onerous pre-requisite requirements. UCSD and UVA offer the course with no specific pre-requisites, instead requiring 'some upper-level CS systems experience'. This does not preclude more pre-requisites as appropriate, and our Northwestern offering requires students to have taken an embedded systems or computer networking course.

Minimizing the focus on embedded concepts helps separate lecture and labs as lecture does not need to cover embedded concepts. Lectures focus on the wireless concepts whereas labs focus on implementing the concepts on actual devices. For example, to make the course material more easily accessible, we favor the use of a simpler programming environment like Arduino. This helps reduce the programming complexity and helps focus on wireless concepts.

For students who are less familiar with general low-level C programming concepts, we use homework and specific references to help students become suitably familiar with low-level basics to be able to complete the labs.

### 4.2 Hardware Selection is Critical

*Insight: sharing hardware resources enables creative freedom when designing labs.* We assumed that using the same hardware platforms across universities would reduce the overhead of offering this course, but only if we used the same lab assignments. Otherwise, we expected it would be simpler to use hardware platforms each instructor was already familiar with. This turned out to be mostly false. Agreeing on a common set of hardware platforms was *essential* to effective sharing of course resources, and actually *increased* instructor flexibility.

By sharing hardware resources across universities, every idea, lab, or assignment created by one instructor became an immediate starting point for the others. This allows us to both customize for our particular offerings but also to be more creative when designing new labs as we are not starting from scratch. Sharing hardware platforms has the benefit of ensuring that the labs or assignments would work because a different university implemented it successfully. It further eases burden when things do go wrong. For example, UCSD developed the LoRa lab in summer 2022 as a new addition. The FA'22 offering, unfortunately, had to scrap the lab due to unforeseen issues with the Heltec board.[1] As the other labs were turnkey and proven, UVA could use course prep time to fix the new lab, which Northwestern and UCSD picked up seamlessly in their next offerings.

If a new instructor were to build upon the lab and create a derivative, they simply have to work on the additional part and do not have to build the lab from ground up. For example, because UVA has semesters, they were able to build additional assignments like using the nRF52840 board to blink LEDs with BLE notifications. In this case, the base use case of ensuring the BLE stack worked on the board was proven by Northwestern and UCSD. UVA only had to design the assignment parts, which let them experiment with more creative ideas as to what could be done with the existing hardware instead of ensuring the hardware worked in the first place.

Finally, interleaving offerings across institutions spreads the maintenance burden of inevitable 'bit rot.' The software environment for the BLE lab required ever-changing workarounds for students with Apple M1 silicon. The resource pointers and the interactive command interface from OpenThread had little changes that broke lab instructions. These types of 'small' things add up and make 'real-world' hardware courses major teaching burdens. As a more specialized class, it is unlikely to be taught more than at best annually at any one institution. Teaching across institutions allows for a more aggressive cadence, and shortens the window of time for external, breaking changes to accumulate.

### 4.3 Quarter System Schedule

*Insight: for best learning outcomes on the quarter system, course schedules may need tweaking of typical pedagogical components.* UCSD and Northwestern run on a quarter system. Semesters have a longer time window, requiring course material to be paced differently.

***a. Labs:*** Northwestern held asynchronous labs with labs not held in class. This required detailed instructions to ensure students could get the lab running by themselves with minimal guidance. Labs effectively functioned as a lab-and-programming-assignment bundle. UCSD used one of the thrice-weekly, 50 min meetings to 'kick off' labs, but much of the lab activities were facilitated outside the class by a very active teaching assistant (TA) who was crucial for guiding students to success. In contrast, UVA was able to roughly dedicate one in-class session each week to labs and had multiple TAs and instructors assisting students during the lab session.

***b. Final Projects:*** UCSD forwent the traditional final project structure and substituted it with a design writeup project. The goal of this assignment is to come up with a project/product idea, perform requirement analysis on it, choose the right protocols

[1]The lab was developed using non-WiFi-enabled Heltec LoRa boards used frequently in existing research. During the term, it was discovered that the WiFi-enabled Heltec LoRa boards require a completely different hardware support library, and there was insufficient time to create a student-friendly, robust lab.

and design parameters, and submit a design pitch document. This writeup is similar to the project proposals used in industry,[2] and adds value to the course. The writeup requires a comparison of all protocols discussed in the class, with a deep-dive into two that seem most applicable for the particular application.

Northwestern started with final projects, but found them difficult to implement successfully because of the short timeline with quarters. Final projects were offered for two quarters with mixed results in terms of project quality. To provide enough time, they were started before lectures finished, but that meant that student groups focused only on topics from the first half of the course. After the success of design writeups at UCSD, the most recent Northwestern offering switched to that format with good results.

## 4.4 Teaching Format Flexibility

*Insight: modularity of topics and time aid in modifying the quarter offering to alternate durations.* We found the course structure lends itself well to adapting to various offering formats. To adapt to a semester schedule, we added some fairly straightforward elements:

***a. Final Exam:*** UVA was able to conduct a final exam about 75% of the way into the semester after all the lectures wrapped up, serving as a traditional test metric.

***b. Guest Lectures:*** UVA invited industry experts and researchers to give talks on industry use of wireless protocols and novel protocols. Some of the topics that were covered in the guest lectures were: wakeup-radio networks, Visible Light Communication, and 4G over Navy channels. The students learn methods to evaluate wireless protocols through the lectures and labs, and can apply those methods to evaluate the protocols discussed during the guest lectures.

***c. Final Project:*** The longer timeline permitted dedicating the final month of the course to final projects. Students came up with ideas for their projects which were based on wireless protocols they learned over the semester, and met with the instructors and TAs to discuss these ideas. They used physical hardware and gateways and their projects were of fine-to-impressive qualities. The project concluded with students demoing their work, and submitting a report.

To adapt the course to a professional master's program with only a few day-long meetings, we simply covered one protocol at each meeting which nicely divided up the course. The roughly four, 50 min lectures per-topic condensed into a 4 h morning session with a 4 h afternoon session available to complete the bulk of the lab work. As the labs were designed to be completed outside of class, students were able to complete any unfinished work at home between the every-other-week sessions.

## 4.5 Applicability to Many Learners

*Insight: IoT wireless protocols synergize with traditional curriculums but are often not taught.* We find that wireless IoT protocols as a course topic enables flexibility for teaching the course to various student audiences. The topic bridges many concepts from CS and EE, making it suitable to cross-list in both programs. For example, concepts like software engineering of embedded networking stacks, observing physical layer in practice, wireless radio states and low power modes, data formats and endianness, and so on are topics at the intersection of CS and EE. Further, as many students will not have seen the material in an undergraduate curriculum, it is suitable to teach at the graduate level as well. This flexibility increases the value of offering this course as it can be adapted to meet various institutional goals and priorities.

[2]Indeed, several of the professional master's students commented in course feedback that they had been tasked to prepare very similar documents in their day jobs.

## 5 LESSONS LEARNED

We describe some lessons from designing and offering this course, which we hope will help instructors when replicating this course.

**Maximize lab time by offloading setup to a pre-lab.** We did not want lab time to be spent on students setting hardware and development environments. To prevent this, we give students a lightly-graded pre-lab which requires students to follow setup instructions in the lab manual, and have it ready before the lab session. This was very effective in saving time and also allowed students the opportunity to use office hours if needed.

**Thread networks are hard to setup, but great for collaborative learning.** We faced issues in setting up a Thread network and keeping it active for a prolonged duration due to specific implementation details. But despite this, using a Thread network makes the lab session an engaging and collaborative experience for students. We encourage instructors to setup a Thread network and display the live nRF Thread topology monitor [33] in the lab. Building a shared, collaborative network promotes community in the class.

**Collaborative offering and use of real-world networks enables the adoption of current research.** We note that cooperatively teaching this course frees up time for instructors to explore current research, and faster adoption of more recently published works. Wireless IoT topics are constantly evolving and the course provides value by covering the state-of-the-art. With multiple instructors introducing research topics, we cover topics including measuring power draw of BLE advertisements [30], retrieving data from LoRa packets collisions [11, 15], and exploring interoperability using the Matter protocol [8].

Also, using real-world wireless networks acts as a starting point for research. The labs for this course use actual wireless networks instead of toy examples, and experiences in class can bootstrap future research projects. Networks developed to support the class can be maintained and reused beyond teaching. This has a snowball effect, as the established network with many active devices becomes a better learning tool. Course alumnae from Northwestern and UCSD have entered graduate research programs based on experience from the course and subsequent research opportunities.

**Students can switch between platforms comfortably.** We anticipated that avoiding switching development environments throughout the course would curb unnecessary difficulty not related to our learning objectives. We tried to select a common software platform for use across course modules and hardware platforms. This turned out to be the wrong approach. We had hoped that VSCode's PlatformIO extension would support multiple boards and reduce "getting started" overhead. Instead, the general platform made supporting each board harder, and students adapted well when we had to change development environments throughout the term anyway.

We recommend instructors focus on the first IDE to get students familiar with common concepts like project configuration, building, flashing, and monitoring serial connections. Once they are familiar, they will find switching platforms easier.

**Utilize smartphones for better student engagement.** Smartphones have BLE, WiFi, NFC, and cellular available and are a great opportunity to use in class. For example, to teach about RSSI (received signal strength indicator), we asked students to pair up and use the nRF Connect smartphone app [31] to investigate the RSSI values when they move their phones away from each other. We asked students use the same app to interact with BLE peripherals they programmed during the BLE lab and assignment. Students can also leverage smartphones for final course projects.

**Challenge to sync lecture materials among instructors.** We observe that sharing lecture slides involves opening each other's slides and adapting the material to our individual styles and adjusting content for different lecture lengths. However, we found it was difficult to discover updates in each other's slides and to keep slides synchronized. Although we haven't tried this, one option is to use the compare and merge feature [25] in Powerpoint, however, this is not available for Mac platforms as of this writing [24].

## 6 COURSE FEEDBACK

In this section, we provide evidence from student comments and evaluation scores to support insights we offer in Section 4.

**Attracts more CS students**. Students appreciated that this course was accessible to CS students and felt that they were given support to succeed. One student commented: *"the course did a good job of addressing concepts so people from a wide range of backgrounds could learn. While I was already familiar with access control, packets, some wireless basics like modulation and such, if I hadn't, I believe I would have picked up that throughout the course. "* (Northwestern)

They also valued the opportunity to learn topics outside of CS: *"Lots of hands on time with the dev kits gave me, personally, a really fresh experience I hadn't gotten before and there was a lot I got to learn from it."* (UVA), *"I'm honestly kinda surprised it's listed solely as a CS course just because a lot of the class is about how information is encoded onto radio waves and the physical challenges involved in sending and receiving waves without interference or in ways that are resilient to disturbances. You definitely learn about a lot of cool things, and it would definitely give you enough knowledge to get started doing IoT things."* (Northwestern).

**Adaptable to different types of offerings**. Students appreciated the utility of the final design project and the overall effectiveness of the course for the quarter offering. *"the final project was also interesting, really applicable to what engineers have to go through balancing different constraints, cost analysis, looking up documentation etc."* (Northwestern). *"The collaborative aspect of the labs and the solo-focus on the homeworks and final design project was a really good way to approach learning. We learn a lot by discussing and engaging with our classmates, and then we can actually apply what we learned to the homeworks/project."* (UCSD).

UVA had a semester offering which added programming assignments to expand on labs, and students appreciated their learning from the assignments. *"The lectures were very helpful to provide a baseline, the labs built on this, and then the assignments and homeworks heightened our knowledge."*, and *"Labs were very good and served as an excellent preparation for the assignments."* (UVA).

**Applicable to many learners**. At UCSD, this course was offered for both undergraduate and graduate students, and the evaluation scores and comments highlight the adaptability of this course. Of the 14 undergraduate as well as graduate students who provided evaluations, 92.9% of the undergraduate students (13/14) and 91.7% of the graduate students (11/12, 2 did not respond) agreed or strongly agreed that this course was intellectually stimulating. 85.7% of the undergraduate students (12/14) as well as the graduate students (12/14) recommended this course overall.

**Bridges topics from CS and EE well**. Students agree that the course bridges CS and EE concepts, and they valued a diverse classroom. *"Honestly this material should be required for all Computer Engineering majors, it ties together computer science/programming and electrical engineering quite well, learning different ways how wireless communication works. (UCSD)". "this should be a main class and not just a special topics elective as the Internet of Things is a huge (part) of EECS in the world of STEM. (UVA)". "I like the different skill set, class discussions were diverse as a result and it was interesting to work with different majors for the lab group. (Northwestern)".*

## 7 FUTURE WORK

We identify some changes for future offerings of this course:

- We want to explore the possibility of students implementing a minimum viable product prototype for design projects in a quarter offering. Students can gain experience implementing their design without requiring full-fledged implementations.
- To encourage student learning with evaluating design choices and tradeoffs, we intend to introduce a design critique in addition to student-led design creation. We would provide a template wireless system design for students to evaluate (or possibly have them evaluate peer designs) based on system design tradeoffs.

## 8 CONCLUSION

Our experience cooperatively developing an Internet of Things course focused on wireless protocols shows how an engaging, cutting-edge course with broad applicability can be shared among universities, enabling individual instructors to offer a complex course with reduced effort. Despite sharing, we find each instructor can adapt the course to their taste, student demographics, and university calendar. We find that key to this is the inherent modularity of the topic coupled with strategic hardware selection. We share this experience to encourage new instructors to adapt this course at their institutions and to serve as a model for creating future engaging upper-level courses for computer science undergraduates without overburdening individual instructors.

## ACKNOWLEDGMENT

This work was in part supported by the National Science Foundation grants CNS-2144940 and CNS-2233894. We thank the reviewers for taking the time and effort to provide valuable feedback. We thank Leo Porter for feedback on learning goals, and Mark Sherriff for suggesting changes to improve the paper.

## REFERENCES

[1] 2022. Growth of IoT. https://techjury.net/blog/how-many-iot-devices-are-there/. Accessed: 05-25-2022.

[2] 2022. IoT growth during pandemic. https://www.pymnts.com/pymnts-post/connectedeconomy/2022/connected-device-use-doubled-during-the-pandemic/?c=whats-hot. Accessed: 03-25-2022.

[3] 2022. Number of connected devices. https://iot-analytics.com/number-connected-iot-devices/. Accessed: 03-25-2022.

[4] Alhussein Abouzeid. 2023. *IoT Spring 2021: ECSE 4660/6660*. Retrieved August 1, 2023 from https://sites.ecse.rpi.edu/~abouzeid/courses-taught/iot-spring-2021-ecse-466066.html

[5] Joel C. Adams, Richard A. Brown, Jalal Kawash, Suzanne J. Matthews, and Elizabeth Shoop. 2018. Leveraging the Raspberry Pi for CS Education. In *Proceedings of the 49th ACM Technical Symposium on Computer Science Education* (Baltimore, Maryland, USA) *(SIGCSE '18)*. Association for Computing Machinery, New York, NY, USA, 814–815. https://doi.org/10.1145/3159450.3159611

[6] Farha Ali. 2015. Teaching The Internet of Things Concepts. In *Proceedings of the WESE'15: Workshop on Embedded and Cyber-Physical Systems Education* (Amsterdam, Netherlands) *(WESE'15)*. Association for Computing Machinery, New York, NY, USA, Article 10, 6 pages. https://doi.org/10.1145/2832920.2832930

[7] Farha N. Ali. 2018. Experiences in Teaching the Internet of Things Courses. In *Proceedings of the 49th ACM Technical Symposium on Computer Science Education* (Baltimore, Maryland, USA) *(SIGCSE '18)*. Association for Computing Machinery, New York, NY, USA, 378–383. https://doi.org/10.1145/3159450.3159549

[8] Connectivity Standards Alliance. 2023. *Matter: The Foundation for Connected Things*. Retrieved August 1, 2023 from https://csa-iot.org/all-solutions/matter/

[9] Lorin W. Anderson and David R. Krathwohl. 2001. *A Taxonomy for Learning, Teaching, and Assessing*. Longman.

[10] Heltec Automation. 2023. *WiFi LoRa 32 (V3)*. Retrieved August 1, 2023 from https://heltec.org/project/wifi-lora-32-v3/

[11] Artur Balanuta, Nuno Pereira, Swarun Kumar, and Anthony Rowe. 2020. A cloud-optimized link layer for low-power wide-area networks. In *Proceedings of the 18th International Conference on Mobile Systems, Applications, and Services*. 247–259.

[12] Barry Burd, Lecia Barker, Monica Divitini, Felix Armando Fermin Perez, Ingrid Russell, Bill Siever, and Liviana Tudor. 2018. Courses, Content, and Tools for Internet of Things in Computer Science Education. In *Proceedings of the 2017 ITiCSE Conference on Working Group Reports* (Bologna, Italy) *(ITiCSE-WGR '17)*. Association for Computing Machinery, New York, NY, USA, 125–139. https://doi.org/10.1145/3174781.3174788

[13] Bradford Campbell and Nabeel Nasir. 2023. *CS/ECE4501 - Wireless for the Internet of Things , Spring 2023*. Retrieved November 1, 2023 from https://www.cs.virginia.edu/~bjc8c/class/cs4501-s23/

[14] Roger D. Chamberlain, James Orr, Doug Shook, and Bill Siever. 2022. Advancing Your Arduino Game: Early and Engaging Scaffolding for Advanced CS. In *Proceedings of the 53rd ACM Technical Symposium on Computer Science Education V. 2* (Providence, RI, USA) *(SIGCSE 2022)*. Association for Computing Machinery, New York, NY, USA, 1196. https://doi.org/10.1145/3478432.3499143

[15] Adwait Dongare, Revathy Narayanan, Akshay Gadre, Anh Luong, Artur Balanuta, Swarun Kumar, Bob Iannucci, and Anthony Rowe. 2018. Charm: exploiting geographical diversity through coherent combining in low-power wide-area networks. In *2018 17th ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN)*. IEEE, 60–71.

[16] Adam Dunkels, Bjorn Gronvall, and Thiemo Voigt. 2004. Contiki-a lightweight and flexible operating system for tiny networked sensors. In *29th annual IEEE international conference on local computer networks*. IEEE, 455–462.

[17] Iowa State University's Center for Excellence in Learning and Teaching. 2022. *Revised Bloom's Taxonomy*. Retrieved August 1, 2023 from https://www.celt.iastate.edu/instructional-strategies/effective-teaching-practices/revised-blooms-taxonomy/

[18] Anna Förster, Jens Dede, Andreas Könsgen, Asanga Udugama, and Idrees Zaman. 2017. Teaching the Internet of Things. *GetMobile: Mobile Comp. and Comm.* 20, 3 (jan 2017), 24–28. https://doi.org/10.1145/3036699.3036707

[19] Branden Ghena. 2023. *CS397/CS497 - Wireless Protocols for the Internet of Things*. Retrieved November 1, 2023 from https://brghena.github.io/courses/cs397/

[20] Mahanth Gowda. 2023. *CSE/EE 597: Wireless and mobile sensing in the age of IOT*. Retrieved August 1, 2023 from https://www.cse.psu.edu/~mkg31/teaching/cse_ee597/

[21] Haitham Hassanieh. 2023. *ECE 439 (Spring 2021): Wireless Networks*. Retrieved August 1, 2023 from https://courses.grainger.illinois.edu/ece439/sp2021/

[22] PlatformIO Labs. 2023. *PlatformIO*. Retrieved August 1, 2023 from https://platformio.org/

[23] Hanna Mäenpää, Sasu Tarkoma, Samu Varjonen, and Arto Vihavainen. 2015. Blending Problem- and Project-Based Learning in Internet of Things Education: Case Greenhouse Maintenance. In *Proceedings of the 46th ACM Technical Symposium on Computer Science Education* (Kansas City, Missouri, USA) *(SIGCSE '15)*. Association for Computing Machinery, New York, NY, USA, 398–403. https://doi.org/10.1145/2676723.2677262

[24] Microsoft. 2023. *Reimplement the Compare feature in Powerpoint for Mac*. Retrieved August 1, 2023 from https://answers.microsoft.com/en-us/msoffice/forum/all/reimplement-the-compare-feature-in-powerpoint-for/618682e9-c53f-4f48-a3c8-aee3e2d8ab16

[25] Microsoft. 2023. *Track changes in your presentation*. Retrieved August 1, 2023 from https://support.microsoft.com/en-gb/office/track-changes-in-your-presentation-35dad781-50f7-4c4f-9b15-cf418f03c279

[26] Microsoft. 2023. *Visual Studio Code*. Retrieved August 1, 2023 from https://code.visualstudio.com/

[27] Pat Pannuto. 2023. *CSE190/CSE291 - Wireless and Communication in the Internet of Things, Fall 2022*. Retrieved November 1, 2023 from https://patpannuto.com/classes/2022/fall/wireless-iot/

[28] Pat Pannuto. 2023. *CSE291 - Wireless and Communication in the Internet of Things, Winter 2022*. Retrieved November 1, 2023 from https://patpannuto.com/classes/2022/winter/cse291/

[29] Pat Pannuto. 2023. *WES 269 - Wireless and Communication in the Internet of Things, Winter 2023*. Retrieved November 1, 2023 from https://sites.google.com/eng.ucsd.edu/wi23-wes269/

[30] Raphael Schrader, Thomas Ax, Christof Röhrig, and Claus Fühner. 2016. Advertising power consumption of bluetooth low energy systems. In *2016 3rd International Symposium on Wireless Systems within the Conferences on Intelligent Data Acquisition and Advanced Computing Systems (IDAACS-SWS)*. IEEE, 62–68.

[31] Nordic Semiconductor. 2023. *nRF Connect for Mobile Mobile*. Retrieved August 1, 2023 from https://www.nordicsemi.com/Products/Development-tools/nrf-connect-for-mobile

[32] Nordic Semiconductor. 2023. *nRF Connect for VS Code*. Retrieved August 1, 2023 from https://www.nordicsemi.com/Products/Development-tools/nrf-connect-for-vs-code

[33] Nordic Semiconductor. 2023. *nRF Thread Topology Monitor*. Retrieved August 1, 2023 from https://www.nordicsemi.com/Products/Development-tools/nrf-thread-topology-monitor

[34] Nordic Semiconductor. 2023. *nRF52840 DK*. Retrieved August 1, 2023 from https://www.nordicsemi.com/Products/Development-hardware/nrf52840-dk

[35] Nordic Semiconductor. 2023. *nRF52840 Dongle*. Retrieved August 1, 2023 from https://www.nordicsemi.com/Products/Development-hardware/nrf52840-dongle

[36] Bill Siever, Roger D. Chamberlain, Elliott Forbes, and Ingrid Russell. 2019. Including Embedded Systems in CS: Why? When? And How?. In *Proceedings of the 50th ACM Technical Symposium on Computer Science Education* (Minneapolis, MN, USA) *(SIGCSE '19)*. Association for Computing Machinery, New York, NY, USA, 328–329. https://doi.org/10.1145/3287324.3287327

[37] Peter Steenkiste. 2023. *18-452/18-750: Wireless Networking and Applications*. Retrieved August 1, 2023 from https://www.cs.cmu.edu/~prs/wirelessS21/

[38] Espressif Systems. 2023. *ESP32 Wi-Fi and Bluetooth MCU*. Retrieved August 1, 2023 from https://www.espressif.com/en/products/socs/esp32

[39] a Linux Foundation Project Zephyr® Project. 2023. *Zephyr*. Retrieved August 1, 2023 from https://www.zephyrproject.org/