P4Checkers: Exploring Modern Network Concepts Through a Classic Game

Lucas Trombeta
Center for Engineering, Modeling and Applied Social Sciences
Federal University of ABC - UFABC
Santo Andre, Brazil
lucas.trombeta@ufabc.edu.br

Dener Edson O. G. da Silva
Center for Engineering, Modeling and Applied Social Sciences
Federal University of ABC - UFABC
Santo Andre, Brazil
dener.silva@ufabc.edu.br

Bruna Cunha Carvalho
Center for Engineering, Modeling and Applied Social Sciences
Federal University of ABC - UFABC
Santo Andre, Brazil
bruna.carvalho@ufabc.edu.br

Felipe Augusto Anon da Silva
Center for Engineering, Modeling and Applied Social Sciences
Federal University of ABC - UFABC
Santo Andre, Brazil
fanon@ufabc.edu.br

João Henrique Kleinschmidt
Center for Engineering, Modeling and Applied Social Sciences
Federal University of ABC - UFABC
Santo Andre, Brazil
joao.kleinschmidt@ufabc.edu.br

Carlos Alberto Kamienski
Center for Engineering, Modeling and Applied Social Sciences
Federal University of ABC - UFABC
Santo Andre, Brazil
cak@ufabc.edu.br

***Abstract*—This innovative practice full paper describes the challenge of engaging students in computer network education, where conveying complex concepts can be difficult. The proposed solution, P4Checkers, integrates the game of checkers with P4 programming to teach networking concepts interactively. Using the P4Docker simulation environment, the game transforms network packets into game actions, providing hands-on experience with network protocols. The effectiveness of P4Checkers was evaluated using the Post-Study System Usability Questionnaire (PSSUQ), showing positive results in improving student engagement and understanding of network concepts. This study highlights the potential of serious games like P4Checkers to enhance network education.**

***Index Terms*—P4, checkers, draughts, network education, educational games, P4 network simulator, computer network education, serious games**

I. Introduction

In the swiftly evolving domain of network programming, the advent of P4 (Programming Protocol-independent Packet Processors) [1] marks a significant shift, revolutionizing how network devices are programmed and managed. The flexibility and programmability offered by P4 cater to the demands of modern networks, allowing engineers and researchers to tailor network behavior with unprecedented precision. This paper introduces P4Checkers, an innovative educational tool that leverages the capabilities of P4 within a game-based learning environment to teach complex network concepts.

In accordance with forecasts outlined by the United States Bureau of Labor Statistics for the present decade [2], there is a discernible trend indicating an escalation in vacancies within Science, Technology, Engineering, and Mathematics (STEM) professions. Specifically within the realm of computer science, notable expansion is anticipated in roles pertinent to computer security and systems development, encompassing positions such as software developers and software quality assurance analysts and testers. On the contrary, roles linked to establishing and administrating computer networks are anticipated to demonstrate notably diminished growth rates, encompassing both percentage escalations and absolute numerical increments.

In contemporary contexts, the majority of routine activities such as email correspondence, office suite applications, and online meetings are predominantly facilitated through cloud computing paradigms, which rely heavily on computer networks. Paradoxically, this reliance has rendered the intricate workings of computer networks less salient to the end user, thereby diminishing the allure of understanding them as a subject of interest. Remarkably, the sophistication of computer networks has advanced to a degree where end users can effortlessly engage with network components without necessitating an exhaustive grasp of underlying principles. For instance, an individual can autonomously install and configure a wireless access point within their residence by simply adhering to instructional guides, devoid of external oversight or specialized knowledge.

On the one hand, undergraduate involvement in programming activities allows the automation of diaries and common tasks. In addition, programming skills can be used as a tool to solve other academic problems, endorsing something similar to self-guided problem-based learning that has been proved as an attractive way to stimulate students to stay connected to the acquired knowledge [3], even when they do not have programming classes on that period. On the other hand, undergraduate involvement in computer network activities typically revolves around configuring computer networks or addressing issues stemming from incorrect configurations, which generally represents a niche area with limited opportunities for engagement.

Those three factors: employment opportunities, effortlessly engaging with network components without deep concepts of computer networks, and limited opportunities for engagement might explain difficulties related to the involvement of students in computer network education.

Integrating serious games into education has enhanced learning through interactive and immersive experiences. P4Checkers utilizes this approach by embedding the game of checkers into a network simulation, where players control network packets instead of traditional game pieces. This setup makes learning more engaging and provides practical, hands-on experience with network protocols and routing. Our study evaluates the effectiveness of this educational strategy utilizing the P4Docker [4] environment to simulate a realistic network where game interactions mirror actual data communications.

By adapting the classic game of checkers, also known as draughts in some locations, into a network programming challenge, P4Checkers aims to bridge the gap between theoretical knowledge and practical application, making it an invaluable tool for students and educators. This game exposes learners to the core functions and potential manipulations within a network, providing a deeper understanding of underlying protocols and network operations.

To gauge the effectiveness of the P4Checkers game as a learning tool, the Post-Study System Usability Questionnaire (PSSUQ) [5] was employed. This standardized questionnaire assesses user satisfaction with system usability, information quality, and interface quality. The results derived from PSSUQ provided valuable insights into how well the students could grasp complex network concepts through interactive gameplay, thereby validating the educational efficacy of P4Checkers. This data-driven approach ensures that the educational tools meet learning objectives and enhance the user experience, a critical factor in educational technology.

This article is structured as follows: Section II outlines foundational concepts and methodologies, introducing tools like P4Docker and serious games; Section III reviews related works, placing this study within the broader academic landscape; Section IV details the P4Checkers scenario and gameplay; Section V analyzes results from the usability questionnaire; Finally, Section VI discuss future directions and concludes this paper.

II. Background

This section discusses the essential theoretical underpinnings and technological infrastructures that support the deployment and effectiveness of the P4Checkers educational tool. It will explore the role of P4Docker as a simulation environment used to deploy P4Checkers, providing enhanced realism and isolation for network experimentation. Additionally, the discussion will delve into the incorporation of serious games as a transformative educational strategy, particularly within the realm of computing and network education. Through a detailed exposition, the methodologies and tools employed in P4Checkers will be substantiated, emphasizing their relevance and significance in modern network education.

A. P4Docker

The P4Docker [4] is a novel tool that offers significant benefits over traditional network simulation environments, providing enhanced isolation and a more realistic test bed for P4 switch experiments. Unlike alternative tools, such as Mininet, which typically runs hosts and switches within the same operating system environment - potentially leading to interactions that do not accurately reflect separate network components - P4Docker leverages Docker containers to isolate each component. This isolation ensures that each simulated host and switch operates independently, mimicking real-world network infrastructure more closely. This setup not only enhances the reliability of network simulations but also provides a more controlled environment for testing and debugging network protocols and configurations.

Furthermore, P4Docker introduces a user-friendly graphical interface that simplifies the design, management, and execution of network experiments. This GUI allows users to construct and manipulate network topologies visually, streamlining the learning curve and enhancing productivity for novice and experienced network developers. The tool also automates the generation of scripts necessary for setting up and tearing down network configurations, facilitating rapid prototyping and iterative testing. Overall, the introduction of P4Docker represents a strategic advancement in network testing tools, combining robust functionality with ease of use to support innovative network design and research. Combined with the ability to operate P4 switches seamlessly, these characteristics led to the development of P4Checkers based on the P4Docker tool.

B. Serious Games

Serious games have emerged as a pivotal educational tool within computing education, particularly for teaching complex subjects like computer networks. These games harness interactive and immersive technologies to create engaging learning environments that foster active participation and a deeper understanding of the subject. Serious games significantly enhance student collaboration and reinforce their grasp of critical network protocols such as TCP, UDP, and IP addressing [6].

The advantages of serious games in educational settings extend beyond simply making learning more interactive. They provide a context where theoretical knowledge can be applied in practical, scenario-based tasks crucial in computer networking. Pereira's study [6] demonstrated that students engaged in solving logical puzzles and uncovering clues related to network routing. These activities improve problem-solving skills and deepen their understanding of network protocols in real systems.

Beyond the realm of computing, the use of serious games in science education has also been extensively documented. The systematic review presented in Ullah's [7] and Kara's [8] work revealed that games employing virtual reality, artificial intelligence, and augmented reality are increasingly used to teach various science subjects, including physics. This approach makes learning more engaging and allows for the exploration of complex scientific concepts in a more intuitive and accessible manner. As serious games expand across different educational fields, their role in enhancing learning outcomes and student engagement continues to be validated, making them an invaluable component of modern educational methodologies.

C. PSSUQ

The Post-Study System Usability Questionnaire (PSSUQ) is a widely recognized tool used to evaluate the usability and satisfaction of users with system interfaces. Developed by IBM [5], it consists of 16 questions that measure various aspects of the user experience, including overall satisfaction, system usability, information quality, and interface quality. In the context of P4Checkers, the PSSUQ was employed to gather quantitative data on how students interacted with and perceived the educational tool. This questionnaire is handy in educational technology research as it provides insights into the effectiveness of the tool in facilitating learning and engaging users, thereby allowing developers to make informed decisions about necessary improvements and modifications.

In the P4Checkers, the results obtained from the PSSUQ were instrumental in assessing the educational impact of the game-based learning approach. The questionnaire allows us to pinpoint areas where the game met or failed to meet the educational needs of students, highlighting aspects such as the interface intuitiveness, the clarity of the information presented within the game, and the overall satisfaction with the interactive learning environment. The feedback collected through the PSSUQ allows the evaluation of the success of P4Checkers in making complex network concepts accessible and engaging. This feedback loop is vital for iterative development, ensuring that the educational tool continuously evolves to meet the changing educational demands and preferences of students in network programming.

III. Related Work

Over the past two decades, educational researchers have dedicated significant effort to studying the methods employed in teaching computer networks. This focused inquiry stems from recognizing the pivotal role computer networks play in modern society and the workforce. Understanding that effective education in this field is crucial, researchers have sought to enhance the learning experience for students by exploring various teaching approaches and methodologies. During the early 2000s, Berglung endeavored to explore an alternative approach employing Remote Method Invocation (RMI) to facilitate widespread student interaction within a computer network class, despite their disparate geographical locations [9]. Their overarching goal has been to improve students' perceptions of computer network courses, ensuring that learners are equipped with both the knowledge and enthusiasm necessary to navigate this complex domain.

A key aspect of this research has been to assess the efficacy of different pedagogical strategies in conveying the intricacies of computer networks to students. This has involved analyzing the impact of traditional lecture-based approaches as well as more interactive methods such as hands-on lab sessions, simulations, and collaborative projects [10]. Researchers have identified strengths and weaknesses by systematically evaluating these teaching modalities, ultimately guiding the development of more engaging and effective instructional practices [11]. Moreover, this ongoing scrutiny has provided insights into how to tailor teaching methodologies to suit the diverse learning styles and preferences of students, thereby fostering a more inclusive and dynamic learning environment [12]. In the same way, Packet Warriors, a game crafted by researchers at Chulalongkorn University, imparts these concepts through a shotgun-style gaming interface, integrating computer network quizzes that demand from the students the ability to identify the correct answers to advance to the subsequent level [13].

Furthermore, the efforts of educational researchers have not been confined solely to instructional techniques but have extended to the broader educational context surrounding computer network courses. This includes investigations into curriculum design, assessment methods, and integrating emerging technologies into teaching practices. Prevalent tools employed by numerous educators are network simulators and emulators, which circumvent the expenses and time constraints associated with establishing a physical infrastructure [14].

Cisco Packet Tracer is widely acknowledged as the preeminent network simulator globally, renowned for its robust feature set and user-friendly interface. Developed by Cisco Systems, Packet Tracer holds a prominent position in computer network education, playing a central role in the training and development of individuals within the field of computer networks [15] . Its comprehensive platform allows users to design, configure, and troubleshoot complex network topologies in a simulated environment, offering a practical and immersive learning experience [16] . Packet Tracer's emulation capabilities enable users to experiment with various networking technologies, protocols, and configurations, providing a valuable educational tool for enhancing both theoretical knowledge and practical skills in network management [17]. Its widespread adoption in educational institutions worldwide underscores its significance as a foundational component in the education and training of individuals pursuing careers in computer networks.

Through meticulous examination and integration of diverse educational facets, scholars endeavor to foster a comprehensive learning environment. This environment not only disseminates technical knowledge but also nurtures critical thinking acumen, problem-solving proficiencies, and a profound comprehension of the societal ramifications inherent in computer network dynamics. Jordan's groundbreaking work exemplifies this holistic approach by employing packet analyzers and network traffic, particularly in relation to the Emotet malware [18]. His endeavor challenges students to discern patterns, potentially enabling them to devise filters and spearhead investigations to ascertain the origins and extent of infection across network hosts. In concert with such innovative endeavors, educational researchers collectively contribute to the transformative evolution of computer network education, thereby equipping students to excel within an increasingly interconnected global landscape.

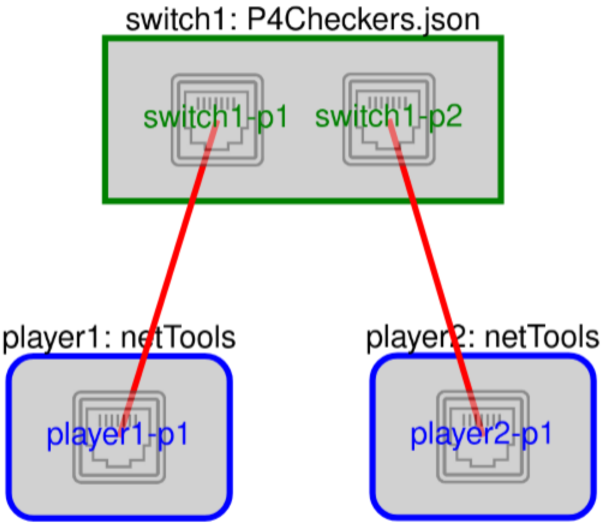

Fig. 1. Physical topology of P4Checkers containers

IV. P4Checkers

P4Checkers is based on the idea of a game played over a network tool or run over a network device. However, it does not require any network tool background or advanced network concepts to play. Learning network tools and concepts is like a side quest in this game. Analogously, the core game mechanism, checkers, needs no introduction for most students.

There are two popular game notations, Portable Draughts Notation and Forsyth-Edwards Notation, that maps all dark squares with numbers 1 to 32(or 50 in 10x10 international checkerboard) to design the gameplays [19]. P4Checkers could be designed over the same notation, but a row and column design looked up more comprehensible once it enounces a two dimension space directly, otherwise, players must unravel in their minds simple numbers to a two dimension space. The P4Checkers' checkerboard is mapped 8x8, both row and column numbers are comprehended between 1 and 8, 1 meaning first row or column and 8 meaning the last row or column, which means, in order to access a position the player should invoice row and column respectively, e.g. 4 2 refers to the row 4 and column 2.

Although P4Checkers is inspired by the American checkers game, there are some restrictions at this implementation: Omits to capture a piece is not considered a illegal move, a man can just capture one man at a time(forward or backward), there is no crowned pieces

A checkerboard is historically composed by light and dark squares, in our game we recreated that environment changing light squares, that are not used, by ”-” and dark squares by ”0”. Player 1 man pieces will be represented by 1 instead of 0 and in the same way player 2 man pieces will be represented by 2 instead of 0.

A. Scenario

To enhance accessibility to the game, a virtual machine running on Ubuntu 22.04 was made available. Within this virtual machine, a Docker environment is configured, housing P4Docker containers. Two of these containers function as host machines, while one serves as a P4 network device, facilitating interconnectivity between them.

The game of checkers is widely recognized as involving two players and a board with game pieces. In the context of P4Checkers, these players are represented by two hosts, designated as player1 and player2, and their interaction is facilitated through a network switch.

As illustrated in Fig. 1, both hosts are interconnected via a network switch that executes P4 code. Given that P4Docker is utilized, the switch in question is an emulated BMv2[1] switch. Each host has an image containing network tools available for each player.

Students have the capability to construct the topology depicted in Fig. 1 within the virtual machine (VM) by executing a script located on the VM's desktop.

In addition to establishing the topology containers and connections, the script also initiates the network switch with the game-specific code, which is written in P4 and compiled into a JSON file. Furthermore, the script launches two instances of the Wireshark[2] software, as illustrated in Fig 2, enabling the monitoring of network packets on each host's network interface.

Students are recommended to type the word ”udp” into the filter bar of Wireshark, which is positioned at the top of the window, as in Fig 2. The filter bar aims to display only packets that correspond to a User Datagram Protocol(UDP) segment.

At this stage, an educator may choose to impart or refresh students understanding of a UDP segment, providing concise elucidation regarding its header structure and delineating the placement of UDP within the Open Systems Interconnection (OSI) and Transmission Control Protocol/Internet Protocol (TCP/IP) models.

B. Game start

Once the student has created the structure with the three containers, it is time to start the game. The game, that is recorded inside the switch, uses UDP segments between 2 hosts in order to design and modify the checkers board. So, the student must start UDP sockets in each host. In our design, player 2 must listen UDP segments in port 5555(a homage to the P4RROT project [20]) and player 1 must send segments to player 2 in port 5555. Our container images for hosts already have the software Netcat installed.

Player 1 must start with

```
$ netcat -u 10.0.2.2 5555
```

and player 2 must start with

```
$ netcat -ul -p 5555
```

[1]https://github.com/p4lang/behavioral-model
[2]https://www.wireshark.org/

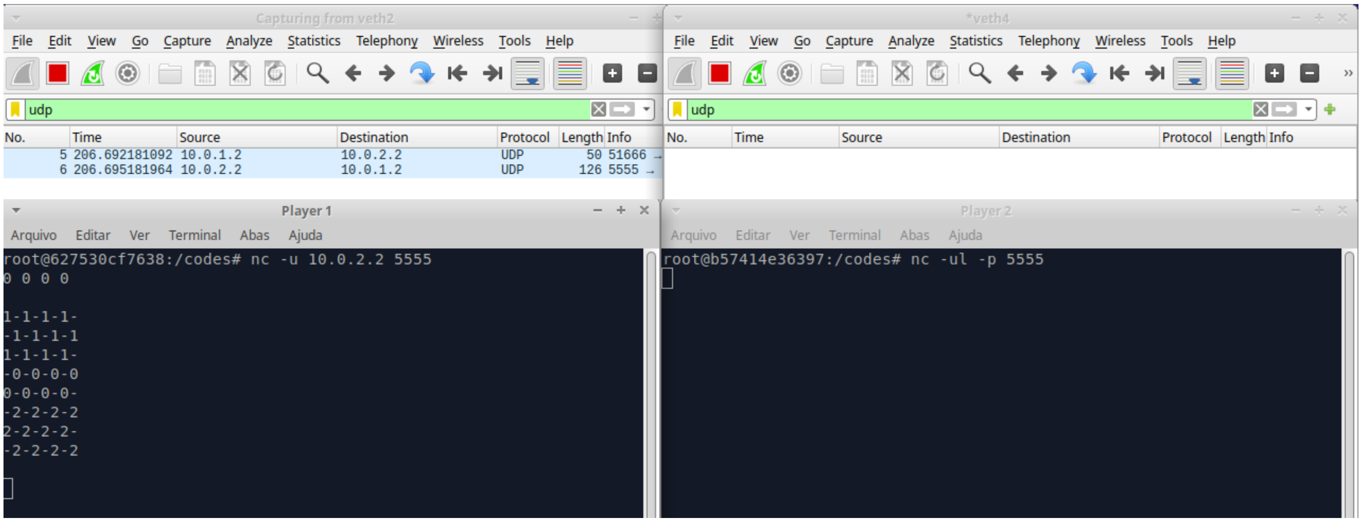

Fig. 2. Terminal and Wireshark windows of both players after a game start

At this point, students start to learn the use of a new tool, Netcat[3], and how they can create a communication between two hosts using just a command-line interface(CLI)

The instruction states that the game must be initiated by player 1 through the command "0 0 0 0" written in the Netcat command-line.

If this command is well-succeeded a checkers board will appear in player 1 command line as seen in Fig 2.

As mentioned before pieces of player 1 are represented by "1", pieces of player 2 are represented by "2", free spaces are represented by "0", illegal spaces (light squares) are represented by "-"

At this point, students should be able to observe two things through Wireshark instances: the segment that player 1 sent to player 2 was not delivered to player 2; and a segment that was received by player 1 with player 2 IP address was not sent by player 2. this behavior is demonstrated in Fig. 2. Wireshark provides rich and detailed information about the packages transmitted between the network interface, such as source, target, protocol, etc. The columns at the Wireshark screen demonstrated in Fig. 2 shows the source and target of each captured package, indicating that a UDP package was sent from host 10.0.1.2 (player 1) to 10.0.2.2 (player 2).

This observation leads to a small but important conclusion: The P4 device is changing the path of the network packet that is sent by this device.

C. *Player 1 first move*

In P4Checkers game player 1 is always the first to play the game. It is suggested that command 3 3 4 4 should be written on CLI, it indicates to the game that player 1 wants to move the piece located in row 3, column 3 to the row 4, column 4.

If this command is well-succeeded, a checkers board with updated positions will appear in the player 2 command line, as presented in Fig. 3. This indicates that it is now the player 2 turn.

At this time, students should be inquired to use Wireshark to answer if there are differences between the packet sent by player 1 and the packet received by player 2. They should observe that although the header informations seemed almost the same, the UDP payload and the header field that is related to the segment length had changed.

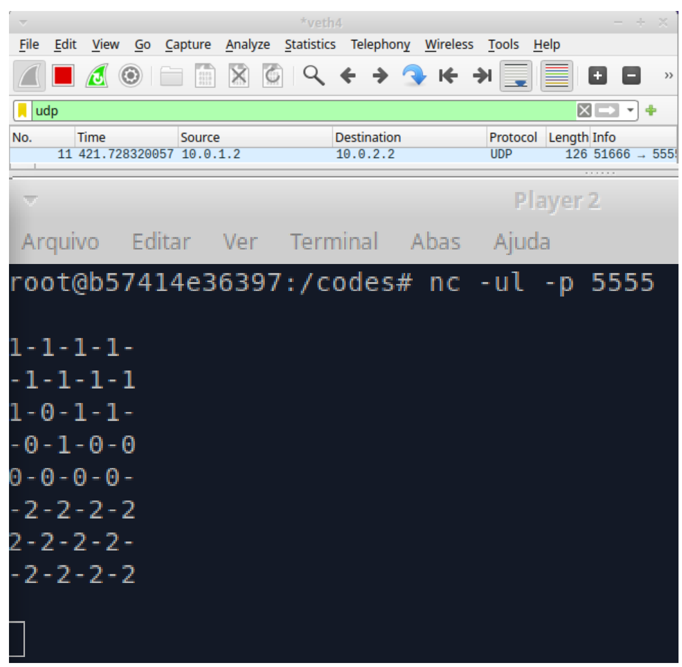

Fig. 3. Player 2 Netcat CLI and Wireshark screen after player 1 move

D. *Player 2 first move*

When a board is shown at player 2's Netcat CLI it is time to play. Player 2 must move one of its pieces. It is suggested

[3]https://netcat.sourceforge.net/

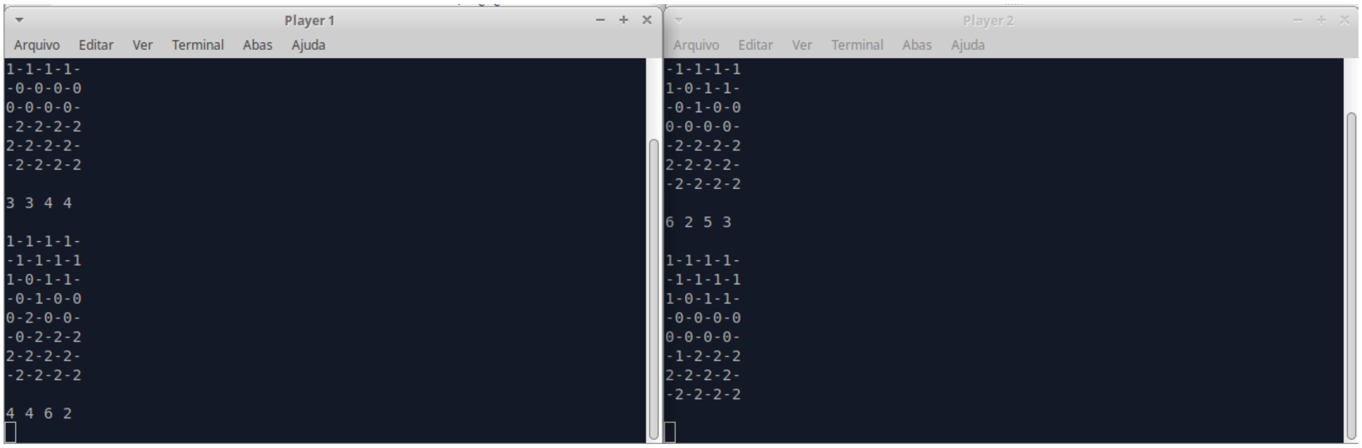

Fig. 4. Player 1's piece at row 4, column 4 executes a jump over player 2's piece situated at row 5, column 3.

that command 6 2 5 3 should be written on CLI, it indicates to the game that player 1 wants to move the piece located in row 6, column 2 to the row 5, column 3.

If this command executes successfully, a checkerboard displaying updated positions will be shown in player 1's Netcat command line. This indicates that it is now player 1's turn.

E. Jump

As elucidated at the outset of this section, capturing pieces in this game is not obligatory. However, captures may occur provided they adhere to the restriction of jumping over one piece at a time as Fig.4 shows.

To execute a jump and capture a piece, the move must be directed to the square immediately following the opponent's piece. In the example brought by Fig. 4, player 1's piece located at row 4 and column 4 jumps over player 2's piece located at row 5 and column 3, ending on row 6 and column 2. The position at row 5, column 3, previously occupied by a piece belonging to player 2, has been converted to a free space, denoted by "0." Consequently, the total number of player 2's pieces has decreased by one.

F. Invalid move

There are input validation mechanisms in place to prevent invalid moves. If a player attempts to position a piece outside the boundaries of the board or make an illegal move (e.g., moving a piece to an unauthorized space or jumping two steps ahead instead of one), the p4 device will detect the action and return a packet indicating "invalid move" as illustrated in Fig 5.

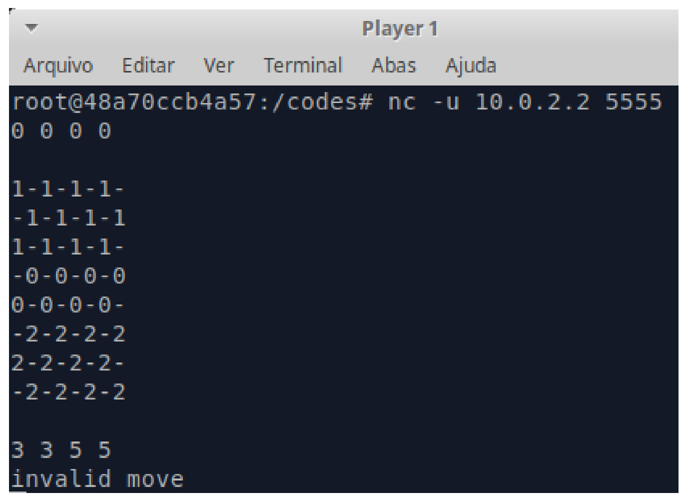

Fig. 5. Player 1 tries to move a piece illegally

G. Turn

In the context of gameplay, a P4 switch register assumes the role of facilitating turn rotation. Upon the execution of a valid move by a player, the turn register undergoes alteration, thereby enabling the progression of the game for the subsequent player.

In instances where a player transmits a network packet outside their designated turn, the P4 device promptly detects such action and responds by dispatching a packet conveying the message, "not your turn", as denoted by Fig 6.

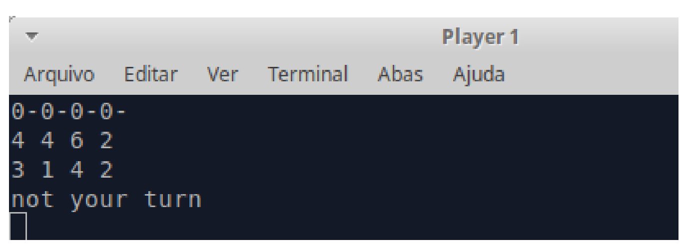

Fig. 6. Player 1 tries to move a piece out of its turn

H. Game ending

The conclusion of a game is exclusively predicated upon one player capturing all of their opponent's pieces. This unique termination condition stems from the inherent limitations of the P4 language, which lacks recursion or loops. Without the capacity for iterative processes or recursive functions, P4Checkers is unable to implement mechanisms that might facilitate alternative victory conditions or draw resolutions. Consequently, the only feasible conclusion to a game is reached when one player successfully eliminates all of the opposing player's pieces from the board. This design ensures that gameplay adheres strictly to the rules and constraints of the P4 language, embodying a simplified yet deterministic framework for game conclusion.

V. Results

This educational methodology underwent evaluation within two distinct classes of IT associate degree programs. One class comprised students enrolled in the cybersecurity course, while the other consisted of students enrolled in the IT management course. The aggregate enrollment across both classes amounted to 36 students. Both classes were consolidated into a single entity for the purpose of evaluating the results, given their equivalent levels of prior knowledge regarding P4 and computer networks.

After classes, data were collected through an online form with PSSUQ v2 questions, and the results are shown in Figure 7. The rating scale ranges from 1 to 7, with 1 indicating strongly agreement and 7 indicating strongly disagreement with the statement. Error bars represent 95% confidence interval(CI). CI is the interval within which the true population parameter (e.g., mean) is expected to fall 95% of the time if the study were to be repeated multiple times.

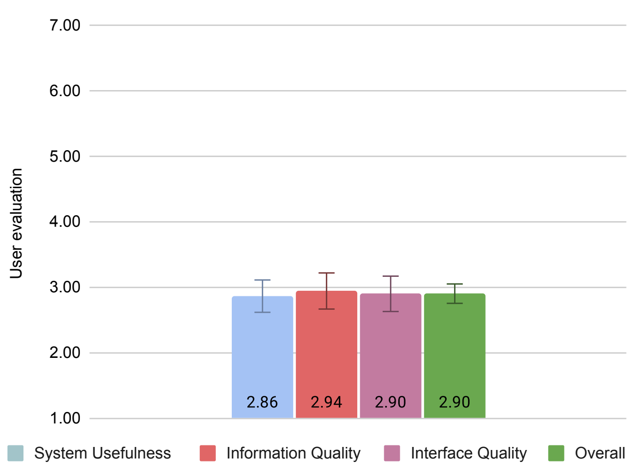

Fig. 7. User satisfaction measured by the PSSUQ

The ratings for parameters related to system usefulness, information quality, and interface quality were close to 3, resulting in an overall rating also near 3. These results indicate that students were able to understand the purpose of the P4Checkers game. Interestingly, these outcomes were better than expected. The game was developed and tested by individuals with significant expertise in P4 language and P4Docker, suggesting that while the instructions were clear to those within this specialized field, they might require additional clarification for individuals not immersed in this area.

Additionally, it was noted by the instructors that inquiries about the technologies employed in the game were made by several students in subsequent classes. Moreover, the knowledge acquired during the lesson was applied by certain students to other tasks involving computer networks. Based on the instructors' own experiences, such behavior was considered atypical for these classes.

VI. Conclusion

The P4 language allows for further development on the networking topic and sometimes beyond, as was demonstrated by this paper, which created an innovative way to help students be in touch with computer network principles without the constraints of a particular network hardware or device and its accompanying operating system. Alongside this, it became evident that the possibilities are almost endless, especially because of the flexibility P4 offers to the programmers and users. P4Checkers has the appeal to bring a world famous game inside a computer network scenario. The fact of being able to start and play a game is itself challenging to students and grabs their attention, as demonstrated at data collect from PSSUQ form. In addiction to that, P4Checkers, different of cited approaches that used educational hands-on artifacts [9] , does not require advanced computer skills as programming, since the computer networks subject might be offered in the same academic semester that students are learning basic programming skills.

Although the game is also designed to challenge advanced computer science students. There was no opportunity to apply it on a post-graduate class. Advanced students would be presented to the code written in p4 language and how the game mechanism is implemented - moves, verifications, turn sequences - and encouraged to develop a significant part of the game: the king's movement, that isn't written by purpose, but all the code supplies to design it are given. A future work is evaluate the game with advanced students implementing the king's movement if they will be able to build the code, and how the game will impact their knowledge about p4 language. Another appealing job prospect will involve assessing groups' interest in the field of computer networks both before and after tasks.

In conclusion, the outcomes associated with the game are exceedingly positive. It effectively captures students' attention during lessons, enhancing engagement and making the learning process more enjoyable. Moreover, it fosters the development of relevant skills and stimulates interest in computer network concepts and technologies related to the game. This approach cultivates a dynamic and interactive educational environment, encouraging students to explore, learn, and develop with enthusiasm and proactivity.

Acknowledgements

This work was supported by the São Paulo Research Foundation (FAPESP) under the grant number 2020/05152-7, the PROFISSA project.

References

[1] P. Bosshart, D. Daly, G. Gibb, M. Izzard, N. McKeown, J. Rexford, C. Schlesinger, D. Talayco, A. Vahdat, G. Varghese, and others, "P4: Programming protocol-independent packet processors," *ACM SIGCOMM Computer Communication Review*, vol. 44, no. 3, pp. 87–95, 2014.

[2] A. Zilberman and L. Ice, "Why computer occupations are behind strong STEM employment growth in the decade," *Beyond the Numbers: Employment & Unemployment, vol. 10, no. 1 (U.S. Bureau of Labor Statistics)*, vol. 10, no. 1, 2021.

[3] L. E. Osgood, "Publishing Student-Generated Problems in an OER," *International Journal of Engineering Pedagogy (iJEP)*, vol. 12, pp. 115–132, 7 2022.

[4] D. Silva, A. Heideker, L. Trombeta, B. Carvalho, J. Kleinschmidt, and C. Kamienski, "P4docker: Enabling efficient p4 switch testbeds with docker integration," in *Anais Estendidos do XLII Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos*, (Porto Alegre, RS, Brasil), pp. 1–8, SBC, 2024.

[5] J. R. Lewis, "IBM computer usability satisfaction questionnaires: Psychometric evaluation and instructions for use," *International Journal of Human–Computer Interaction*, vol. 7, no. 1, pp. 57–78, 1995.

[6] R. Pereira and W. Viana, "A Serious Game for Teaching Computer Networks: A Comparison between Remote and In-Person Scenarios," in *2023 IEEE Frontiers in Education Conference (FIE)*, (Los Alamitos, CA, USA), pp. 1–9, IEEE Computer Society, 10 2023.

[7] M. Ullah, S. U. Amin, M. Munsif, M. M. Yamin, U. Safaev, H. Khan, S. Khan, and H. Ullah, "Serious games in science education: a systematic literature," *Virtual Reality & Intelligent Hardware*, vol. 4, no. 3, pp. 189–209, 2022.

[8] N. Kara, "A Systematic Review of the Use of Serious Games in Science Education," *Contemporary Educational Technology*, vol. 13, p. ep295, 5 2021.

[9] A. Berglund and A. Pears, "Students' understanding of computer networks in an internationally distributed course," in *Proceedings of the The 3rd IEEE International Conference on Advanced Learning Technologies (ICALT'03)*, pp. 380 – 381, 5 2003.

[10] Z.-g. Xiong, X.-m. Zhang, Y. Xie, and J.-x. Chen, "Research on reform of teaching contents and methods of "computer network" to the local university," in *2010 International Conference on E-Health Networking Digital Ecosystems and Technologies (EDT)*, vol. 2, pp. 218–220, 2010.

[11] S. Wei, Z. Qunyi, and L. Xuefen, "Research of educational technology of computer networks," in *2010 International Conference on E-Health Networking Digital Ecosystems and Technologies (EDT)*, vol. 2, pp. 192–195, 2010.

[12] G. Fuxiang, Y. Lan, Y. Yu, C. Xiuli, and Z. Dongmei, "Research on the Teaching Method for Computer Network Course," in *2009 International Forum on Computer Science-Technology and Applications*, vol. 2, pp. 414–416, 2009.

[13] T. Uiphanit, P. Bhattarakosol, K. Suanpong, and S. Iamsupasit, "Packet Warriors: An Academic Mobile Action Game for Promoting OSI Model Concepts to Learners," *International Journal of Interactive Mobile Technologies (iJIM)*, vol. 13, p. 41, 6 2019.

[14] J. Gomez, E. F. Kfoury, J. Crichigno, and G. Srivastava, "A survey on network simulators, emulators, and testbeds used for research and education," *Computer Networks*, vol. 237, p. 110054, 12 2023.

[15] J. Allison, "Simulation-Based Learning via Cisco Packet Tracer to Enhance the Teaching of Computer Networks," in *Proceedings of the 27th ACM Conference on on Innovation and Technology in Computer Science Education Vol. 1*, (New York, NY, USA), pp. 68–74, ACM, 7 2022.

[16] N. b. Abdul Rashid, M. Z. Bin Othman, R. Bin Johan, and S. F. Bin Hj. Sidek, "Cisco Packet Tracer Simulation as Effective Pedagogy in Computer Networking Course," *International Journal of Interactive Mobile Technologies (iJIM)*, vol. 13, p. 4, 9 2019.

[17] M. Vijayalakshmi, P. Desai, and M. M. Raikar, "Packet Tracer Simulation Tool as Pedagogy to Enhance Learning of Computer Network Concepts," in *2016 IEEE 4th International Conference on MOOCs, Innovation and Technology in Education (MITE)*, pp. 71–76, IEEE, 12 2016.

[18] J. Allison, "Network Packet Analysis as a Unit of Assessment: Identifying Emotet," in *Proceedings of the 22nd Koli Calling International Conference on Computing Education Research*, (New York, NY, USA), pp. 1–2, ACM, 11 2022.

[19] C. Can, "A web-based visualization for analysis of international draughts," Master's thesis, Eindhoven University of Technology, Eindhoven, 3 2021.

[20] C. Györgyi, S. Laki, and S. Schmid, "P4rrot: Generating p4 code for the application layer," *ACM SIGCOMM Computer Communication Review*, vol. 53, no. 1, pp. 30–37, 2023.