15 Years of IoT Education: a Shift from Theory to Practice

Antonio Boiano
DEIB - Politecnico di Milano, Italy
antonio.boiano@polimi.it

Fabio Palmese
DEIB - Politecnico di Milano, Italy
fabio.palmese@polimi.it

Alessandro E. C. Redondi
DEIB - Politecnico di Milano, Italy
alessandroenrico.redondi@polimi.it

Matteo Cesana
DEIB - Politecnico di Milano, Italy
matteo.cesana@polimi.it

Abstract

Over the past 15 years, the Internet of Things (IoT) course at Politecnico di Milano has undergone a significant transformation, from a theory-driven module on communication protocols to a practice-oriented experience focused on hands-on learning. Originally serving only students in Telecommunication and Computer Science Engineering, the course has expanded to serve a broader and more diverse audience, including Automation, Biomedical, and other engineering programs. This paper presents the evolution of the course structure, with a particular emphasis on the shift toward lab-based teaching, the redesign of evaluation methods, and the continuous integration of contemporary IoT technologies. A suite of seven carefully designed labs, ranging from simulated prototyping to protocol analysis and a capture-the-flag challenge, form the core of the course. Evaluation has shifted from traditional written exams to continuous assessment through practical assignments, though challenges remain in maintaining fairness and academic integrity, especially in the age of AI-assisted tools. Lessons learned from this evolution highlight the importance of adaptability, blended assessment strategies, and the ongoing need to align technical education with the dynamic IoT landscape.

CCS Concepts

• **Applied computing → Education**; • **Computer systems organization → Embedded systems**; • **Networks → Network protocols**.

Keywords

Internet of Things, Teaching, Education

1 Introduction

The Internet of Things (IoT) has rapidly transitioned from a niche research domain to a cornerstone of modern digital infrastructure, powering applications from smart homes to industrial automation. At the heart of any IoT system lies its communication layer—responsible for connecting constrained devices reliably, efficiently, and securely across diverse environments. As IoT technologies evolve, so too must the way we teach them, especially in engineering programs that aim to prepare students for real-world system design and deployment.

In 2010, Politecnico di Milano launched a dedicated IoT course as part of its Master of Science programs in Telecommunication and Computer Science Engineering. The course was designed to introduce students to the fundamental concepts of IoT systems, with a particular emphasis on the communication protocols and technologies that enable networked devices to interact. Initially structured as a theory-heavy course supported by a few targeted lab sessions, it focused on topics such as wireless propagation, energy modeling, and low-power communication protocols like IEEE 802.15.4 and Zigbee, with hands-on exercises using simulation environments provided by IoT frameworks such as TinyOS [6]. Evaluation was based on a traditional final written exam, aimed at assessing theoretical understanding.

In the last ten years, however, the course has undergone a significant transformation. What began as a lecture-centered curriculum has gradually evolved into a lab-driven, highly practical learning experience where students explore IoT communication technologies through guided practical exercises and simulations. In parallel, the method of student evaluation has shifted from classical exams to a continuous assessment model, based on practical in-course challenges, lab work, and take-home assignments. This approach encourages active participation and reflects the applied nature of the subject matter more effectively.

This paper reflects on that evolution, highlighting the motivations behind the pedagogical shift, the changing role of hands-on learning in IoT education, and the outcomes and lessons learned from fifteen years of teaching communication-centric IoT at Politecnico di Milano.

2 Laying the Foundations (2010–2016)

When the IoT course was first introduced at Politecnico di Milano in 2010, the Internet of Things was still an emerging concept in academic programs. Offered as a 5 European Credits (ECTS) course (equivalent to 50 hours of frontal lecture) within the Master of Science degrees in Telecommunication Engineering and Computer Science and Engineering, it was designed to provide students with a structured introduction to communication technologies for the IoT. The course attracted significant interest, with an average enrollment of more than 100s students per year.

The curriculum was structured to build from the ground up. It began with a conceptual introduction to the IoT, including a comparison with the traditional Internet, and a discussion of the unique constraints and requirements of IoT systems. This was followed by an exploration of hardware abstractions, where students learned about the typical architecture of an IoT "thing"—including sensors, actuators, microcontrollers, and radio interfaces. Particular emphasis was placed on energy consumption modeling, with analytical tools and examples illustrating how the behavior of the CPU and radio impacted overall device lifetime in battery-operated deployments.

A primer on wireless communication fundamentals provided essential background on signal propagation, path loss, and interference, preparing students to understand the limitations of low-power communication in real-world environments. This foundation enabled a deeper study of core IoT communication technologies, including IEEE 802.15.4, Zigbee, 6LoWPAN, and CoAP. These protocols were presented from both a theoretical and architectural perspective, with attention to their layering, interoperability, and suitability for constrained networks.

The didactic mix in this initial phase consisted of approximately 30 hours of theoretical lectures, 14 hours of numerical exercises (focused on link budget calculations, timing analysis, and protocol behavior), and just 6 hours of hands-on laboratory sessions. These labs introduced students to platforms like TinyOS and Contiki [3], enabling basic experiments with protocol stacks and low-power networking concepts. While limited in time and scope, the labs helped contextualize key ideas from lectures and exercises.

Student evaluation was conducted through a traditional written exam with open questions and numerical exercises, testing their understanding of communication models, protocol design, and system-level trade-offs. This structure provided a strong theoretical foundation but left limited space for creative exploration or system-level integration—an aspect that would become a focus in later iterations of the course.

Overall, the early years of the IoT course were characterized by a strong emphasis on foundational theory, with selected hands-on exercises serving to support and contextualize core communication concepts. This approach reflected both the maturity of IoT technologies at the time and the academic focus of the program. However, as IoT platforms and development tools rapidly evolved, and student expectations shifted toward more applied learning, this balance would soon begin to change.

3 From Protocols to Prototypes (2016–2022)

As IoT technologies matured and became more accessible, interest in the IoT course at Politecnico di Milano grew rapidly—both from students and other academic programs. Originally designed for Telecommunication and Computer Science Engineering students, the course was soon opened to a broader audience, including Automation Engineering, Biomedical Engineering, Music and Acoustic Engineering, and even non-traditional technical programs such as Cyber Risk Strategy and Governance. This diversification reflected the increasingly interdisciplinary nature of the IoT field and introduced new expectations in terms of teaching methods and course content. As an example, given the heterogeneous backgrounds of students, many basic networking concepts (e.g., HTTP) now need to be introduced to equalize foundational knowledge, while some of the more advanced theoretical details are omitted and instead left to practical experimentation.

With this expansion came a dramatic increase in enrollment, with the number of students peaking at almost 500 per year. To manage this scale, the course had to be split into two parallel sections, each handled by a different instructor and accommodating up to 250 students. This change brought not only logistical challenges but also new opportunities to rethink the delivery model and better align it with the needs of a diverse student population. Figure 1 shows the number of students enrolled in the course per year, grouped by degree program. As one can see, the two predominant groups are Computer Science (INF) and Telecommunication (TLC) engineers, with the former having a huge growth over the years. Starting from Academic Year 2019/20, the course was open also to Automation (AUT) engineering as well as Bioengineering (BIO), attracting considerable number of students also from these areas.

Pedagogically, this phase marked a decisive shift from a primarily theoretical approach to one focused on practical, system-level understanding. While core concepts like wireless propagation, protocol architectures, and network constraints remained central, the emphasis increasingly moved toward helping students build and experiment with complete IoT communication pipelines.

One of the most significant changes was the adjustment of the didactic mix: the number of hours allocated to numerical exercises was reduced, while time dedicated to laboratory activities was increased. Labs were no longer just illustrative

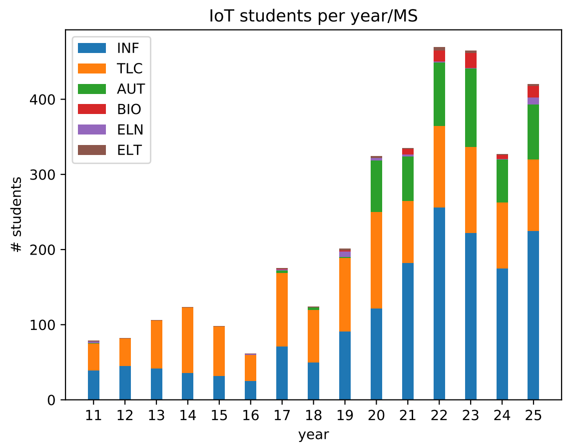

Figure 1: Number of students enrolled in the IoT course, grouped by degree program.

add-ons—they became core components of the course structure, designed to actively reinforce the material introduced in lectures.

New guided labs were introduced using platforms such as Node-RED [2] and ThingSpeak[4], which allowed students to design and simulate real-world IoT scenarios without requiring dedicated hardware. With Node-RED, students could build visual data flow applications, connect virtual sensors and actuators, and orchestrate cloud communication using protocols like MQTT and HTTP. Meanwhile, ThingSpeak provided an intuitive and open interface for collecting, visualizing, and analyzing time-series data in the cloud, making it easier for students to understand the end-to-end data lifecycle of IoT systems.

These labs emphasized application-level integration, helping students understand how communication protocols operate within broader system architectures. They also encouraged iterative development and debugging, echoing practices common in industry. Although hands-on work with physical hardware was still limited, the overall course design shifted clearly from being a course about protocols to a course about building with protocols, a distinction that better prepared students for the complex, integrated nature of real-world IoT solutions.

However, this shift also exposed limitations in the existing evaluation model. The traditional assessment format—centered around a final written exam with open-ended theoretical questions and protocol design exercises—increasingly struggled to reflect the new, practice-oriented nature of the course. As labs took on a more central role in the learning process, the written exam alone was no longer an adequate measure of student understanding or engagement.

To begin addressing this misalignment, instructors introduced a set of practical in-course challenges that students could complete for extra points. These assignments, often built around solving specific problems using lab tools or debugging simulated IoT setups, provided a more authentic and skill-based way to demonstrate learning. While these challenges were initially optional, they laid the groundwork for a deeper transformation of the evaluation approach that would take shape in the following years.

4 Labs at the Core (2022–2024)

The current iteration of the course features a refined didactic mix: approximately 26 hours of theoretical lectures, 14 hours of laboratory sessions, and 10 hours of numerical exercises. This balance enables students to engage with the material through multiple lenses, conceptual, analytical, and experimental, helping bridge the gap between abstract theory and real-world implementation.

The lecture component has been significantly updated to reflect emerging trends and technologies in the IoT landscape. Alongside foundational topics, students are now introduced to a wider range of communication protocols and architectures, including:

- Long-range technologies such as LoRa, NB-IoT, LTE Cat-M1, and 5G, with discussion on coverage, latency, and energy trade-offs.
- Short-range wireless standards like Bluetooth/BLE, Zigbee, Thread, and the Matter protocol stack for device interoperability.
- An introduction to fieldbus systems such as CAN bus, providing a bridge to Industrial IoT use cases and legacy integration scenarios.

This expanded scope helps students develop a broader understanding of the IoT protocol landscape, including how technologies differ in their assumptions, use cases, and deployment environments.

The lab suite has expanded to include seven structured sessions, each designed to reinforce specific layers of the IoT communication stack and promote system-level thinking:

(1) Hardware simulation: The first lab introduces students to Wokwi[1], a browser-based emulator for Arduino and ESP32 boards. Students implement basic sensor-actuator applications and experiment with GPIO control, interrupts, and timers. This environment enables them to quickly prototype device-side logic and understand the execution flow of microcontroller-based systems, without requiring physical hardware or dealing with low-level toolchain setup. Furthermore, students can experiment with device-to-device transmission using Wokwi's

[1]https://wokwi.com/

internal communication protocols. Additionally, they can test application-layer protocols such as MQTT, leveraging Wokwi's ability to connect the emulator to the public internet. This allows students to develop and debug IoT applications that span from local sensing to cloud integration, all within a fully virtualized environment.
(2) Energy consumption: In this session, students are introduced to measuring energy consumption of IoT devices using the INA226 current sensor connected to a real development board. They analyze the impact of CPU activity, radio transmission, and sleep modes on overall energy usage. The lab also introduces critical trade-offs between communication performance and power efficiency, a central concern in IoT system design.
(3) Application layer protocols: Students implement REST-style interactions using both HTTP and CoAP, exploring how these protocols function under constrained conditions. Through a series of experiments, including interactions with realistic HTTP and CoAP servers and traffic monitoring with Wireshark[2], students compare message overhead, latency, and reliability, gaining insight into protocol design decisions and their impact on real-world performance in lossy networks.
(4) MQTT: The MQTT lab introduces the publish/subscribe paradigm, where students build a lightweight IoT messaging pipeline using the Mosquitto MQTT broker [7] and simulated clients using the Eclipse PaHo library[3]. They implement topic-based filtering and test message delivery under different QoS levels. This session emphasizes scalable and decoupled communication patterns commonly used in commercial IoT systems.
(5) Node-RED prototyping: Expanding on the MQTT lab, students use Node-RED to visually program the logic of a complete IoT application. They simulate sensor input, process the data stream, and route information to local dashboards or cloud services such as ThingSpeak. This lab connects protocol-level knowledge with system-level orchestration and shows how low-code tools can accelerate development and testing.
(6) LoRa: This lab is dedicated to LoRa and LoRaWAN communication. Using Software-Defined Radio (SDR) tools, students visualize the effect of different spreading factors on signal bandwidth and range[8]. In parallel, they make use of the LoraSim simulator [1] to explore the impact of different parameters on the scalability performance of a LoRa network. Furthermore, the teacher demonstrates the deployment of simple applications on Arduino MKR WAN boards connected to The Things Network[5], reinforcing students' understanding of long-range, low-power wireless communication in real-world settings through observation of physical hardware interaction.
(7) CTF: The final lab, given at the end of the course, consists of a Capture-the-Flag activity composed of several hands-on challenges centered around IoT communication technologies. The current version includes six distinct challenges, spanning both low-level protocols such as Bluetooth Low Energy (BLE) and application-layer protocols like CoAP and MQTT. Each challenge is designed to assess students' understanding of theoretical concepts introduced in class (e.g., the structure and purpose of a BLE advertisement) as well as their ability to use practical tools covered in the previous lab sessions (e.g., connecting to and subscribing to an MQTT broker).

Together, these labs form the backbone of the course, serving as a practical complement to the theoretical lectures. Each lab is scheduled to follow the introduction of key concepts in class, allowing students to apply their knowledge in a structured, hands-on setting. This alignment supports a systems thinking approach, encouraging students to reason across layers—from hardware limitations to communication protocols and application behavior.

4.1 Evaluation

In parallel, the course now adopts a continuous assessment model, replacing the traditional final exam with a combination of lab deliverables, in-course challenges, and take-home assignments. This format not only promotes sustained engagement throughout the semester but also better reflects the iterative, problem-solving nature of IoT development. The result is a learning experience that is active, contextualized, and aligned with the skills required in modern IoT engineering.

As an example, the following four deliverables were requested to students during the current iteration of the course:

(1) Wokwi and energy consumption: Students are presented with a realistic IoT scenario, such as a parking lot monitoring system using proximity sensors, and tasked with developing a prototype using Wokwi. The implementation must meet predefined functional and energy efficiency requirements, such as incorporating deep sleep modes to reduce power consumption. In addition to the simulated implementation, students are provided with current consumption traces collected from real hardware. Based on these traces, they must analyze the energy profile of their prototype and estimate its expected operational lifetime under given conditions. Finally, they need to comment on possible further energy optimizations of

[2] https://www.wireshark.org/
[3] https://eclipse.dev/paho/

the system as well as solving a numerical exercise on the optimization of the transmission parameters.

(2) CoAP and MQTT: Students receive a set of packet capture (PCAP) files containing traffic traces for both CoAP and MQTT protocols. Their first task is to analyze the traces and identify key characteristics and behaviors of each protocol—such as request/response patterns, publish/subscribe flows, and session handling. Building on this, students are presented with a realistic IoT use case (e.g., smart valve monitoring and control) and asked to evaluate the implications of protocol design choices on energy consumption. This includes comparing different configurations, such as CoAP's observable mode and various QoS levels in MQTT, and discussing their impact on transmission frequency, message overhead, and device power usage.

(3) Node-RED and LoRa/LoRaWAN: This assignment is divided into two parts. In the first, students are required to implement a prototype using Node-RED, focusing on a typical cloud-side IoT workflow. The task includes reading input data from a CSV file, publishing and subscribing to MQTT topics, processing JSON payloads, and visualizing data through plots or dashboards. The goal is to familiarize students with event-driven programming and cloud integration using a visual, flow-based development environment. The second part centers on LoRa/LoRaWAN. Students complete a numerical exercise to compute collision probabilities in typical LoRa deployments, gaining insight into scalability limitations of the physical layer. They also use the LoRaSim simulator to explore the behavior of large-scale LoRa networks, analyzing the impact of parameters such as spreading factor distribution, network density, and traffic patterns on performance and reliability.

(4) System design: The final deliverable is a comprehensive open-ended design exercise in which students are tasked with proposing a complete IoT system architecture for a given application scenario. This includes selecting appropriate hardware platforms, communication protocols, and defining the system's operational behavior. Students must justify their design decisions, present a block diagram illustrating key components and data flows, and write pseudocode describing the main functional logic of the system.

    In addition to the design task, students are also assigned a set of traditional numerical exercises, reinforcing foundational topics such as link budget analysis, message delay estimation, and energy consumption modeling. This combination of open design and analytical rigor aims to synthesize all learning objectives of the course.

Students are allowed to work individually or in pairs when completing the challenges, fostering collaboration while maintaining manageable group sizes. While the course emphasizes continuous assessment through practical assignments, students retain the option to take a traditional final written exam as an alternative evaluation path. In the most recent iteration of the course, over 70% of students successfully passed the exam through the challenge-based assessment route, confirming both the feasibility and effectiveness of this evaluation approach.

To successfully pass the course, all assigned challenges must be completed and positively evaluated. Although it is acknowledged that students may utilize AI-based tools, including large language models (LLMs), to support their work, the diversity of the deliverables—ranging from numerical analyses to system diagrams, pseudocode, and protocol trace interpretation—helps instructors evaluate the depth of students' understanding and detect superficial or unsupported submissions.

That said, grading remains a significant workload, especially given the size of the class and the qualitative nature of many of the submissions. Nevertheless, the approach has proven effective in assessing practical competencies in a nuanced and realistic manner, while aligning closely with the evolving nature of IoT engineering practice

5 Lessons learned and future directions

Over more than a decade of evolution, the IoT course at Politecnico di Milano has undergone significant changes in content, structure, and pedagogy. This long-term transformation has offered several important insights into teaching communication technologies in a field that is both technically complex and rapidly evolving.

- Student engagement and satisfaction are closely linked to practical learning. A clear takeaway from the course's evolution is that the number and quality of practical activities strongly correlate with student satisfaction. As the course moved from a lecture-heavy structure toward one centered around guided labs, simulations, and hands-on challenges, both formal evaluations and informal feedback from students have shown a marked improvement. Students consistently report that they value the opportunity to "learn by doing" and that lab-based learning helps them connect theoretical concepts to real-world applications.
- Assessment continues to be one of the most complex aspects of the course. As the curriculum has shifted toward a more hands-on, applied format, ensuring fair and meaningful evaluation has become increasingly challenging—particularly in the context of large cohorts and the widespread availability of AI tools, such as large language

models (LLMs), which can assist students with code generation, debugging, and analysis.

The traditional written exam, once central to the course, has been made optional and may now be entirely replaced by continuous assessment through structured challenges and lab deliverables. However, verifying the authenticity and depth of student understanding remains difficult. Oral exams, while effective, are impractical given class sizes that can reach several hundred students.

This reality underscores the need for innovative evaluation strategies—potentially combining automated testing frameworks, individualized or adaptive assignments, and peer review mechanisms—to maintain academic integrity while preserving scalability and educational value.

- The field's pace demands continual adaptation. One of the defining characteristics of IoT as a subject area is the fluidity of standards and technologies. Protocols rise and fall in relevance rapidly, and new frameworks emerge every year. To remain current and useful, the course content has required regular updates, often on a yearly basis. This has included the introduction of newer protocols like Matter, Thread, NB-IoT, and BLE, as well as the inclusion of Industrial IoT concepts. However, this continuous adaptation has also required difficult choices: older topics (such as legacy stacks or outdated OS abstractions) have had to be compressed or removed to make space for more relevant material. This balancing act—between depth and breadth, between foundational knowledge and current practice—remains a core instructional challenge.

6 Conclusion

The evolution of the IoT course at Politecnico di Milano over the past decade reflects a broader shift in engineering education—from theory-centric instruction toward experiential, systems-level learning. Starting from a traditional lecture format focused on communication technologies, the course has progressively integrated hands-on labs, simulation tools, and challenge-based learning to better align with the skills demanded in today's IoT landscape.

This transition has not only improved student engagement and satisfaction, but has also highlighted critical challenges related to scalability, assessment, and curriculum agility. In particular, the need to evaluate practical skills at scale, while accounting for the rapid pace of technological change and the impact of AI-assisted tools, remains a moving target. At the same time, the course's iterative design and responsiveness to student needs have proven essential in maintaining its relevance and impact.

Ultimately, the experience illustrates that teaching IoT communication technologies today requires more than delivering technical content: it demands ongoing adaptation, thoughtful integration of tools and platforms, and a teaching philosophy that embraces uncertainty and innovation. The lessons learned from this journey may offer a roadmap for other educators seeking to redesign technical courses in similarly dynamic domains.

References

[1] Martin C Bor, Utz Roedig, Thiemo Voigt, and Juan M Alonso. 2016. Do LoRa low-power wide-area networks scale?. In *Proceedings of the 19th ACM international conference on modeling, analysis and simulation of wireless and mobile systems*. 59–67.

[2] OpenJS Foundation & Contributors. 2025. Node-RED. https://nodered.org.

[3] 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.

[4] The MathWorks Inc. 2025. ThingSpeak. https://thingspeak.mathworks.com.

[5] The Things Industries. 2025. The Things Network. https://www.thethingsnetwork.org/.

[6] Philip Levis, Samuel Madden, Joseph Polastre, Robert Szewczyk, Kamin Whitehouse, Alec Woo, David Gay, Jason Hill, Matt Welsh, Eric Brewer, et al. 2005. TinyOS: An operating system for sensor networks. *Ambient intelligence* (2005), 115–148.

[7] Roger A Light. 2017. Mosquitto: server and client implementation of the MQTT protocol. *Journal of Open Source Software* 2, 13 (2017), 265.

[8] Lorenzo Vangelista. 2017. Frequency shift chirp modulation: The LoRa modulation. *IEEE signal processing letters* 24, 12 (2017), 1818–1821.