Applying Project-based Learning to Improve Computer Networks Courses: An Experience Report

Zheng Song, Nidhi Shah, Jinhua Guo, Qiang Zhu
Computer and Information Science Department
University of Michigan at Dearborn
Email: {zhesong, nidhii, jinhua, qzhu}@umich.com

***Abstract*—Project-based learning (PjBL) has been increasingly adopted in computer science courses to improve students' engagement and learning outcomes. Although a computer networks course is in great need of a PjBL course module, no such module is available due to the huge gap between PjBL's design requirements and the current structure and content of the course. This paper introduces a novel PjBL module for a computer networks course, which challenges the students with a real world problem of developing the communication system for a smart lock. Following the PjBL design principles, we devise several scaffolding activities and assignments, which can be integrated into a semester-long computer networks course. We test ran the PjBL module in both undergraduate- and graduate-level computer networks courses. Our preliminary evaluation results show that the proposed PjBL module is well received by the students and helps improve their learning outcomes.**

I. Introduction

A computer networks course builds a foundation for students to understand modern distributed computing systems, including cloud computing, edge computing, and the Internet of Things. Compared with other computer science courses, teaching computer networks is particularly challenging. The expected learning outcomes [1] include many theoretical concepts that are mostly based on technical documentation and standards (e.g., the network protocols), which are very abstract and a bit difficult to understand. Moreover, the theory and practice split of computer networks leans more on the side of the theory [2], making it hard for students to practice the course materials to enhance their understanding.

To improve the learning effectiveness of computer networks, many hands-on laboratory modules have been developed for students to apply the theoretical concepts needed to solve real world problems. These hands-on modules fall into the following categories: 1) capturing and analyzing data packets using Wireshark [3], [4]; 2) building a client/server application using socket programming [5], [6]; 3) tracing the data packets that are transmitted in the backbone network [7], [8], [9]; 4) configuring, testing, and troubleshooting a network of multiple virtual nodes [10]; 5) simulating dynamics of network protocols and analyzing their performance under different scenarios[11].

However, many of such laboratories widely used nowadays were designed 10 to 20 years ago. The technology stack of computer networks has evolved significantly, while many easy-to-use programming abstractions and network management tools have been developed. Students feel lack of motivation as it is hard to connect these laboratories to skills required by their future employment. Hence, an up-to-date, hands-on project is required to better motivate students and help them learn computer networks.

This paper reports how we designed and test ran a new course project following the project-based learning framework guidelines. **Pr**o**j**ect-**B**ased **L**earning (PjBL) is a pedagogical methodology through which students learn the subject by actively engaging in real-world and personally meaningful projects. Our project challenges students to design and implement a communication system for a smart lock, which has been increasingly used by IoT systems. Instead of reproducing an expected unique solution to a well-defined problem, students need to understand the lock's usage scenario and design their own communication system accordingly.

Although PjBL has been applied to many computer science courses, we have not seen a mature PjBL module for the entry-level computer networks course, as it is hard to strike the right balance between effectiveness and feasibility. Projects that follow the PjBL design guidelines to the hilt take a dedicated semester to finish and are unaffordable for most one-semester long network courses [12], [13]. In contrast, some other PjBL projects [14] are more likely for laboratories and are less effective in engaging students.

To solve this problem, our project design follows the PjBL principles and in the meantime accommodates time and logistics constraints. To fit the project into an entry-level computer networks course, we 1) pick an authentic problem for students but leave the design options open for their choices; 2) develop scaffolding resources to reduce student's learning curve, and 3) adjust the contents of lectures to align with the progress of the project. Furthermore, we design the scaffolding resources to work on both Raspberry Pis [15] and a virtual machine with Raspbian [16] system image so that students who do not have access to real IoT devices can still participate in the project. Our PjBL module design and supportive materials are released as an open-access package[1].

The rest of this paper is organized as follows. Section II introduces the background and related work. Section III describes our project design. Section IV introduces test runs of the proposed project, analyzes students' feedback, and

[1]available at: https://github.com/nidhibaid/ComputerNetworksPJBL

discusses lessons we learned in the procedure. Section V concludes the paper and outlines future directions.

II. Background and Related Work

In this section, we first introduce the current computer networks course curriculum, various teaching methods, and the project-based learning pedagogical framework. We then summarize the existing research work on PjBL for computer networks.

A. Computer Networks Course Curriculum

Computer networks have become a fundamental component of most information technology systems and the computer networks course is commonly included in computer science or engineering education [2]. The ACM/IEEE Computer Science Curricula [1] specifies the expected high-level learning objectives and low-level learning outcomes of the entry-level computer networks course. The high-level learning objectives include: 1) understand how the networks behave and the key principles behind the organization and operation of the networks; 2) serve as a starting point to more advanced network courses; 3) learn to use networking tools and write networked software. The detailed expected learning outcomes include knowledge about the layered network architecture and how each layer works.

Different from other computer science courses such as programming languages, web technologies, or database systems, the concepts taught in a computer networks course are relatively theoretical and abstract [2]. Students cannot directly relate the theoretical materials and the laboratory exercises to their personal experiences or skills required for future employments [2]. To solve this problem, many research studies attempt to improve students' motivation and interest by connecting the theory with students' own practical experience and facilitating active participation of students.

B. Teaching Methods in Computer Networks Courses

A recent survey paper [2] divides existing research studies on teaching computer networks courses into the following categories: 1) methods based on using visualization objects such as network simulators, multimedia applications, packet-tracing tools, or visual analogies; 2) methods based on using virtualization techniques; 3) methods precipitating active learning paradigm; and 4) methods based on practical hands-on laboratory exercises. Some work also noticed that with the continuous development and integration of Internet of Things technologies, students' interest in network technology is also constantly improving [17]. Although the mitigation of the theoretical lectures towards the practical laboratory experience is inevitable, there still lacks of a mature course module that can connect students' experiences with IoT devices to the learning objectives of a computer networks course and engage them in hands-on experiences.

C. Project-based Learning

Project based learning (PjBL) is a *student-centric* pedagogical strategy in which students apply the theoretical materials taught in a class to solve an *ill-defined* real world problem [18]. PjBL is increasingly adopted in undergraduate computer science courses and has been proven effective for students in attaining the learning objectives and developing the soft skills (e.g., communication, critical thinking, and team collaboration) for their employment.

PjBL has become one of the spotlight areas of active learning research in the past decades. Several project design principles have been widely acknowledged, known as the Gold Standard for project design and PjBL teaching practices [19]. We highlight the most important aspects of the Gold Standards:

1) Project Selection: The project needs to solve an authentic and meaningful problem at the appropriate level of challenge.
2) Students' Ownership: Students should feel ownership of their project by identifying their own project and making decisions about the design.
3) Feedback and Publicity: Students should share their progress and final outcomes in public and should be given periodical feedback for improvement.
4) Managing and Scaffolding Students' Learning Activities: Teachers should work with students to organize tasks and schedules, set checkpoints and deadlines, and provide instructional resources to guide and support students in reaching project goals.

D. PjBL for Computer Networks

Although PjBL has been applied to many CS courses, we cannot find a mature PjBL module for an entry-level computer networks course, as it is hard to strike the right balance between its effectiveness and workload. Following the PjBL design guidelines to the hilt [12], [13], a project takes a dedicated semester (around 150 hours) to finish, which is unaffordable for an entry-level network course. In the PjBL design introduced in [12], a group of students will spend the first 4 weeks developing a project proposal and the rest 10 weeks implementing their proposed project. To propose a meaningful project, the students need to understand computer networks knowledge to a certain extent, requiring them to have taken the entry level computer networks course as prerequisite.

In contrast, [14], [11] developed projects that are more likely to laboratories and are less effective engaging students. These projects include using network analyzer tools for network analysis and using a network simulator to learn network design and troubleshooting. Although the students' workloads are more manageable, such a design disregards the principles of PjBL and may be ineffective in engaging students.

E. IoT Protocols

Our PjBL module requires students to learn several popular application layer protocols for IoT environments, including Websocket[20], MQTT [21], and CoAP [22]. Websocket makes it possible for clients within a private network to communicate bidirectionally with a public-accessible server. Websocket maintains a long-live connection by periodically sending keep-live messages, thereby reducing its latency and incurring higher costs. MQTT and CoAP are designed for IoT environments and use binary header to reduce overhead. MQTT is based on TCP and follows the pub/sub pattern. CoAP is based on UDP and does not require a connection establish procedure. More details about the IoT protocols are provided in our scaffolding materials.

III. Design of PjBL Activities

We design a PjBL module, requiring a team of $3 \sim 4$ students to design and implement a communication system for a smart lock. Smart locks are increasingly used in smart homes and Industry 4.0 environments. A smart lock receives locking/unlocking instructions from an authorized device and performs the corresponding operations. It also monitors access and sends alerts for different events. The instructions, access logs, and event alerts are sent over the network between the lock and its users.

In the rest of this section, we first introduce the general design of our project, followed by the design challenges we met as well as our design choices. We then introduce the students activities in our PjBL module.

A. Project Design Overview

Each team is required to define their own usage scenario of the smart lock and to summarize the QoS requirements on its functionality. For example, a smart lock designed for smart homes requires the battery to last longer, but the user can endure a longer responding time (a few seconds); a smart lock designed for a rolling shutter door in a manufacturing plant requires faster responses, but are insensitive to energy consumption as it is usually connected to a power supply. In the former scenario, the QoS requirements for the communication system are: 1) its energy consumption should be extremely low; 2) its reliability can be relatively low (e.g., 2 nines, 99%); 3) its latency can be relatively high (less than 2 seconds). In the latter scenario, the QoS requirements are: 1) its energy consumption can be very high (e.g., five nines, 99.999%); 2) its reliability should be extremely high; 3) its latency should be less than 200 ms.

Each team is required to define their system components, pick the communication protocol they use (some options are: HTTP, WebSocket [20], MQTT [21], and CoAP [22]), and parameterize their communication system so that it can satisfy the QoS requirements of the usage scenario. Each team is required to meet with the instructor or the TA to discuss their usage scenario, QoS requirements, and their communication system design. During the discussion, they should specify how they design their system to meet the QoS requirements of the usage scenario.

The final stage of the project is for each team to implement their system design. Each team is required to implement the major functionalities of the communication system so that they can demonstrate how it works and measure its QoS.

B. Design Challenges and Choices

The gold standard for project based learning summarizes several guidelines for a project design. A project that follows these guidelines to the hilt will undoubtedly be more effective for students' learning. However, its applicability will also be hindered by many realistic factors. When designing our project, we tried our best to strike the right balance between the project's effectiveness and its feasibility.

Challenge 1: Student's Ownership. The gold standard specifies that "Students make some decisions about the project, including how they work and what they create, and express their own ideas in their own voice." Creating a sense of ownership helps students care more about the project and work harder [23]. In many project-based learning practices, students are required to propose a project of their own interest. For example, in a software modeling and design course, students are encouraged to propose software of their own interest. While students learn the theoretical knowledge of methods, tools, and processes related to designing a software system, they can practice these knowledge in their projects.

However, different from other computer science courses, it is hard for students who just started to learn computer networks to choose an appropriate project that can link to the course learning objectives. The course focuses more on abstract and theoretical technologies that build the foundation of modern networks, but these materials are a bit far away from students' daily life to make them feel attached.

Our design choice is to fix the topic of the project but leave the system's QoS requirements open for the students to decide. Students may connect their observations on how a real world usage scenario expects the smart lock to function to the project's learning objective of understanding different network protocols' impact on the system's performance.

Challenge 2: A Challenging Problem. PjBL's gold standard specifies that "the project is framed by a meaningful problem to be solved or a question to answer, at the appropriate level of challenge." For the students to carry out this project while studying the theoretical knowledge of computer networks, we find it very important to make sure that the students have an appropriate workload.

The project itself requires knowledge about IoT protocols, open source implementations of these protocols, and impact of different parameters on the system's QoS. If scaffolding materials are not given, it may take too long for students to learn them. On the other hand, if too many scaffolding materials are given, the students may feel that the project is just about re-producing the expected solution.

Our design choice is to pick a few IoT protocols and frameworks for the students. We select three most popular IoT protocols (WebSocket, MQTT, and CoAP) as well as their implementations (websocketd [24] for websocket, Paho [25] for MQTT, and CoAPthon [26] for CoAP). We first provide a 15 minutes video introduction to each protocol, including how to run them using the selected software libraries or executable. We then ask the students to explore how different features of a protocol can impact its QoS. A boilerplate project is given for each protocol to help students stay on track.

Challenge 3: Authenticity. PjBL's gold standard specifies that "the project involves real-world context, tasks and tools, quality standards, or impact, or the project speaks to personal concerns, interests, and issues in the students' lives."

The first problem we encounter towards satisfying these requirements is that not every student has personal experience with smart locks. Some students may not have any previous experience with IoT devices. To help students understand the usage scenario of smart locks and the requirements of its users, we collect some user comments from a smart lock product's page on Amazon. As shown in Fig.1.(a), the product received over 9000 ratings at the time we prepared the materials for the PjBL module. We then filtered the low ratings and manually selected some user's typical comments to demonstrate how the QoS of a network system may impact the user's experience.

Fig.1.(b)(c)(d) demonstrate the user's complaints on the low reliability, the long responding time, and the huge energy consumption of this product, respectively. We ask the students to summarize the comments related to each QoS characteristics. We expect to provide students a real world context for them to understand how different communication systems and applications require dissimilar QoS characteristics.

The second problem we encounter towards satisfying these requirements is that many students do not have an IoT device. Although we encourage students to purchase a Raspberry Pi or an Arduino [27], we also design all project activities in a way that students can work on them by emulating a Raspberry Pi system [16] on their personal computers. We will further discuss how working on virtual IoT devices impacts students' engagement and learning outcomes in Section IV.

C. *PjBL Activities*

Fig. 2 shows our PjBL activities and their timeline. These activities are described in detail as follows.

Activity 0: Project Launching. The course project is announced in the first week's lecture. The lecture starts from the wireless communication protocols that students are familiar with, like WiFi and Bluetooth. Then, the smart lock's structure is analyzed, including its mechanical structure and software communication system. A smart lock's Amazon production page is displayed to the students, as well as the problems identified in users comments.

A demonstration [28] of how to build an internet-connected, smartphone-controlled deadbolt actuator powered by a Raspberry Pi is also provided to students. The instruction demonstrates in great detail how to use a cloud-based MQTT server to relay messages between a mobile client and a smart lock. Performance bottlenecks of the system are analyzed.

A typical networked system is given for students reference. The system contains a smart lock, a gateway, and a client. The client communicates with the gateway, while the gateway communicates with the lock. The smart lock system should provide the following functionalities:

1) the lock sends heartbeats and operation logs to the gateway, to be reviewed by the user on the client.
2) Lock and unlock if given the correct password (permanent or temporary).
3) Activate or deactivate the temporary password (once used to unlock the door, the temporary password should be disabled automatically).
4) Send notifications if someone breaks the lock.
5) For user operations like 2) and 3), return operation results, and error messages to the client.

Activity 1: Testbed Setup

This activity is organized as a checkpoint lasting for 2 weeks. The purpose is to set up the environment in which the students would be working for the course project. The idea is to be able to send and receive messages from one device to another, which would eventually have a role in the system design of the smart lock network setup. Once these devices are set up successfully, the student should be able to capture packets that are sent to and received from the devices. Three devices that would be identified at the end of the checkpoint are: a mobile phone, a laptop/desktop (Windows/Mac), and a virtual machine.

Detailed instructions are given to students about how to setup a Raspberry Pi virtual machine and a XAMPP HTTP server on the students' PCs, as well as how to install and run Wireshark on the virtual machine. Students are asked to visit the HTTP server using their VM and their mobile phone, while measuring the network latency and packet size. Students are required to work individually and turn in an experiment report containing their measurements and screenshots.

In the first week's lecture, we introduce basic network concepts, including packet switching network, packet-based data transmission, packet size measurement, and end-to-end latency. We also introduce how to use Wireshark to capture packets and measure the packet size and latency.

Activity 2: Programming an HTTP Server

This activity is organized as a checkpoint lasting for 2 weeks. Students are required to write a simple HTTP server, which can handle HTTP GET requests sent from a browser. Upon receiving the requests, the server will first authenticate the user, by checking whether a correct password is contained in the request parameter. The server then locks/unlocks the door according to the requests, and send a response back to the client's browser. A python example of HTTP server is provided along with the assignment. Students are encouraged to append on the example, but can also choose to use the language they are familiar with.

Students work individually on the following tasks: 1) Accept GET requests to the given URIs (or say PATH): "lock", "unlock", and "viewLog". All accepted requests will be appended to the log file. For other URIs, return a 404 Error. A password will be sent as a URL parameter in the GET requests; 2) Compare the received password with a correct password hardcoded in your program. If the right password

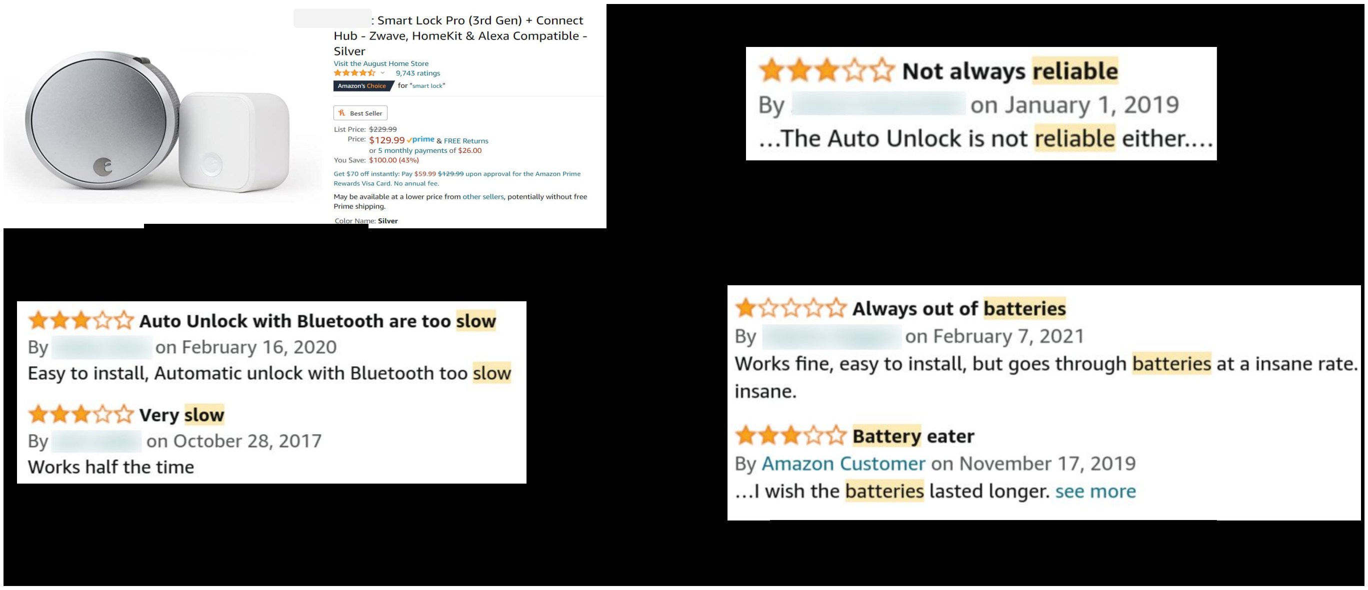

Fig. 1: Smart Locks' Amazon Comments

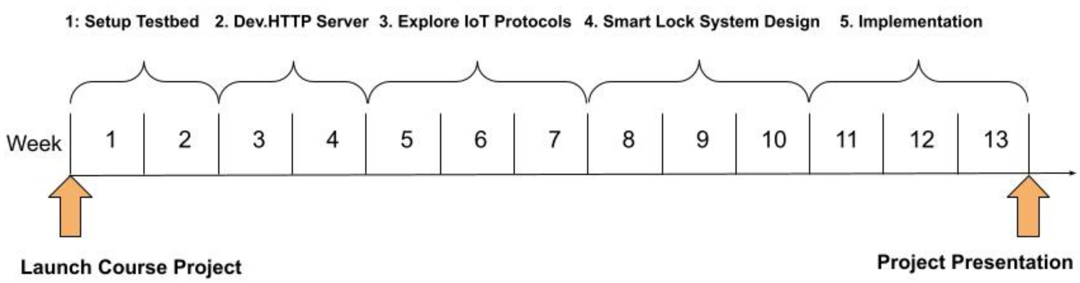

Fig. 2: PjBL Activity Timeline

is not provided, return a "401" error; 3) For authorized users: upon receiving a lock or unlock request, the server modifies the status of the lock file according, regardless of the file's current status. The server then returns 200 OK in the response status line and the content of the lock status file to be displayed on the client's browser; 4) For authorized users: the server returns 200 OK in the response status line and the content of the log file to be displayed on the client's browser.

Before launching this activity, we have introduced both application layer protocols (including HTTP) and socket programming in the lectures.

Activity 3: Exploring IoT Protocols

This activity is organized as a checkpoint lasting for 3 weeks. For each protocol, students are provided with a 15 minutes video, an instruction about how to setup a working environment, and a boilerplate project to understand how it works. Students are asked to form teams of 3-4 members and explore the features of IoT protocols by completing given tasks. They are also given a research paper about the QoS features of each protocol. Each team is asked to compare and contrast the features and performance of these protocols and submit a brief report.

For example, the students are asked to explore how a WebSocket server with a public IP address can push a message to its client with a private IP address. They are also asked to observe the packet exchange procedure between the server and the client, and report the size and latency of such a procedure. They are expected to find that 1) why WebSocket can be a good option if the gateway needs to push control messages to a lock behind a firewall; 2) WebSocket utilizes a long-live TCP connection to avoid establishing new connections for each message exchanges; 3) WebSocket's packet sizes are quite large compared to IoT protocols; 4) WebSocket needs to send keep-alive packets periodically, which can consume extra energy. Similarly, the tasks for CoAP are about resource discovery, confirmable and non-confirmable messages, and differences of methods (e.g., compare how "GET" is different from "Observe"); the tasks for MQTT are about different QoS levels, "Keep-alive" messages, and "Last Will Testament."

Before launching this activity, we have introduced the transport layer protocols (i.e., TCP and UDP) and their characteristics (connection establishment overhead, reliability, and header size) in the lectures.

Activity 4: System Design

This activity is organized as a checkpoint lasting for 3 weeks. Each team is asked to submit a design report to 1) specify the basic functional/non-functional requirements of their system; 2) introduce system/protocol design and the reasons behind their design choices; 3) divide the implementation workload among team members. The report should focus on how their communication procedure impacts the system's performance.

Activity 5: Final Presentation and Assessment Each team has 3 weeks to implement their designed system in the setup environment. After that, each team is asked to submit a 20 minutes video presenting the system design and implementation for smart locks. The presentation is required to at least cover the following aspects: 1) a brief introduction of their system design; 2) parts of the design that have been implemented; 3) demonstration of how their system works; 4) performance measurements of the system (e.g., latency, reliability, communication overhead, etc.).

Students are asked to review the presentations of other teams and give review feedback. Students' review will be counted towards the final grade each team receives. Members of each team are asked to report their individual effort in the team project and grade the performance of their teammates.

IV. Test Runs and Result Analysis

In this section, we report our experience of test running the proposed PjBL module from two computer networks courses at both undergraduate and graduate level in Fall 2020 and Winter 2021, respectively. Due to the COVID-19 pandemic, both classes were taught online with the final exam being changed to an online, open-book format. To evaluate the effectiveness of our PjBL module, we asked each student to submit a report regarding their experience with this project, as we were unable to directly compare students exam results with the previous semesters'. The focus of the report is to summarize the student's design, research, engineering, and communication efforts, as well as what the student learned from the course project.

The two test runs were conducted for the two courses, which features slight differences in their project modules. In the rest of this section, we will first introduce more details about the test runs, and then analyze students feedback.

A. Differences in Two Test Runs

Table I compares the differences in the two test runs. In fall 2020, 20 master's students enrolled in our graduate-level computer networks course. They were divided into 6 teams, with 3 $\sim$ 4 members in each team. The course project only had three stages: 1) Activity 0, project launching; 2) Activity 4, project design, and 3) Activity 5, final presentation and assessment. The instructor gave a 2 hours lecture introducing the basics of IoT protocols after launching the project. No boilerplate implementations were given, so students need to explore these protocols from scratch. No instructions on how to set up the Raspberry Pi virtual machine were given so each team was required to have at least one Raspberry Pi or Arduino. As it was a graduate level course, we required the students to implement some security features for their system.

In winter 2021, 29 undergraduate students enrolled in our undergraduate-level computer networks course, forming 9 teams of 3 $\sim$ 4 members. All the activities for the proposed PjBL module and scaffolding materials were provided for the course. Teams with 3 members were asked to explore 2 IoT protocols for Activity 3, while teams with 4 members were asked to explore all three IoT protocols. No security enhancement was required.

TABLE I: Differences in two Test Runs

Test Run	1	2
Student's Level	Master's	Undergrad
Enrollment	20	29
Number of Teams	6	9
Real Device Required	Yes	No
Boilerplate	No	Yes
Security Features Required	Yes	No
Activities Included	0, 4 and 5	All 6

B. Methodology

Each report was reviewed by two reviewers. After carefully reading each report, each reviewer individually assigned values to all characteristics of the report, which include: 1) student's overall opinion towards this project; 2) whether the student showed interest in this project; 3) the main area the student felt interested in; 4) indication of the student for the project to help learn computer networks knowledge; 5) satisfaction of the student with his/her team; 6) mentioning of the student regarding reading research papers or searching on the Internet for additional materials; 7) comparison of dissimilar protocols by the students; 8) indication of the student's security concerns and his/her efforts towards addressing these concerns; 9) mentioning how the team's design help optimize the system's performance. If the two reviewers held different opinions towards one report, a third reviewer was asked to help decide the final values.

C. Test Run Results

In fall 2020, five out of six teams of students chose to implement MQTT and the other chose HTTP. Figures 3 and 4 show the smart lock systems captured from the video presentations of team 5 and 6, respectively. Both teams used electric strikes to make a real smart lock.

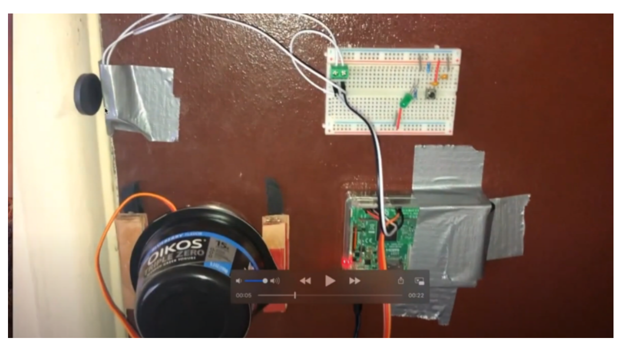

Fig. 3: Fall'20 Team 5 System Implementation

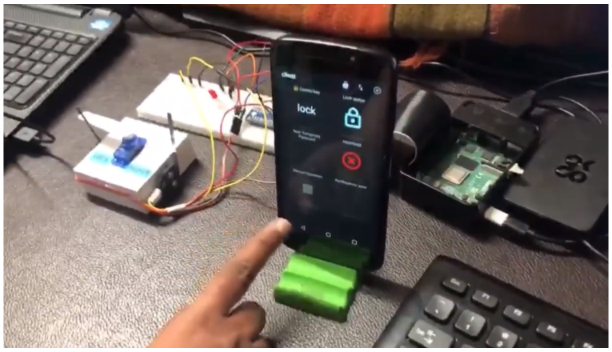

Fig. 4: Fall'20 Team 6 System Implementation

In Winter 2021, 3 teams chose CoAP, 1 team chose Websocket, 1 team chose HTTP, and the remaining 4 teams chose MQTT. All 9 teams implemented their smart lock systems on a Raspberry Pi virtual machine instead of a real device. Figures 5 and 6 are the screenshots captured from video presentations of team 1 and team 3, while the students were analyzing the system's performance using Wireshark.

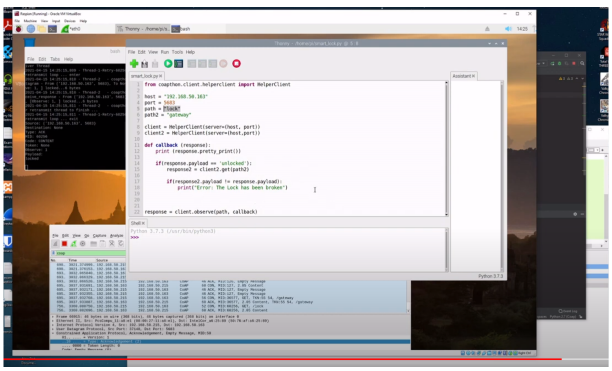

Fig. 5: Winter'21 Team 1 System Under Testing

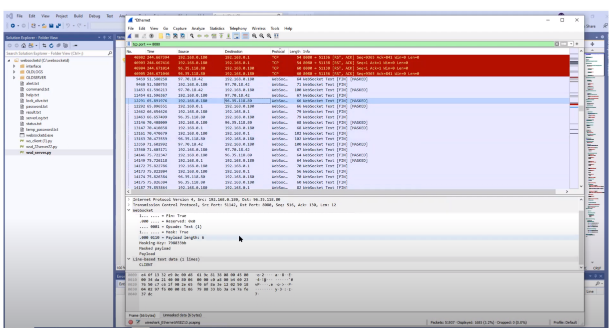

Fig. 6: Winter'21 Team 3 System Under Testing

Table II summarizes student's opinions revealed in their reports. From the table, we can observe that 1) 82.7% to 90% students gave overall positive feedback to the PjBL module; 2) 75.9% to 80% students felt interested in the project. However, students in the first test run were more interested in exploring IoT devices (40%), while students in the second test run were more interested in learning networking knowledge (37.9%); 3) More students in the second test run felt that the project helped them better learn network knowledge (89.7%), compared with 55% in the first test run; 4) More students in the first test run mentioned in their reports that they referred to research papers or online materials to understand the protocols better; 5) More students in the second test run mentioned that they gained better understanding of these protocols by comparing their features and impacts on QoS; 6) More students in the first test run were not satisfied with their teammate's work.

TABLE II: Students' Report Analysis in Two Test Runs

	# in TR1	Pct. 1	# in TR2	Pct. 2
Overall Opinion(positive)	18	90%	24	82.7%
Interested (positive)	16	80%	22	75.9%
Interested in IoT	8	40%	3	10.3%
Interested in Security	5	25%	2	6.9%
Interested in Network	4	20%	11	37.9%
Help Learn Network Know.	11	55%	26	89.7%
Unsatisfied with Teammates	5	25 %	1	3.4%
Extra Readings	8	40 %	2	6.9 %
Compare Protocols	3	15 %	16	55.2%

D. Result Interpretation

Many graduate students from Fall 2020 had experience working in the industry. They possessed practical problem-solving skills but lacked the implementation knowledge about computer networks. Most of them have had direct or indirect experience of working with applications that are enabled by protocols (e.g., HTTP). The reports from the students expressed their experience of learning the skills by implementing a project that is currently relevant. A student noted the exhilarating feeling of having implemented something small but working, "*It took several different tries to manipulate the data as I needed it to be. However, even on a novice example of working with protocols, it was extremely enlightening, and a fun experience to see it all work together.*" Other student expressed their rise in level of motivation to apply the skills accumulated from the course into their everyday interactions with networks and explore further options by writing, "*I appreciate this project's purpose and it has inspired me to investigate my own home's network security and empowered me to add a few smart objects (door and humidity sensors) to my own home, while keeping in mind good networking practices.*" Learning by doing is not just a theoretical concept, most students appreciated the project because they were able to have a tactile experience of the abstract concepts introduced in the lessons, "*The project provided me with a good opportunity to apply some of the concepts covered in class. I learned a lot about IOT systems and MQTT. I also learned more about network security*". Most of the class were engaged and were able to grasp the basic concepts of networking. The open-ended nature of the project encouraged the students to self-study which helped them to explore deeper into the subject. One student reported, "*As I was researching why MQTT was the best protocol for our project, I was extremely interested in the security considerations.*"

Despite its many benefits, there were a couple of setbacks. Some students reported dissatisfaction regarding the team members. The insufficient level of engagement and commitment by some team members made the overall learning of these students slightly different. Although they were individually able to complete the project, they had to work extra and, in some cases, did not implement all requirements as per the learning suggestions. This resulted in reduced practical learning. The other problem faced with having the project completely open-ended was that one team chose HTTP protocol for IoT application. Even though this was not the favored route, the students reported learning more from their unfavorable choice. Comparing their work in the end with that of their peers made them learn about the importance of developing the system design by considering the network performance factors such as latency and quality of service.

This run of the PjBL experiment was successful as the students were mature and were driven to learn and explore. The important take away from Fall 2020 was to introduce partially guided project and provide need-based assistance to the students.

In the second test run of the proposed PjBL module, the students were from undergraduate studies. They were mostly introduced to the concept of computer networking for the very first time. From the experience of the first test run, we were equipped with the facilities to deliver a successful online course and had spent the winter break to develop a partly guided project. The aim and requirements of the project remained the same. The two new things that were introduced were informational checkpoints and need-based assistance from the Tutor and the Teaching Assistant (TA). Through the checkpoints, students were given an open ground to experiment and still have them not lose the sight of what the aim of the project is by empowering them with the practical knowledge of the building blocks of the system design.

Most students felt uncomfortable at first, the two barriers were the coding language and the use of virtual machine. The students were in their learning phase and were comfortable with C++ or Java as these are the preferred languages for computer science courses at the university. For both these barriers, we provided individual sessions with TA and detailed instruction sheets for both Linux and Windows operating systems. Once the systems were set, the next challenge that was overcome was helping students debug and avoid infrastructural pitfalls. To achieve this we had weekly office hours (4 hours per week) and TA sessions (4 hours a week on average).

The checkpoints have immensely helped in steering the student's attention to the requirements of the project, that in software engineering terms being non-functional ones. Students engaged in conversations amongst their teams to understand which protocol is "faster" than the other, what Quality of Service is, why these protocols is different in terms of QoS, and how the differences in QoS make a difference. One student expressed himself in the following words: "*After comparing the factors that contribute to each protocols QoS, I learned that MQTT offers higher reliability than CoAP, but that it is also less efficient and has higher overhead.*" He goes on to say that "*Checkpoint 5 was also a very important learning experience for me because I was able to practically implement concepts I learned from in previous Checkpoints, while also gaining valuable experience working in a team environment.*" The success of having a partially guided project could also be observable from the level of motivation with which students ended their semester. One of the students expresses this in his report in the following words, "*It is pretty cool to now be able to construct possible ideas for my own home automation that can be specifically made for my needs. I'm also able to understand those needs better by doing this project because it taught me things I never knew about such as Quality of Service within programming.*"

While most students felt engaged, motivated, and enthusiastic about the project, some did feel overwhelmed and have reported neutral experience. They had hard time coping with the remote format and new materials. They did not feel comfortable in exploring things with minimum guidance and would have appreciated more traditional course format. Some students found the scope of the project somewhat intimidating.

V. Conclusion and Future Work

This paper reports how we designed and test ran a new course project following the project-based learning framework guidelines for an introductory computer networks course. Our project challenges students to design and implement a communication system for a smart lock. Our project design follows the PjBL principles and in the meantime accommodates time and logistics constraints. We test ran the PjBL module in computer networks courses at both the undergraduate and graduate levels. Based on the report analysis, we conclude that the proposed PjBL module indeed improves students' engagement and learning outcomes. Inspired by our test run experience, we suggest the following two directions for better applying PjBL in a computer networks course and similar computer science courses.

A. Adapting to online environment

The two test runs were both conducted when all classes were offered online due to the COVID pandemic. Several students complained about how their teammates did not show up in their regular meetings and failed to respond to emails on time. Some students might feel less committed to their teammates due to lack of in person communications. On the other hand, moving the project to online environments made it easier for part-time students who could not attend in person courses to participate in course projects.

Switching to the online mode caused logistics issues. For in person courses, it would be much easier to distribute IoT devices to the students and collect them at the end of a semester. Although we used a Raspberry Pi virtual machine as a substitute, it brought much heavier workload as we need to make sure all assignments and boilerplate programs work on both real devices and VMs. Besides, we pinpoint the main reason for lower interest in the second test run to lack of physical devices. Implementing the smart lock on real IoT devices makes the students feel they are working on a realistic problem and become more engaged.

Hence, one important future work direction is to study how to improve student commitment and engagement when projects are carried out fully online.

B. PjBL Scalability

The success of PjBL relies heavily on instructing projects in an ad-hoc manner, which negatively impacts its scalability. To give students timely guidance on how to effectively work as a team and connect theoretical knowledge to the practical problem on hand, instructors need to periodically check the progress of each project and understand their low-level details [29]. The low scalability of PjBL limits its applicability in large-size classes, including the blooming massive open online courses (MOOC).

Solving the scalability problem is particularly important for computer science courses, due to the complexity of CS course projects and the rapid increase of CS enrollment. Existing approaches tackle the problem by offloading the instructors' workload to other PjBL stakeholders (e.g., other students in the team [30] or industrial partners [31]), who have no expertise in education. Such solutions improve scalability for PjBL by sacrificing its effectiveness.

We envision autonomous instructing as the future direction for PjBL. We expect to leverage recent advances in artificial intelligence (AI) to extract domain knowledge from enormous archived projects guided by instructors, and instruct students accordingly, thereby ensuring both scalability and effectiveness. Although AI is already applied in intelligent tutoring systems (ITS) [32], [33] to assist teaching, realizing autonomous instructing for PjBL is still a non-trivial task as it is hard to analyze the multimodal students inputs, including design documents, software source codes, software testing results, and other course-specific outcomes.

VI. Acknowledgement

The authors want to thank Thomas Limbaugh for helping proofreading the assignments. This research is supported by NSF grant #2104337 and the Creative Learning Award from Hub for Teaching and Learning Resources at UM-Dearborn.

References

[1] S. Draft, "Computer science curricula 2013," *ACM and IEEE Computer Society, Incorporated: New York, NY, USA*, 2013.

[2] M. Prvan and J. OžEGOVIć, "Methods in teaching computer networks: a literature review," *ACM Transactions on Computing Education (TOCE)*, vol. 20, no. 3, pp. 1–35, 2020.

[3] V. Y. Hnatyshin and A. F. Lobo, "Undergraduate data communications and networking projects using opnet and wireshark software," in *Proceedings of the 39th SIGCSE technical symposium on Computer science education*, 2008, pp. 241–245.

[4] C. A. Chin, L. Sharma, and G. V. Crosby, "Evaluation of the effect of wireshark-based laboratories on increasing student understanding of learning outcomes in a data communications course," in *2013 ASEE Annual Conference & Exposition*, 2013, pp. 23–554.

[5] W. E. Toll, "Socket programming in the data communications laboratory," in *Proceedings of the twenty-sixth SIGCSE technical symposium on Computer science education*, 1995, pp. 39–43.

[6] G. Jayalaxmi, G. Suvarna, and C. Sujatha, "Outcome based approach to teaching, learning and assessment in computer networks: An experience," in *2012 IEEE International Conference on Engineering Education: Innovative Practices and Future Trends (AICERA)*. IEEE, 2012, pp. 1–6.

[7] W. F. Marti, U. W. Pooch, and J. A. Hamilton Jr, "Packet tracing: a new paradigm for teaching computer network courses," in *Proceedings of the 1st conference on Integrating technology into computer science education*, 1996, pp. 162–164.

[8] A. Wang and W. Jiang, "Research of teaching on network course based on ns-3," in *2009 First International Workshop on Education Technology and Computer Science*, vol. 2. IEEE, 2009, pp. 629–632.

[9] J. N. Matthews, "Hands-on approach to teaching computer networking using packet traces," in *Proceedings of the 6th conference on Information technology education*, 2005, pp. 361–367.

[10] M. Wannous and H. Nakano, "Nvlab, a networking virtual web-based laboratory that implements virtualization and virtual network computing technologies," *IEEE Transactions on Learning Technologies*, vol. 3, no. 2, pp. 129–138, 2009.

[11] J. Guo, W. Xiang, and S. Wang, "Reinforce networking theory with opnet simulation," *Journal of Information Technology Education: Research*, vol. 6, no. 1, pp. 215–226, 2007.

[12] F. Freitag, R. Tous Liesa, and L. Navarro Moldes, "Assessing the participatory design of a project-based course on computer network applications," in *CSEDU 2019: proceedings of the 11th International Conference on Computer Supported Education, volume 1, May 2-4, 2019, in Heraklion, Crete, Greece*. Scitepress, 2019, pp. 368–374.

[13] M. Vijayalakshmi and M. M. Raikar, "Development of network applications and services through project-based learning to meet 21 st century skills," in *2021 IEEE Global Engineering Education Conference (EDUCON)*. IEEE, 2021, pp. 1167–1174.

[14] J. Dong and H. Guo, "Enhance computer network curriculum using collaborative project-based learning," in *2011 ASEE Annual Conference & Exposition*, 2011, pp. 22–611.

[15] E. Upton and G. Halfacree, *Raspberry Pi user guide*. John Wiley & Sons, 2014.

[16] "Raspbian," https://www.raspbian.org/, accessed: 2021-09-30.

[17] L. Liu, Y. Zou, Y. Shi, L. Yang *et al.*, "Teaching reform of computer network course for engineering applications," *Advances in Educational Technology and Psychology*, vol. 2, no. 1, pp. 220–227, 2018.

[18] R. M. Capraro, M. M. Capraro, and J. R. Morgan, *STEM project-based learning: An integrated science, technology, engineering, and mathematics (STEM) approach*. Springer Science & Business Media, 2013.

[19] "What is project based learning," https://www.pblworks.org/what-is-pbl, accessed: 2021-09-30.

[20] "The websocket protocol," https://datatracker.ietf.org/doc/html/rfc6455, accessed: 2021-09-30.

[21] "Mqtt: The standard for iot messaging," https://mqtt.org/, accessed: 2021-09-30.

[22] "Coap," https://coap.technology/, accessed: 2021-09-30.

[23] J. Larmer, J. R. Mergendoller, and S. Boss, "Gold standard pbl: Essential project design elements," *Buck Institute for Education*, pp. 1–4, 2015.

[24] "Websockets the unix way," http://websocketd.com/, accessed: 2021-09-30.

[25] "Eclipse paho," https://www.eclipse.org/paho/, accessed: 2021-09-30.

[26] "Coapthon's github page," https://github.com/Tanganelli/CoAPthon, accessed: 2021-09-30.

[27] "Arduino - home," https://www.arduino.cc/, accessed: 2021-09-30.

[28] "Raspberry pi smart phone connected door lock," https://www.instructables.com/Raspberry-Pi-Smart-Phone-Connected-Door-Lock/, accessed: 2021-09-30.

[29] S. Mason, "Collaborative learning in computing education: Faculty perspectives and practices," in *ACM ICER'20*, 2020, pp. 136–146.

[30] T. Nguyen, M. Uto, Y. Abe, and M. Ueno, "Reliable peer assessment for team-project-based learning using item response theory," in *ICCE'15*, 2015, pp. 144–153.

[31] J. Whatley *et al.*, "Evaluation of a team project based learning module for developing employability skills," *Issues in Informing Science and Information Technology*, vol. 9, no. 1, pp. 75–89, 2012.

[32] T. Crow, A. Luxton-Reilly, and B. Wuensche, "Intelligent tutoring systems for programming education: a systematic review," in *ACE'18*, 2018, pp. 53–62.

[33] A. K. Erümit and İ. Çetin, "Design framework of adaptive intelligent tutoring systems," *Education and Information Technologies*, vol. 25, no. 5, pp. 4477–4500, 2020.