POWVER

Technical Report 2021-14

Title: **Revising Computer Science Networking Hands-On Courses in the Context of the Future Internet**

Authors: Juan A. Fraire, Juan Eduardo Duran

Report Number: 2021-14

ERC Project: Power to the People. Verified.

ERC Project ID: 695614

Funded Under: H2020-EU.1.1. – EXCELLENT SCIENCE

Host Institution: Universität des Saarlandes, Dependable Systems and Software
Saarland Informatics Campus

Published In: IEEE Transactions on Education 64 (2)

This report contains an author-generated version of a publication in IEEE Transactions on Education 64 (2).

Please cite this publication as follows:

Juan A. Fraire, Juan Eduardo Duran.
Revising Computer Science Networking Hands-On Courses in the Context of the Future Internet.
Fraire, Juan A., and Juan E. Durán. "Revising Computer Science Networking Hands-On Courses in the Context of the Future Internet." IEEE Transactions on Education 64 (2), May 2021, 133-138.

POWER TO THE PEOPLE.
VERIFIED.
erc

Revising Computer Science Networking Hands-On Courses in the Context of the Future Internet

Juan A. Fraire*†‡, Juan E. Durán*,
*Facultad de Matemática, Astronomía, Física y Computación (FAMAF), UNC, Córdoba, Argentina
†CONICET, Instituto de Estudios Avanzados en Ingeniería y Tecnología (IDIT), Córdoba, Argentina
‡Saarland University, Saarland Informatics Campus, Saarbrücken, Germany

***Abstract*—*Contribution:* In a context where hands-on courses are biased towards specific technologies, a novel creativity-provoking instructional approach for networking undergraduate courses is successfully applied following action research principles and active and creative learning techniques.**

***Background:* Extensive engineering-oriented networking courses have been proposed with a strong focus on specific protocol solutions. At the same time, the amount and complexity of techniques is notably increasing with the advent of the Future Internet. As a result, the curricula looses focus on the fundamentals of networking algorithms.**

***Intended Outcomes:* We address algorithmic learning in networking for computer sciences, where students are expected to (*a*) create, (*b*) develop, (*c*) analyze and (*d*) compare algorithms and processes regardless of protocol-specific technologies. At least 70% of the students are expected to meet this goal while enhancing their engagement and motivation in a time-constrained course schedule.**

***Application Design:* To achieve (*a*)-(*b*), we instrument an active experimental strategy, while objectives (*c*)-(*d*) are tackled with creative learning techniques, both applied in an action research framework. The approach is supported by state-of-the-art networking application interfaces and simulators. Furthermore, a blended and game learning component favors the engagement via comparison and competition of students' project metrics.**

***Findings:* The experiment is carried out by professors of the Computer Science Bachelor's degree taught in FAMAF. Results show that the applied methodology met the intended outcomes, and improved by 7% in a two-year cycle. Furthermore, the approach was very well-received based on student's feedback.**

***Index Terms*—Computer Science; Networking; Future Internet**

I. INTRODUCTION

Current generations depends more and more on the digital world, significantly increasing the interest on information, communication and networking technologies now encompassed in the so-called Future Internet context [1]. Networking is not only a well established research area in computer science, but also a highly demanded skill by the industry. Indeed, engineering and computer science curricula need to offer highly effective networking courses in order to correctly train professionals who can design, maintain and secure wired and wireless network environments. Nonetheless, it is very hard for students to gain intuition on dynamic properties of computer networks, protocol exchanges and packet flows in classroom sessions or in exercises on paper [2].

To overcome these challenges, courses have considered hands-on *laboratories* comprising networking tools, equipment and proprietary software [3]–[7]. While these might be justified in specialized postgraduate courses, they tend to result difficult to manage in undergraduate courses as well as expensive, vendor-biased and technology-specific, which render them obsolete in very short time. However, emerging Future Internet networking concepts and solutions such as Internet of Things (IoT) Information Centric Networking (ICN) [8], Software Defined Networking (SDN) [9], Delay-Tolerant Networking (DTN) [10], Networks-on-Chip (NoC) [11] among others are already demanding a more flexible (non-technology specific) and dynamic (long-lasting) instructional approach.

In this paper, we argue that efficient undergraduate networking courses should guarantee a sustainable and stable utility by focusing on the fundamentals concepts of networking, while selectively mapping them to always changing technological solutions. Also, freely available supporting software should be on the spot. The objective of this work is to propose and evaluate a long-lasting instructional approach for network protocol development and analysis that is not technology specific and based on free tools. To this end, we leverage state-of-the-art active [12], experimental [13] and creative [14] learning to cover algorithmic based on networking application interfaces and simulators coded by students [15]. At least 70% of the students are expected to incorporate a protocol-agnostic know-how as well as the ability to map it to current and future protocols or applications.

To apply the approach, we follow an action research methodology [16], [17] to adapt an existing curricula for the Computer Networks and Distributed Systems (CNDS) course of the Computer Science degree, taught in Faculty of Mathematics, Astronomy, Physics and Computing (FAMAF) of the Universidad Nacional de Córdoba in Argentina. Results are summarized after a two-year experimental cycle concluded in 2019, where new pedagogical techniques and specific content axes were designed. Since existing courses follow similar approaches than the original CNDS [18], the outcomes of this case study are a valuable asset to evolve them into a novel creative-provoking networking hands-on course. Grading strategy, experiences, lessons learned and student feedback are thoroughly discussed and analyzed.

The remaining of this paper is organized based on the elapsed action research stages followed in two cycles in 2018 and in 2019: planning in Section II, design in Section III, action in Section IV, and conclusions in Section VI.

II. Planning

Context. The computer science bachelor degree at FAMAF focuses on both theoretical and technological aspects of computing. All third year students of computer science have to take part in the CNDS course during the first semester, which consists of: (i) **Lectures and practice:** four hours per week of theory and exercises on paper. (ii) **Laboratory hands-on:** four hours per week of hands-on practice in front of the computer. Students' have to pass Operating Systems and Algorithms and Data Structure courses to apply for CNDS course and exams. Student's performance on these subjects constitutes key background analyzed in Section V. CNDS exams consist of two partial exams based on the lectures and practices and two laboratory evaluations as a prerequisite to take the final exam in a so-called *regular* condition. If all partial exams and evaluations are graded higher than 6 (out of 10) and average at least 7, the student is considered *promoted* and approves the CNDS course. Students that are not regular nor promoted, can either retry the course in the next year or register to give an *open* final exam, which is more demanding than the regular final exam. Thus, students tend to exhibit notable motivation to obtain a promotion, but the original instructional approach used until 2017 failed to correctly exploit it.

Diagnosis. Prior to 2017, the CNDS lecture part was strictly based on the content flow suggested by the Tanenbaum book [19], which explores the layered network model from the physical layer up to the application layer. However, the laboratory part was biased towards the application layer, as most of the students' effort was aimed at writing network-based client/server Python scripts using the *sockets* library. The programs were only evaluated in terms of code correctness and efficiency. Similar networking courses face the same issue: the classroom becomes a chance to develop Python skills instead of learning networking concepts. Indeed, an oral survey performed in 2017 showed that, although discussed in the theoretical lectures, 70% of the graded students were not able to express by their own words the very fundamentals of networking, namely (*i*) congestion and flow control, (*ii*) routing and (*iii*) medium access control (layers 4, 3 and 2).

Research. The objective of this work is to research on pedagogic strategies and adequate curricula to concentrate the coding effort in reasoning about modern protocols in the context of Future Internet. In particular, we expect at least 70% of the students to (*a*) create (define requirements and design algorithms), (*b*) develop (choose architecture and implement), (*c*) analyze (methodologically observe, interpret and explain) and (*d*) compare (identify advantages and disadvantages) algorithms and process for each these items. Items (*a*) and (*b*) can leverage the Python programming expectations from CNDS, while (*c*) and (*d*) requires a novel approach to support students' comprehension of fundamental networking. The driving research questions are: can such an improvement of CNDS be applied while keeping an attainment rate of 70%? Is an action research framework suitable for this evolution?

Background. Approaches for specific hands-on wireless networks are plenty [3]–[7], but fail to cover fundamental algorithmic creation and analysis. Research in [20] tackles the analysis and develop part, but ignore the creative aspect. Authors in [21] deal creative challenges for security, but not for fundamentals of general networking protocols. To the best of authors knowledge, there is no previous work simultaneously tackling (*a*)-(*d*) in a hands-on networking course.

III. Design

Content. Content-wise, the down-top approach in Tanenbaum's book [19] was replaced by a top-down one as suggested by modern networking books such as Kurose's text [22]. At the same time, some well-prepared material from the 2017 edition of the class based on different sources [23]–[25] was also recovered and leveraged from a top-down perspective. A set of 4 laboratories activities (Lab 1 to 4), detailed in Table I were derived:

Lab1 and 2 are part of a development stage where the challenge is to develop efficient coding skills to provide correct socket-based network applications. These activities, which corresponds with Chapter 2 of Kurose's book covers objectives (*a*)-(*b*) by using Python. Specifically, Lab1 is used as an introductory activity where students are conducted through a first interaction with Python to retrieve a HTTP-compliant web-page via sockets. HTTP/2 protocol is discussed as Future Internet protocol and compared with HTTP/1. Lab2 further increases the difficulty as students have to develop a full network server based in a home-made file transfer protocol which can interact with a slightly modified client from Lab1. Based on provided skeleton code, students are requested to deliver the final code in a repository with a brief report on the difficulties and architectural decisions. Code quality and integrity is the main factor for grading this stage.

Labs 3 and 4 are introduced to accomplish objectives (*c*)-(*d*) while aiming at developing creative and analytical skills applied to transport, network and link layer networking algorithms. Because of the time constraints of the course, it is impossible to implement such algorithms in kernel space or low-level firmware/drivers such as proposed in highly specific courses [7]. Thus, similarly to [26], Omnet++ [27], a free C++

Table I
Content Design

List of Activites			FI Protocols
Introduction to Python and Sockets		Devlopment Stage (protocol implementation)	
Lab 1: Client Application (Chapter 2)	**Python**		
Socket client			**HTTP/2**
Standarized protocol and RFC (HTTP)			
Lab 2: Server Application (Chapter 2)	**Python**		
Socket server			
Home-made file-transfer protocol (HFTP)			
Oral exam of Lab 1 and 2			
Introduction to C++ and Omnet++		Analysis Stage (algorithm design-analysis)	
Lab 3: Transport (Chapter 3)	**Omnet++**		
Queuing and flow control (UDP/TCP)			**DTN, ICN,**
Error Recovery (Go-Back-N, Selective)			**Multipath TCP**
Lab 4: Network & Link (Chapter 4, 5)	**Omnet++**		
Dynamic routing (ring/random topology)			**SDN, IPv6,**
Medium access mechanisms (Aloha)			**IoT, LISP, WSN**
Oral exam of Lab 3 and 4			

based discrete-event networking simulation platform, was chosen as a test-bench. Experiment controllability, observability and repeatability are important benefits of this approach. As a matter of fact, a Python-based network simulator with the flexibility of Omnet++ would be just ideal, but no such tool is available yet. Other platforms such as NS3 [28] could also be considered.

To facilitate the Omnet++ learning curve, in Lab 3, students start with a skeleton code and are guided to develop a simple point-to-point network model over which the packet loss rate is measured in different buffer overflow cases. Then, they are requested to freely create flow and congestion control solutions (discussed in Kurose's chapter 3) and analyze and compare them. Discussions are held on the impact of applying these solutions in Future Internet versions of DTN, ICN, MTCP. Lab 4 further extends the model to multiple-node topologies where forwarding and link access decisions demand the creation of routing tables and negotiation strategies. In this activity, students are again left free to create the best possible strategy to optimize the delivery of data while analyzing resource consumption. Chapter 4 of Kurose's book can be consulted to inspire the design of an arbitrarily efficient routing protocol, while chapter 5 is used as inspiration for medium negotiation solutions. Software-Defined Networking (SDN) approaches and Locator/Identifier Separators such as LISP are openly discussed after this lab. Although highly creative and open, Labs 3 and 4 introduce students to technical and scientific writing allowing them to deliver in-depth analysis and evidence (e.g., curves, histograms) for their achievements (or failures). These reports have a high incidence in the grading of the second stage.

Pedagogy. The content approach conducts students to be experimentally involved in the learning process. This is an active learning approach that results in improved students *engagement* [12]. In particular, the utilization of prototypes and simulated algorithms allows students to (*a*) *create* and (*a*) *develop* through reflection on doing, an experimental learning investigated in [13]. On the one hand, being based on networking application interfaces coded by students, Lab1 and 2 are defined in the context of blended learning discussed in [15]. On the other hand, freedom of choice in Lab3 and 4 are central creativity-provoking activities that strictly follows the methodological approach known as creative learning [14]. The blended approach is known to improve *time efficient* learning, while creative exercises favors computational thinking, (*c*) *analysis* and (*d*) *comparison*, all objectives of the present action research experiment. In order to capitalize on one partner's resources and skills, all activities are executed in groups of three students [29]. Works in [29]–[32] implements some but no all of these pedagogical techniques in a networking course.

Grading. These instructional methodologies map into a grading criteria which measure the attainment of objectives (*a*)-(*d*). The two partial examinations are as follows: one for development stage (Lab1 and 2), and the one for the analysis stage (Lab3 and 4), averaging to a final CNDS laboratory grade. The systematic process to evaluate students is divided in three axes as follows.

Lab 1 and 2 (development stage):

- **CC.** (4/10 pts. max) **Code correctness (group):** Commited Python code is run through a detailed test cases suite including demanding edge cases (2 pts. for the client in Lab 1, 2 pts. for the client in Lab 2).
- **OD.** (4/10 pts. max) **Oral defense (individual):** students are requested to explain Lab1 and 2 and consulted on possible code modifications (2 pts.). Also, conceptual Python networking questions are posed (2 pts.).
- **WR.** (2/10 pts. max) **Written report (group):** Architectural documentation on the code justifying decisions taken for both the server and the client is evaluated.

Lab 3 and 4 (analysis stage):

- **WR.** (4/10 pts. max) **Written report (group):** Algorithm design and analysis reports are evaluated for each project (2 pts. per Lab).
- **OD.** (4/10 pts. max) **Oral defense (individual):** Students are requested to deepen on reported analysis and metrics (2 pts.). Conceptual questions from layer 4, 3 and 2 are posed (2 pts.).
- **CC.** (2/10 pts. max) **Code correctness (group):** Omnet++ code is visually inspected to evaluate the implemented algorithms and simulation models.

CC and OD are direct measures of (*a*) create and (*b*) develop, while WR and OD indicates the attainment of (*c*) analysis and (*d*) comparison.

IV. Action

The content and pedagogy design was applied and improved in two consecutive editions of the CNDS course in 2018 (60 students) and 2019 (76 students). This section presents transformation steps based on feedback collected during the action stage, following the principles of action research.

Stared Tasks. Inspired by students that exhibited notable engagement, we updated activities with optional starred tasks, which have a higher level of difficulty for extra points in the exam. In Lab 2, this maps to server capable of attending multiple clients simultaneously (polling or multi-threading). In Lab 4, the routing algorithm can be optionally applied on a complex topology of more than 50 nodes (requires hierarchical or highly polished distributed routing algorithms).

Public Expositions. Students enjoying more developed social soft-skills proved to support the attainment of objectives. To further exploit this, we instrumented optional public expositions opportunities. Similarly to [32], this allows students to defend the proposed algorithms while receiving questions and criticism from the audience and teachers. Meanwhile, slower or delayed students can easily catch-up, identify non-explored alternatives and learn on the process.

Game Contests. After discovering that students competed between groups, we established game contests to profit from game learning theory [31]. In Lab 2, server code is shared to other groups to find how a malicious client could provoke an unwanted behavior. In Lab 3, students competes to achieve the best delivery ratio under congestion. These fun activities gave the students a direct feeling of the problems and motivation of networking security and performance aspects.

By applying a circle of planning and action, these reflections where materialized into concrete pedagogical elements in an initial version in 2018, and later enhanced in 2019. To amplify the engagement and motivation we materialized these actions into three extra graded axes as follows.

- **ST.** (+1 pt.) **Starred tasks (group, optional):** groups delivering an operating multi-client server (thread or poll) for Lab2 or a routing solution capable of routing in a 50 node network in Lab 4 obtain an extra point.
- **PE.** (+1 pt.) **Public exposition (group, optional):** groups that uses the opportunity and succeed in publicly defending their server architecture (Lab2) or control algorithm (Lab4) are offered one extra point.
- **GC.** (+1 pt.) **Game contest (group, optional):** groups successful in finding security breaches in other's groups Lab2 or able to provide the best delivery ratio metrics in the scenario defined for Lab3 obtain an extra point.

Packet Traces. We finally learned that students were lacking of a concrete appraisal of the interaction of the explored layers. To compensate for this, we developed a closing (not-graded) activity introducing packet traces to map protocol-specific features to each of the concepts discussed throughout the course. The Wireshark tool, a free and open-source packet analyzer is considered for this purpose.

Content-wise, we took action on the reflection of a Lab5 assignment (link layer topic) in 2018, which was merged into Lab4 in 2019. Resulting class hour budget was measured to 16 hrs. for the first stage (40%), and 24 for the second one (60%). Regarding practical content, we learned and iterated on three aspects. (*i*) Providing virtual machines to students significantly reduced the burden of Omnet++ installation in Lab3 and 4[1]. (*ii*) The learning curve of Omnet++ was reduced by creating video tutorials that students could follow off-line anytime. (*iii*) An academic reporting tutorial was elaborated after inspecting difficulties in preparing the WR tasks in 2018. Items (*i*)-(*iii*) were fully incorporated in 2019.

V. Results

Results of the action research after the two cycles in 2018-2019 are positive. As indicated in Table II, 80% of the 60 students attained the required concepts in 2018, while 92% of the 76 achieved the objective in the refined course in 2019. The rest of the students need to re-take the course next year or pass the final open exam. Final grades where higher on average in 2019, although more dispersed ($\sigma_{2019} = 1.1 > \sigma_{2018} = 0.8$). We thus assess that the effectiveness of the approach improved 7% in-between the two cycles.

Axes. The performance over the individual axes are presented in the lower part of Table II. Performance on CC was improved on 9.24% in 2019, being Lab3 and 4 the main contributors to the overall effect. We explain this by the timely distribution of the Omnet++ virtual machine and tutorial video, which allowed the students to focus on the

Table II
Attainment Details

2018	stud.	60	passed	**80%**	avg.	**8.3**	min	**6.5**	max	**10**	std.d	**0.8**
2019	stud.	76	passed	**92%**	avg.	**8.9**	min	**6.5**	max	**10**	std.d	**1.1**

create/develop: CC, OD; analyze/compare: OD, WR; engagement/motivation: GC, ST, PE

2018	**CC**		**OD**		**WR**		**GC**		**ST**		**PE**	
Lab	**avg.**	**cnt.**	**avg.**	**cnt.**	**avg.**	**cnt.**	**avg.**	**cnt.**	**avg.**	**cnt.**	**avg.**	**cnt.**
1	1.93	50	1.70	50	0.80	50	-	-	-	-	-	-
2	1.22	50	1.50	50	0.76	50	1.00	8	1.00	10	1.00	1
Total	**3.15 / 4**		**3.20 / 4**		**1.56 / 2**		**8 / 50**		**10 / 50**		**1 / 50**	
3	1.01	35	0.30	35	0.98	35	1.00	8	-	-	-	-
4	0.93	35	0.66	35	0.94	35	-	-	1.00	5	-	-
5	0.86	35	0.62	35	0.82	35	-	-	-	-	1.00	2
Total	**2.80 / 4**		**1.58 / 2**		**2.74 / 4**		**8 / 35**		**5 / 35**		**2 / 35**	
%	**70.00%**		**79.00%**		**68.50%**		**22.86%**		**14.29%**		**5.71%**	
2019	**CC**		**OD**		**WR**		**GC**		**ST**		**PE**	
Lab	**avg.**	**cnt.**	**avg.**	**cnt.**	**avg.**	**cnt.**	**avg.**	**cnt.**	**avg.**	**cnt.**	**avg.**	**cnt.**
1	1.95	61	1.67	61	0.78	61	-	-	-	-	-	-
2	1.10	61	1.45	61	0.77	61	1.00	25	1.00	10	1.00	3
Total	**3.27 / 4**		**3.26 / 4**		**1.55 / 2**		**25 / 61**		**10 / 61**		**2 / 61**	
3	1.40	50	0.81	50	1.52	50	1.00	21	-	-	1.00	2
4	1.72	50	0.79	50	1.56	50	-	-	1.00	27	1.00	4
Total	**3.52 / 4**		**1.80 / 2**		**2.94 / 4**		**16 / 50**		**30 / 50**		**4 / 50**	
%	**77.13%**		**78.67%**		**77.17%**		**30.26%**		**24.34%**		**5.92%**	
Impr.	**9.24%**		**-0.42%**		**11.23%**		**24.46%**		**41.31%**		**3.47%**	

coding process instead of the tool installation burden. Improvement is also noticeable on the WR axis, with a stronger weight on the second stage portion. Based on feedback from students, we learned that the academic reporting tutorial was instrumental to improve the written quality and organization of the reports. We recorded that the increase in quality is more evident in Lab3 and 4 due to their notable analytical demand: students are free to propose solutions, but they have to properly justify and evaluate them. Oral defense scores remained practically unchanged on the two cycles, but also no action was specifically taken on that regard. Engagement axes, especially GC and ST, where boosted in the second cycle. This was indeed consequence of a formalized and timely announcement of these optional mechanisms in the 2019 cycle. Public expositions, on the other hand, were not so popular, mostly due to the fact the classroom time was rather limited to accommodate them, but also students proved to be remarkably shy to expose themselves in both editions. Interestingly, groups that faced the PE tend to be composed of politically active students (students union members), who enjoys good speeches, succeeding at convincing about the positive aspects of the chosen solutions[2].

Stages. Fig. 1 presents grades discriminated by stages: 1 (Lab1 and 2) and 2 (Lab3 and 4). Results show that on average, higher grades were obtained for the first evaluation in comparison with the second one. This is consistent with the Python programming interest of computer science students. Regarding student count, 83% of the students passed stage 1, but only 52% succeeded on the second stage in 2018. In 2019, these numbers increased to 95% and 70%. On the one hand, this is evidence that teaching network fundamentals is indeed more challenging than technology-specific programming tasks, despite the motivational and pedagogical condiments.

[1]The VM and Docker images can be obtained via the following URL: https://drive.google.com/open?id=1okuup7_q5XotPIvE653iaBX1UymIomrF. The username and password for the VM is "usuario" and "clave" respectively.

[2]It is the authors' opinion that such a capability is crucial in professional success. It is our vision that the extra point are well deserved by these groups.

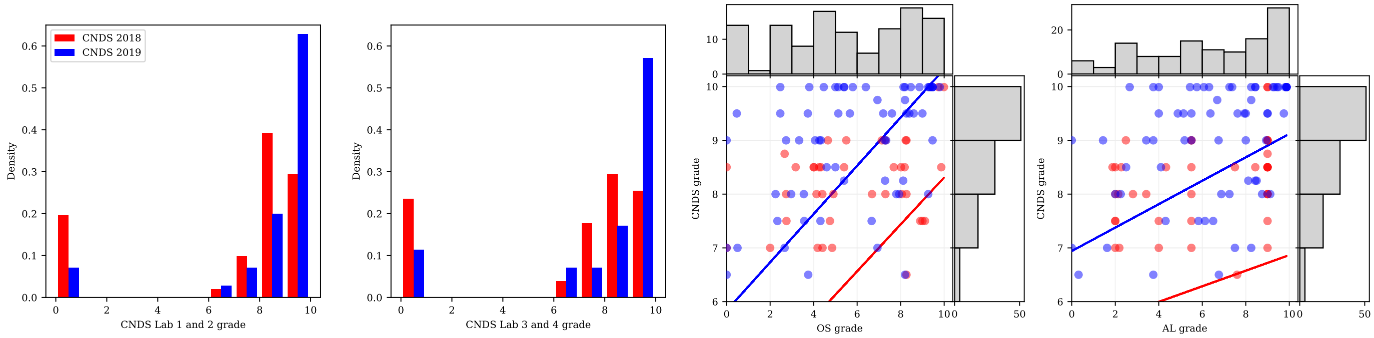

Figure 1. CNDS Lab1 to 4 grades (two on the left), and final grade correlation with student's background on Operating Systems (OS) and Algorithms (AL).

On the other hand, we explain the improvement on stage 1 by the enhanced game contest in 2019, with notable participation of 25 students (see Table II). The improvement in stage 2 is also related to the motivation axes, but also to the Omnet++ virtual machine and tutorials that eased the overall task.

Background. We evaluated the performance correlation of each student with their grading on the two previous subjects in FAMAF: operating systems (OS) and algorithms and data structures (AL). Fig. 1 presents scatter and histogram plot of CNDS, OS and AL grades for the 2018 and 2019 academic years. Grades lower than 6 in CNDS are all 0 (not passed) and are not shown to facilitate the appraisal of the distribution. On the one hand, we observe that students that passed the CNDS course present rather concentrated grades (in-between 9-10). Similar concentrations are also observed for AL, but OS histogram presents a more uniform grade distribution. On the other hand, the scatter plot indicates that students with higher grades in OS (7-10), also obtained higher in CNDS (8-10). This effect is less marked in AL. However, lower SO grades are rather distributed in higher and lower CNDS grades. To measure these effects, a linear regression was computed for 2018 (red line) and 2019 (blue line). A Pearsons' correlation of 0.36 and 0.52 was determined for SO, which combined with the linear regression evidences a moderate dependence on the topic. Pearsons' correlation was considerable lower for AL: 0.12 and 0.24, indicating a somehow unexpected lower correlation with the algorithmic challenges addressed in that subject. In both cases, however, correlation was registered higher in the second cycle. We explain this by the fact that the improvement after the action research cycles favored the revelation of each student's background.

Poll. An optional poll was provided to the students in order to obtain feedback. 36% and 27% of the 2018 and 2019 students answered the polls summarized in Figure 2. Students graded both the development (Lab1-2) and analysis (Lab3-4) stages positively, but slightly higher scores are observed for the former in 2018. This is somehow expected for computer science students having a first interaction with such a popular language like Python. Nevertheless, the tendency started to revert in 2019. Regarding difficulty, the analysis stage was pointed as more difficult, but difficulty self-assessment was improved in 2019, also explained by the actions taken on the second cycle. Furthermore, the poll allowed students to provide custom comments for the future editions of the course:

1) *"In addition to providing a video-presentation for Omnet++, also give one on how we should make the reports so that they meet the standards of an academic paper."*
2) *"I do not have a laptop and Omnet++ sometimes did not work properly in the classroom, I was only able to advance on my desktop computer at home."*
3) *"I had difficulty installing Omnet++. I learned a lot about networking and there is sure much more to learn. Thank you so much for everything!"*
4) *"Reports and analysis based in Omnet++ were very interesting, it was great to develop our own algorithms, it was very motivating. It would be better to have some instructions on report and graph making."*
5) *"Labs 3 and 4 helped me to understand the theory part better because I could see with my own eyes abstract concepts such as flooding and congestion.*
6) *"I would like to know more about cables and wireless."*
7) *"This was the first time I used Git, and had to learn without supporting material."*

Feedback 1) to 5) corresponds to 2018, and actually triggered the actions discussed in Section IV. Items 6) and 7) come from the poll in 2019, and motivates some of the finer upgrades as we enter the third 2020 cycle, which will include:

1) Repository video lecture: we will provide a basic Git tutorial including an integration with Omnet++ interface.
2) Mini-workshops: short talks to explain technological topics i.e., Ethernet cable crimping, router configuration, etc.
3) Public exposition: to overcome time consumption in classroom and shy students, we will propose a recorded video PE with off-line questions and feedback.
4) Python Omnet++: This would be a crucial breakthrough in CNDS as stage 1 and 2 could be blended together[3].

VI. CONCLUSIONS

In this paper we presented the application of an action research approach to redirect a hands-on computer networking course in the context of a computer sciences curriculum. We argued that a shift to a discussion of fundamental problems is necessary given the frenetic standardization of the Future Internet. Following an action research principle, we develop

[3] Authors are already advising a master student in developing a Python wrapper for Omnet++. Results are promising at the time of writing, and will be published in future Omnet++ summits.

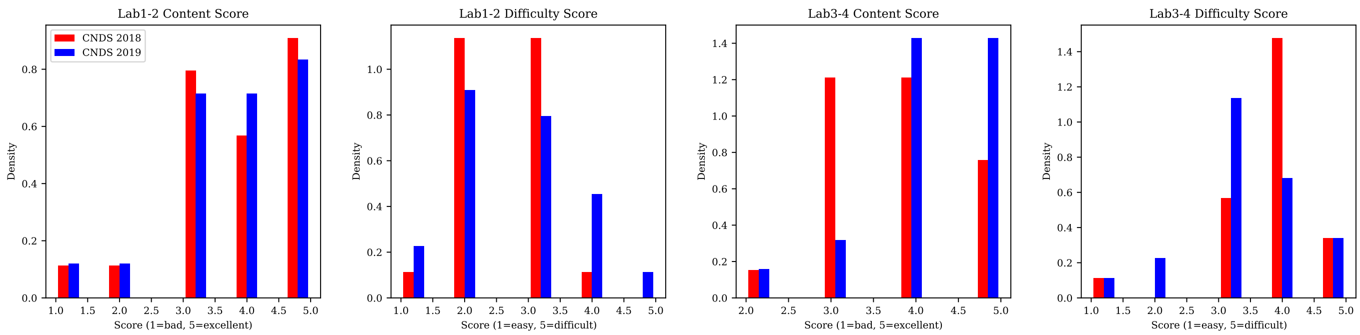

Figure 2. CNDS stage 1 and 2 student self-assessed lab content score and difficulty according to polls collected after 2018 and 2019 courses.

a novel active and creative approach on which 80% and 92% of the students attained the objectives in a two-year cycle. This is compelling evidence that the first cycle met the expectations and that the second iteration further improved the appraisal of fundamental networking concepts. We encourage other educators to consider the outcomes as students graduate prepared to develop and assess a broad range of Future Internet protocols regardless of technology-specific protocol specifications.

Acknowledgement

This research has received support from the ERC Advanced Grant 695614 (POWVER) and by the DFG Grant 389792660, as part of TRR 248 (https://perspicuous-computing.science).

References

[1] W. Ding, Z. Yan, and R. H. Deng, "A survey on future internet security architectures," *IEEE Access*, vol. 4, pp. 4374–4393, 2016.

[2] B. Momeni and M. Kharrazi, "Improving a computer networks course using the partov simulation engine," *IEEE Transactions on Education*, vol. 55, no. 3, pp. 436–443, Aug 2012.

[3] M. M. Raikar, P. Desai, and J. G. Naragund, "Active learning explored in open elective course: Internet of things (iot)," in *2016 IEEE eighth int. conf. on technology for education (T4E)*. IEEE, 2016, pp. 15–18.

[4] A. Gampe, A. Melkonyan, M. Pontual, and D. Akopian, "An assessment of remote laboratory experiments in radio communication," *IEEE Transactions on Education*, vol. 57, no. 1, pp. 12–19, Feb 2014.

[5] E. Taslidere, F. S. Cohen, and F. K. Reisman, "Wireless sensor networks; a hands-on modular experiments platform for enhanced pedagogical learning," *IEEE Trans. on Ed.*, vol. 54, no. 1, pp. 24–33, Feb 2011.

[6] Z. Dawy, A. Husseini, E. Yaacoub, and L. Al-Kanj, "A wireless communications laboratory on cellular network planning," *IEEE Transactions on Education*, vol. 53, no. 4, pp. 653–661, Nov 2010.

[7] H. Jasani, "Vendor neutral hands-on labs using open-source products for wireless networks courses," in *2010 IEEE Frontiers in Education Conference (FIE)*, Oct 2010, pp. S3F–1–S3F–6.

[8] C. Fang, H. Yao, Z. Wang, W. Wu, X. Jin, and F. R. Yu, "A survey of mobile information-centric networking: Research issues and challenges," *IEEE Communications Surveys Tutorials*, pp. 1–1, 2018.

[9] J. Suh, Z. Bojovic, M. Despotovic-Zrakic, Z. Bogdanovic, and A. Labus, "Designing a course and infrastructure for teaching software-defined networking," *Computer Applications in Engineering Education*, vol. 25, no. 4, pp. 554–567, 2017.

[10] R. Amirthavalli, R. Dhaya, and M. S. Chandrasoodan, "A survey of routing algorithms in delay tolerant networks," in *2016 2nd iCATccT conference*, July 2016, pp. 469–473.

[11] S. Hesham, J. Rettkowski, D. Goehringer, and M. A. A. E. Ghany, "Survey on real-time networks-on-chip," *IEEE Transactions on Parallel and Distributed Systems*, vol. 28, no. 5, pp. 1500–1517, May 2017.

[12] C. C. Bonwell and J. A. Eison, *Active Learning: Creating Excitement in the Classroom. 1991 ASHE-ERIC Education Reports.* ERIC, 1991.

[13] D. A. Kolb, *Experiential learning: Experience as the source of learning and development*. FT press, 2014.

[14] M. S. Peteranetz, A. E. Flanigan, D. F. Shell, and L.-K. Soh, "Computational creativity exercises: An avenue for promoting learning in computer science," *IEEE Transactions on Education*, vol. 60, no. 4, pp. 305–313, 2017.

[15] D. R. Garrison and N. D. Vaughan, *Blended learning in higher education: Framework, principles, and guidelines*. J. Wiley & Sons, 2008.

[16] E. T. Stringer, *Action research*. Sage publications, 2013.

[17] P. S. M. dos Santos and G. H. Travassos, "Action research use in software engineering: An initial survey," in *2009 3rd International Symposium on Empirical Software Engineering and Measurement*. IEEE, 2009, pp. 414–417.

[18] R. Guha and M. Bassiouni, "Modular design for distributed systems and network security courses in computer science curriculum," in *6th Int. Conf. on IT Based Higher Edu and Training*, July 2005, pp. 10–14.

[19] A. S. Tanenbaum, *Computer Networks*. Upper Saddle River, NJ, USA: Prentice Hall PTR, 1985.

[20] C. Goldstein, S. Leisten, K. Stark, and A. Tickle, "Using a network simulation tool to engage students in active learning enhances their understanding of complex data communications concepts," in *Proceedings of the 7th Australasian conference on Computing education-Volume 42*. Australian Computer Society, Inc., 2005, pp. 223–228.

[21] J. M. D. Hill, C. A. Carver, J. W. Humphries, and U. W. Pooch, "Using an isolated network laboratory to teach advanced networks and security," *SIGCSE Bull.*, vol. 33, no. 1, p. 36–40, Feb. 2001. [Online]. Available: https://doi.org/10.1145/366413.364533

[22] J. F. Kurose and K. Ross, *Computer Networking: A Top-Down Approach Featuring the Internet*, 2nd ed. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 2002.

[23] W. Stallings, *High Speed Networks and Internets: Performance and Quality of Service*, 2nd ed. Upper Saddle River, NJ, USA: Prentice Hall PTR, 2001.

[24] D. Bertsekas and R. Gallager, *Data Networks (2Nd Ed.)*. Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 1992.

[25] D. E. Comer, *Computer Networks and Internets*, 5th ed. Upper Saddle River, NJ, USA: Prentice Hall Press, 2008.

[26] A. Gomez-Sacristan, V. M. Sempere-Paya, and M. A. Rodriguez-Hernandez, "Virtual laboratory for qos study in next-generation networks with metro ethernet access," *IEEE Transactions on Education*, vol. 59, no. 3, pp. 187–193, Aug 2016.

[27] A. Varga and R. Hornig, "An overview of the omnet++ simulation environment," in *Proc. of the Int. Conf. on Simulation Tools and Techniques for Comms, Networks and Systems*, ser. Simutools '08. ICST, Brussels, Belgium: ICST, 2008, pp. 60:1–60:10.

[28] G. F. Riley and T. R. Henderson, *The ns-3 Network Simulator*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 15–34.

[29] R. E. Slavin, *Cooperative Learning. Research on Teaching Monograph Series.* ERIC, 1983.

[30] Ö. Delialioğlu, "Student engagement in blended learning environments with lecture-based and problem-based instructional approaches," *Journal of Educational Tech. & Society*, vol. 15, no. 3, pp. 310–322, 2012.

[31] M. Prensky, "Digital game-based learning," *Computers in Entertainment (CIE)*, vol. 1, no. 1, pp. 21–21, 2003.

[32] L. M. Regueras, E. Verdu, M. J. Verdu, and J. P. de Castro, "Design of a competitive and collaborative learning strategy in a communication networks course," *IEEE Transactions on Education*, vol. 54, no. 2, pp. 302–307, May 2011.