Towards a Collection of Packet Trace Interactive Exercises for Computer Networking Education

Maxime Piraux
UCLouvain
maxime.piraux@uclouvain.be

Ludovic Taffin
UCLouvain
ludovic.taffin@uclouvain.be

Olivier Bonaventure
UCLouvain
olivier.bonaventure@uclouvain.be

ABSTRACT

Students do not only learn by reading textbooks and listening to their professors. They also learn by answering questions challenging their knowledge and enabling them learn from their mistakes. In this paper, we present a set of automated exercises that empower students to learn in an online setting. These exercises take the form of cloze-like tests requiring students to fill in missing packet field values of several key Internet protocols. Those educational resources are open and free to use. We invite the computer networking teaching community to collaborate to add new exercises covering further key Internet protocols.

1. INTRODUCTION

The understanding of key network protocols is part of the objectives of many networking courses. To achieve this, most of them rely on textbooks containing text descriptions of those protocols. Such textbooks typically describe the protocol packet format and finite state machine, as well as some examples. They are thus often restrained to a theoretical introduction of the protocols. Using a packet dissector such as Wireshark or `tcpdump` can make the protocols become real to students, as they learn by observing the packets exchanged inside a network. Some textbooks already include packet traces and invite students to use those tools [4, 5]. In this white paper, we present a set of interactive cloze-like exercises based on real packet traces.

To implement those interactive exercises, we leverage the INGInious automated grading platform [3]. INGInious provides a simple and secure way to execute and test untrusted code. It has been developed for the automatic grading of programming assignments. INGInious is completely language-agnostic and is able to run many environments. INGInious offers a modular architecture, with several available frontends. INGInious can be integrated to existing Learning Management Systems (LMS) such as edX and Moodle thanks to its LTI module.

Assignments in INGInious are defined as tasks within courses. A task can be comprised of several questions. Questions can be multiple choice questions, open questions requiring short text answers and programming questions requiring the student to write code. After submitting their answers, students receive an automated feedback based on the evaluation scripts part of the task. A powerful plugin system allows extending the type of tasks that can be presented to students.

Figure 1: INGInious exercice with a TCP packet trace. The student is asked to infer the TCP Acknowledgment Number of an ACK and is presented with an explanation of this field.

2. LEARNING INTERNET PROTOCOLS FROM PACKET TRACES

We extended INGInious with a plugin leveraging the Wireshark dissector to present dissected packet traces inside questions [7]. This new type of questions for INGInious can be leveraged in several ways. First, the packet trace can be displayed in its entirety for the student to answer some multiple choice questions. For instance, a student could be asked to choose the correct `Transaction ID` in the trace of a DNS query. INGInious also allows short answers that can be pattern-matched for correctness. For example, a student could be asked to list the domain names of name servers in the authority section of a DNS query. Second, the packet trace order can be randomized for the student to order them back based on their knowledge of the protocol and the packets field. For instance, a student could be asked to reorder packets of a TCP handshake. Third, some fields of the packet trace can be masked for the student to infer and input the hidden value. This is very similar to a cloze test in a natural language. For example, a student could be asked to compute the `TCP Acknowledgment Number` of an `ACK` sent in response to an incoming segment. This last example is illustrated in Figure 1.

In Figure 1, a packet-trace based INGInious exercise is presented. The exercise is composed as follows. First a title and header indicate the context of the question. Below, a table lists the packets present in the packet trace. The student can click on any packet in the table and be presented with the detailed packet payload and dissection. In the Figure, the fourth packet is selected and its content is shown below. On the left-hand side, the packet payload is displayed in both hexadecimal and ASCII format. On the right-hand side, a partial dissection of the packet by Wireshark is displayed. The student can click on a given field to highlight its position in the packet payload, similarly to the Wireshark GUI. Packets with missing fields are indicated in the trace table with an exclamation sign, and fields that must be filled are followed by a text box.

After filling the missing values in the packet trace and submitting their answers, the INGInious server computes the feedback and presents it to the student in a matter of a few seconds. Figure 2 illustrates the feedback received by a student that erroneously computed the `TCP Acknowledgment Number` in the example exercise previously presented. Packet-trace based exercises allow specifying dedicated feedback per packet field type. Here, the student is presented with an explanation of the `TCP Acknowledgment Number` field so that they can learn from their mistake and correct it. The INGInious feedback format is based on reStructuredText and can thus contains links, mathematical equations and images.

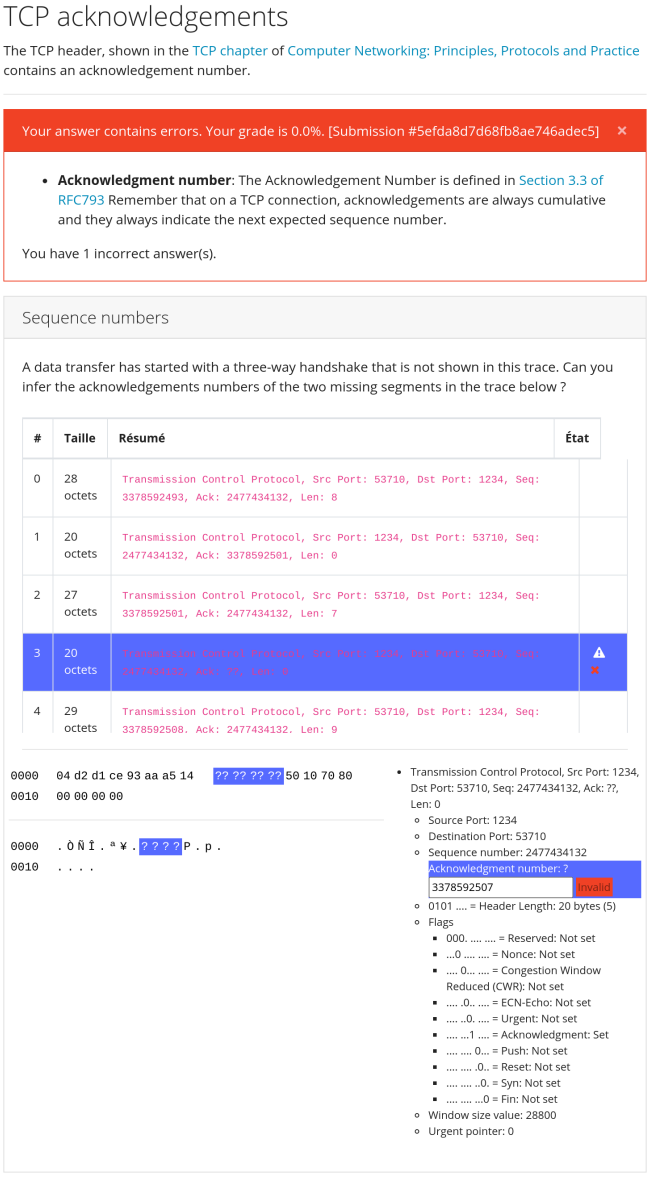

Figure 2: INGInious exercice with a TCP packet trace. The student incorrectly computed the TCP Acknowledgment Number of an ACK.

We integrated these packet traces in an online textbook [1] offering both theory and practical exercises in a single website [2]. This approach allows students to first read and understand the theory and principles of computer networking, and then seamlessly exercise their knowledge with small practical exercises presenting several concepts of network protocols as presented in Figure 1.

3. CONTRIBUTING

The open-source CNP3 course contains INGInious exercises [1] for most of the core Internet protocols, i.e. IPv6, TCP, DNS and HTTP. Those exercises could be expanded in several ways. First, new exercises for the existing protocol discussed in the course could be added. For instance, there is no exercise on TCP retransmissions for the moment. Second, new Internet protocols could be the subject of new exercises, such as the new transport protocol QUIC.

We encourage the community of computer networking educators to consider contributing to the CNP3 project by implementing new exercises and submitting them to the CNP3 exercises repository [1]. They will be reviewed and integrated as part of a quarterly release.

Documentation and tutorials exists on how run INGInious and exercises [6], as well as on how to run and to use our extension including the network dissector [7].

4. REFERENCES

[1] Olivier Bonaventure. CNP3 INGInious exercises. `https://github.com/cnp3/INGInious-packets`, Accessed Jun-26-2020.

[2] Olivier Bonaventure et al. Computer networking : Principles, protocols and practice, 2020. `https://beta.computer-networking.info/` - Accessed Jun-26-2020.

[3] Guillaume Derval, Anthony Gego, Pierre Reinbold, Benjamin Frantzen, and Peter Van Roy. Automatic grading of programming exercises in a MOOC using the INGInious platform. *EMOOCS'15*, pages 86–91, 2015.

[4] Kevin R Fall and W Richard Stevens. *TCP/IP illustrated, volume 1: The protocols*. addison-Wesley, 2011.

[5] Walter Goralski. *The illustrated network: how TCP/IP works in a modern network*. Morgan Kaufmann, 2017.

[6] Anthony Gégo and Guillaume Derval. INGInious documentation. `https://inginious.rtfd.io`, Accessed Jun-26-2020.

[7] Maxime Piraux. INGInious network trace problem. `https://github.com/cnp3/INGInious-problems-network-trace`, Accessed Jun-26-2020.