Demo: *BubbleNet*: Towards developing an IoT-based Physically Distant Classroom For Personal Bubbles

Brandon Purvis, Dheryta Jaisinghani, Sarah Diesburg
Department of Computer Science
University of Northern Iowa, USA
purvisb@uni.edu, dheryta@cs.uni.edu, diesburg@cs.uni.edu

Haroon Rashid Lone
Dept. of Computer Science
University of South Dakota, USA
haroon.lone@usd.edu

***Abstract*—The COVID-19 pandemic has presented unprecedented challenges across the world and universities are not saved either. Standard classroom activities in COVID era are even more challenging, with the primary challenge being ensuring physical distancing. We present a smart classroom system, *BubbleNet*, that attempts to relax these challenges. *BubbleNet* leverages cost-effective (∼ $30) IoT nodes with motion sensors. The IoT nodes collaborate with each other via OpenThread - the latest open-source mesh networking protocol released by Google. In this paper, we present the development and demonstration of *BubbleNet* for monitoring physical distancing rules in a classroom.**

***Index Terms*—COVID, mesh network, IoT, OpenThread**

I. Introduction

COVID-19 brought a set of unprecedented challenges for the world. To contain the virus, country-wide lockdowns were put into effect. The world abruptly transitioned from physical to virtual as people were encouraged to work from home. Universities started with having flipped classrooms [1]; however, as things progressed towards a pseudo-normal standard, universities adopted a hybrid model of education. In one example of this hybrid model, half of the class meets in person in the designated learning room and the other half remains on a virtual platform such as Zoom. The idea here is to ensure physical distancing with just 50% occupancy in the classroom. The problem with the approach is that it is hard to ensure restricted movement in the class such that the physical distancing norms are not violated [2].

Most state-of-the-art models for smart classrooms aim at delivering the learning content smartly [3]; in contrast, we are dealing with the problems particularly pertaining to assuring physical distancing during the COVID-19 pandemic. We aspire for a smart classroom that can raise alarms when the COVID security protocols are not met. Current approaches proposed to ensure physical distancing are either not scalable in real environments or they are costly and breach privacy. For example, authors in [4] propose to employ complicated deep learning strategies on video streams captured from camera to ensure physical distancing. Likewise authors in [5] propose a prototype with camera and Raspberry Pi to capture visuals and find distances between two people algorithmically. Another solution proposed in [6] mandates wearing of a wrist band that ultimately helps them to leverage ultra-wideband signals and employ two-way ranging calculations to ensure the mandatory six-feet distancing. While these approaches are innovative and impressive, their scalability, privacy and costs are a matter of concern.

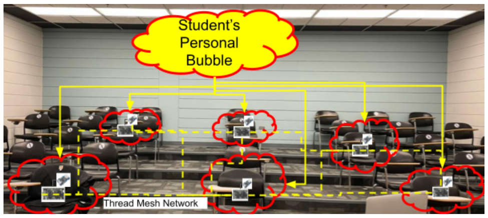
In contrast, we propose a novel cost-effective prototype that does not require any active participation from the end-users. Our system *BubbleNet*, proposes the idea of a student's personal bubble as shown in Figure 1a. *BubbleNet* leverages the power of OpenThread – the latest mesh networking protocol released by Google [7] to create personal bubbles for students who are seated exactly six-feet apart. It consists of thread IoT nodes that can sense motion in the classroom. The readings from these sensors are a reflection of how and when someone invades a student's personal bubble and thereby violating the rules of physical distancing. With the live dataset collected, the system can raise an alarm when the protocol of physical distancing is not ensured. These nodes send data over a dynamic mesh network to a central server By virtue of the OpenThread protocol. the network is highly robust to node failures, thus ensuring the reliability of our proof-of-concept. [8] Furthermore, we demonstrate it functioning in a real classroom that is occupied with 50% of student capacity.

Next, we proceed with explaining the System Details (Section II), the System Demonstration (Section III), followed by the Conclusion (Section IV).

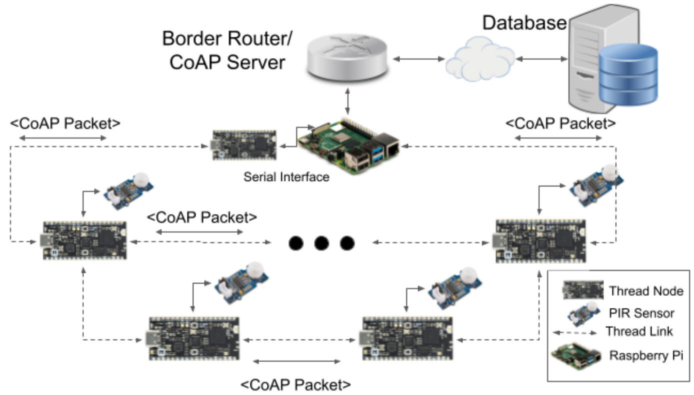
II. System Details

A. System Overview

The core idea behind *BubbleNet* is that each person lives in his/her personal bubble. We define a personal bubble as a space around a person within which he/she can comfortably sit or move while being six-feet distant from the peer. Our system ensures compliance with physical distancing guidelines by notifying the user when multiple people are detected within a bubble for a prolonged period of time. We demonstrate the idea in Figure. 1a. The classroom shown in the image has been partitioned spaces six+ feet apart for students to sit. Red bubbles here reflect personal bubbles of the students who take the seat. Each seat is equipped with the node we have developed and the nodes communicate over the `Thread` mesh protocol. It is expected that the teacher will move around the class for checking work assigned to the students, and by virtue of this movement, he/she can violate the rule of being physically distant. *BubbleNet* can check this violation



(a) *BubbleNet*'s System Overview. Red bubbles denote a person's personal area, and no one else should enter that to ensure the six-feet apart policy



(b) *BubbleNet*'s System Implementation. This is a representative system implementation. We show the placement of Thread nodes (Routers, End-Devices, Leader, Border Router), CoAP Server, PIR Sensor, and Thread Mesh Links.

Fig. 1: Demonstrating *BubbleNet*'s System Overview and System Implementation.

TABLE I: Device Details for Implementing *BubbleNet*

	Entity Name	**Hardware/Software Detail**
Hardware	Mesh Node	NRF52840 MDK Chip
	Thread Mode	RCP (Radio Co-Processor)
	Motion Sensor	SeedStudio Grove PIR Sensor
	Raspberry Pi	Ver. 3B+
Software	OpenThread	20191113-00534
	Database	SQLite 3.16.2
	CoAP Server	FreeCoAP

whenever a teacher enters a student's personal bubble, and the system can raise an alarm.

B. System Implementation

In this section, we explain the implementation details of *BubbleNet*. Table I lists the major hardware and software components of *BubbleNet* and Fig. 1b demonstrates the interconnection between these system components. We expand on each component next.

BubbleNet's core networking paradigm is based on the `Thread` protocol which is an IPv6 based mesh networking protocol designed for low power IoT nodes. Given the capabilities of `Thread`, it is widely-adopted protocol choice for IoT devices today. Major market players such as Samsung, Apple, and Google join the `Thread` consortium. `Thread` network does not have a single point of failure and roles of nodes can adapt to changes in the network. We leverage an open-source implementation of `Thread` released by Google called OpenThread for *BubbleNet*.

A `Thread` mesh network consists of the following device types – Border Routers, Routers, Router Eligible End Devices (REED) and Sleepy End Devices (SED). Since the `Thread` network primarily works with `IEEE 802.15.4`, the main purpose of a Border Router is to route traffic to external networks via WiFi or Ethernet. As the name suggests, Routers are the nodes that provide routing services to other network nodes. They may change their role to REED based on the needs of the network. REEDs are the devices that function as end-devices. They are capable of becoming a router when needed. As a REED, the device does not perform network routing. Lastly, SEDs are the minimal devices that can only communicate through a router. An end-device consumes lesser power than any other node in the network. One of the nodes from Border Router or Router can assume a role of a Network Leader. The leader manages the network by assigning addresses to other nodes and handling routers. In our setup, we have a Border Router, Routers, End-Devices, and one Leader. The nodes can be identified with a Routing Locator (RLOC) Address that is assigned to them by the network leader.

We use the `NRF52840` MDK chip from Nordic Semiconductors to develop our `Thread` nodes. These nodes operate in the Radio Co-Processor (RCP) mode. In RCP mode, the OpenThread stack (up to MAC layer) runs on the MDK chip itself. To monitor the motion in the classroom, we use a motion sensor from Seeed Studio named the Grove PIR sensor. It is a digital sensor and reports a value of 1 when there is a movement detected. The sensor is interfaced with the NRF52840 MDK chip over a GPIOTE driver that listens for changes in voltage on a specified pin.

All the nodes are connected in a mesh topology with one of them being a leader and one being the border router. The nodes in the network sense the motion and send the data to a central server via the border router. We leverage the Constrained Application Protocol (CoAP) for this data transmission. CoAP is an application layer protocol that uses restful model and is specifically designed to work in such low power networks. We implement CoAP server using an open source implementation named FreeCoAP [9]. In our setup, the server runs on the border router's host processor. However, the CoAP server can technically run anywhere on the Internet. The server reads CoAP packets and stores than in an SQLite database.

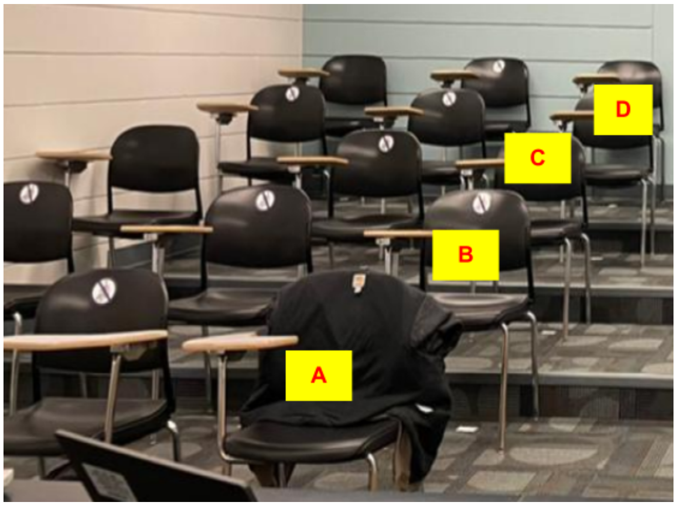


Fig. 2: Experimental setup for evaluating *BubbleNet*. Each number here is the RLOC address of the `Thread` node deployed.

TABLE II: Details of Sensor IDs during Experiment

Sensor ID	Detail
A	Student-Stationary
B	Empty Seat
C	Student-Stationary
D	Empty Seat

Algorithm 1: Classroom State Analysis

Assumption: Adjacent personal and buffer node
if *(CASE 1) no movement in either bubble over 2 minutes* **then**
  seats are unoccupied, situation ok;
end
else if *(CASE 2) movement detected in personal bubble* **then**
  seat is occupied, situation ok;
end
else if *(CASE 3) increased movement detected in personal bubble over 2 seconds AND movement on buffer bubble for any amount of time* **then**
  someone is standing too close to personal bubble, situation trouble;
end
else if *(CASE 4) increased movement detected in personal bubble under 2 seconds AND movement on buffer bubble over 2 seconds:* **then**
  someone has occupied the buffer bubble, situation trouble;
end
else if *(CASE 5) increased movement detected in personal bubble under 2 seconds AND movement on buffer bubble under 2 seconds* **then**
  someone is walking by, situation warning;
end

III. System Demonstration

We consider one column deployment of the `Thread` nodes in a live classroom that is configured to be physically distant with chairs being marked. The chairs that can be taken have appropriate marking. This setup is shown in Figure. 2. We identify each `Thread` node with an alphabetical letter, and we list the details of each of these nodes in Table II. A seat that can be occupied is referred to as a 'Personal Bubble' and a seat that cannot be occupied is referred to as a 'Buffer Bubble'.

We collect data with these four nodes deployed in the column. We noticed some recorded behaviors that allow us to make assumptions, namely:

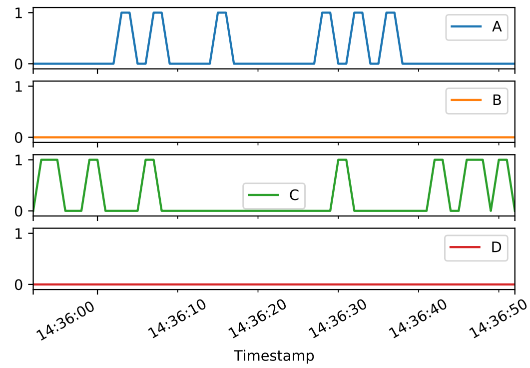
1) People make slight movements that are continuously picked up on our sensors. The longest amount of time our sensor did not pick up a very still person was one minute. Therefore, we make the assumption that people will make detected movements at least once every two minutes (double time to be safe). Conversely, we can assume that if no movement has been detected for over two minutes, the seat has become unoccupied.
2) When another person joins an already-seated person inside of a single personal bubble, the baseline movement detected increases.
3) We configure the radius of our personal bubble in relation to other bubbles in such a way that a seated person is only detected by one bubble, yet it is difficult for another person to stand by the seated student without being detected by the somewhat overlapping buffer bubbles (e.g. bubble radius of 5 feet spaced 6 feet apart).
4) People walking by bubbles in passing tend to clear a bubble's detection in under 2 seconds.

We test the efficacy of *BubbleNet* with the following five usecases –

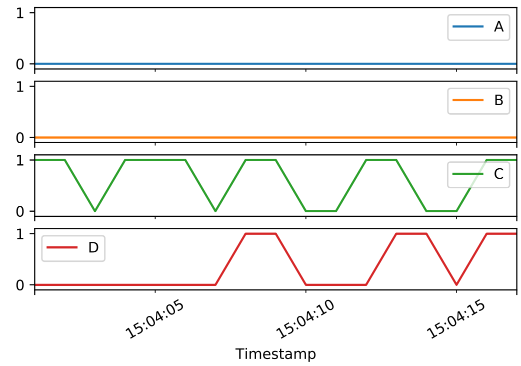
1) No one in the room. All seats unoccupied or all bubbles empty.
2) Students are in their assigned seats (personal bubbles) and stationary.
3) Someone is standing too close to a personal bubble.
4) Someone is occupying a buffer bubble.
5) Someone is walking by.

We process and detect each of these cases in software using the algorithm, Alg. 1.

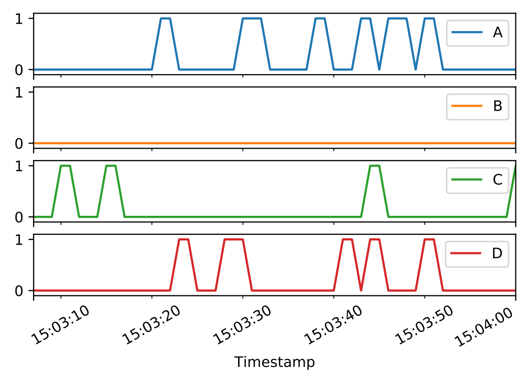
For each case listed above, we demonstrate a real classroom experience. We begin with Case 1 where no one is in the room. In this case none of the `Thread` nodes report activities. Next, students arrive in the classroom and they take their seats. Here we notice that node A and node C movement, as shown in Figure.3a. Moving forward, Case 3, Case 4 and Case 5 are imitating the behavior of teacher in the class. A teacher may move around to check how students are performing. Also, a teacher may sit or just stand still behind a student to closely



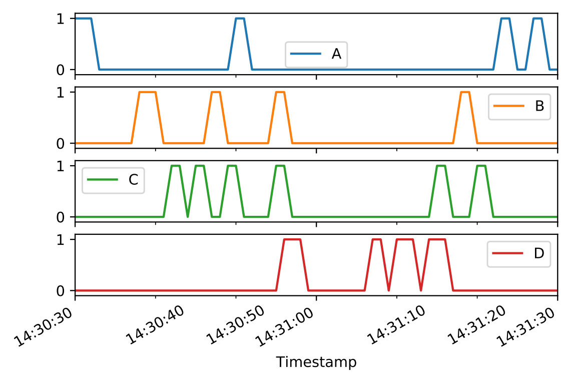
(a) Case 2 - Personal Bubbles Occupied (Bubble A and C)



(b) Case 3 - Someone too close to a Personal Bubble (Notice the activity in Buffer Bubble D)



(c) Case 4 - Someone has occupied Buffer Bubble (Notice the increased activity in Buffer Bubble D)



(d) Case 5 - Someone is Walking By (Notice the activities in all Bubbles)

Fig. 3: *BubbleNet*'s Evaluation (1 $\Longrightarrow$ Activity Detected, 0 $\Longrightarrow$ No Activity Detected)

monitor his/her work. We demonstrate each of these cases in Fig. 3b, Fig. 3c, and Fig. 3d, respectively.

Conference Presentation: We can replicate the exact same setup in one of the conference rooms and demonstrate how one person (a conference attendee) can invade other person's personal bubble thereby, violating physical distancing guidelines. We do not need any special accommodations for the purpose of this demonstration.

IV. Conclusion and Future Work

We proposed *BubbleNet* - a smart classroom system to aid in complying with COVID-19 physical distancing. We explained the development of the *BubbleNet* system and demonstrated the operation of *BubbleNet* in a live classroom. *BubbleNet* relies on a highly robust and dynamic thread mesh network. We plan to improve the fidelity of our assumptions through repeated testing and machine learning as an important future work. In addition, adding more sensors (such as heat, light, or sound) would likely improve the detection accuracy further.

Acknowledgment

Authors would like to thank Dr. Andrew Berns, Dept. of Computer Science, University of Northern Iowa, USA for helping us to solve various implementation related challenges faced during development of this system.

References

[1] B. Achée, "My CS1 Class Flipped over COVID-19," *Consortium for Computing Sciences in Colleges*, 2021.

[2] The Wall Street Journal, "Some schools have been open for months. here's what they learned." 2021, [Online; accessed 22-March-2021]. [Online]. Available: https://apple.news/AeZ306u9QS7m43UavIggPeQ

[3] M. K. Saini and N. Goel, "How Smart Are Smart Classrooms? A Review of Smart Classroom Technologies," *ACM Comput. Surv.*, 2019.

[4] N. S. Punn, S. K. Sonbhadra, and S. Agarwal, "Monitoring COVID-19 social distancing with person detection and tracking via fine-tuned YOLO v3 and Deepsort techniques," 2020.

[5] Y. Tachibana and N. Segawa, "Physical Distance Monitoring System for COVID-19 Using Raspberry Pi and a Monocular Camera: Poster Abstract," ser. SenSys '20. New York, NY, USA: Association for Computing Machinery, 2020.

[6] Y. Cao, A. Dhekne, and M. Ammar, *Physical Games in K-12 despite COVID-19*. Association for Computing Machinery, 2021.

[7] H. Kim, S. Kumar, and D. E. Culler, "Thread/OpenThread: A Compromise in Low-Power Wireless Multihop Network Architecture for the Internet of Things," *IEEE Communications Magazine*, 2019.

[8] B. Purvis, D. Jaisinghani, S. Diesburg, and H. Lone, "Demo:bubblenet: Towards developing an iot-basedphysically distant classroom for personal bubbles," in *IEEE Conference on Distributed Computing Systems (MASS)*, 2021.

[9] GitHub, "Free CoAP," 2021, [Online; accessed 22-March-2021]. [Online]. Available: https://github.com/keith-cullen/FreeCoAP