



Teaching of TCP Fundamental Operations Using a Digital Tool

Ilse Alicia López-Pedroza and Francisco de Asís López-Fuentes^(✉)

Department of Information Technology, Universidad Autónoma Metropolitana - Cuajimalpa,
Av. Vasco de Quiroga 4871, Cuajimalpa, 05348 México City, Mexico
ilsepedrozaa1@gmail.com, flopez@cua.uam.mx

Abstract. Today, digital tools play an important role in education. These tools are a support to strengthen the knowledge of students while for teachers it can become part of their work material. Using digital didactic tools, the students can understand an algorithm with a certain degree of difficulty. We present in this paper a digital tool to support the teaching and learning of transmission control protocol (TCP) fundamental operations. Using interactive examples and animations, our didactic tool provides complementary information that help to understand the TCP basic concepts.

Keywords: Didactic software · Networking · TCP protocol · Simulation

1 Introduction

During the year 2020 the world has faced a pandemic that kept us in social confinement, this includes the fact that schools have been closed for a long period. However, in this situation, alternatives were sought that could help distance learning. The main resource used to meet these needs have been digital tools and video conferences. The importance of technologies in education has been reflected, even increasing their price at the market and worldwide level. A greater benefit could be seen in education, some teachers realized that these technological resources could be used in face-to-face classes or for homework. The importance of didactic tools in teaching is reflected worldwide, it has already been accepted as a complement to the information given by teachers or as the means of teaching where teachers take the resources as the main source for their classes. A teaching tool helps support inside or outside the classroom, as it is easy to access and intuitive. However, the main limitation faced the digital tools based on internet is access to the internet since several localities still do not have the necessary internet infrastructure.

Networking courses are an important core of different bachelor curriculum related to information technology and computer engineering. However, motivating students to learn topics related to networking such as internet protocols can be often difficult and boring mainly due because its theoretical subjects. In this work, we present a digital tool to support the teaching/learning process of the fundamental operations of the Transmission

Control Protocol (TCP) [1]. Using this tool, we expect that students can acquire, reinforce, and exercise their knowledge about this important protocol. Interactive examples and animations are used to guide the user through this tool.

The rest of this work has the following organization. Section 2 presents information related to protocols and networking didactic tools. In Sect. 3 information about basic and fundamental aspects of TCP are described. In this section, we also explain the TCP basic operations to be implemented in our didactic tool. Section 4 presents the general design of our didactic tool, while its implementation is described in Sect. 5. Section 6 describes the tests and evaluations done to our tool. The article concludes in Sect. 7.

2 Related Work

Networking protocols are described in many textbooks [2–6], and its applications have been discussed extensively in the computer networking literature [7, 8]. In this section, we review some didactic tools related to our work presented in this paper. Authors in [9] present NEO as a web tool where communication protocols can be learned. This tool mainly shows detailed definitions about communication protocols. There are animation sections where are shown the characteristics and operations of each OSI/ISO and TCP layer reference models. Isiunne is proposed in [10] as a methodology for the development of teaching/learning tools using animations. This tool has simple animations, which are used to explain data communication concepts, the ISO/OSI reference model and data communication protocols. Its design has been planned for different user profiles as administrator, teacher, or student. Isiunne contains evaluations of the concepts shown, which are administered depending on the user profile with which it is entered. Kiva Network simulator (KivaNS) is proposed in [11] as a free and open source java-based application. KivaNS allows user to design data network schemes and to simulate the IP routing through these networks. Thus, we can find data network diagrams and packet routing simulations. KivaNS is mainly oriented to simulate the IP behavior, and emulates the basic operation of technologies in the link layer as Ethernet. The main objective of this tool is to help to design and understand the data networks operations, specifically the packets routing in the TCP/IP architecture, without needing a real infrastructure and traffic analysis tools. Other digital tools related to teaching of computer networks can be found in the literature, however most of these tools still have many characteristics to be covered and improved.

3 TCP Background

The Transport Control Protocol (TCP) is a protocol in the transport layer, and its main task is the reliable transportation of data through the network [2]. This protocol allows to exchange information between computers and application programs. TCP ensures that the transferred data arrive correctly, secure and in order. Main attributes of TCP are [2]:

- a fully duplex bidirectional virtual circuit.
- data is transmitted as a data stream.
- its reliable data transmission is based on:

- sequence numbers
- checksum
- acknowledgements
- retransmission when an acknowledgement is received timeout.
- greater efficiency based the sliding-window principle.
- urgent data and push function
- graceful connection shutdown
- transport-user addressing using port number.

The TCP protocol header has different fields with the following significance [2], [5]: *source and destination port number*, are fields of 16-bits and denote the initial and end points of a virtual circuit. The *sequence number* is a 32-bits word which refers to the send direction. The *acknowledge number* also is a 32-bits word and it applies to the number of bytes received by the other end [2]. The *data offset* contains the length of the TCP header. The *flags* are bits used to trigger actions in TCP. There are six flag bits in the TCP header, and one or more of them can be activated at the same time [2]:

1. URG - pointer in *Urgent* field is valid.
2. ACK - acknowledgment number is valid.
3. PSH - the receiver should pass this data to the application as soon as possible.
4. RST - resetting of the connection.
5. SYN - synchronize sequence numbers to initiate a connection.
6. FIN – the sender shutdowns the connection, and the data flow is finished.

The *window* contains the number of bytes that a receiver can accept in its data buffer for this connection. This field is used to control the data flow. The *checksum* includes a code which is used in the receiver to detect some error in transmission. The *urgent pointer* points to a data byte called urgent data which must be read immediately. The *options* field is used to add extra facilities not covered by the regular header.

The TCP protocol contains many functions which ensure that the data arrives correctly. For example, TCP establish a successful and secure connection to ensure that information is not lost. The main functions to be considered in our didactic tool are TCP connection establishment (normal case), retransmission and sliding window. In the following we give a briefly description of these functions.

3.1 TCP Connection Establishment

A TCP connection is established through an agreement of three. One side waits for a connection while the other side executes the connection. For this connection, protocol needs to specify the address, the port number, the TCP maximum segment size to be accepted, and optionally some user data. After this, transmitter waits for the response. The segment arrives at the receiver and it checks if there is any process in the list. Depending on what is on the list, the connection is rejected or accepted. If the connection is accepted, then a confirmation segment is returned, otherwise a reject response is sent to the transmitter. A SYN segment consumes 1 byte of sequence space [5]. Figure 1 shows how a TCP connection is established.

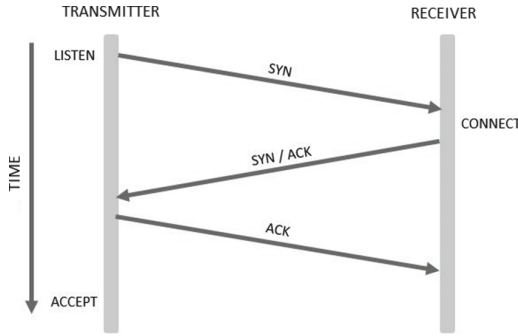


Fig. 1. Establishing a TCP connection.

3.2 Retransmission

During the information transmission may be loss of packets. Therefore, the retransmission is the way to ensure that the packets arrive correctly to its destination. TCP returns an ACK each time it receives data, on that path a timer is started. If the timer expires before the ACK arrives then the packet sending process is performed again. We can have n number of retransmissions, but all depends on the amount of lost packet that exist during the transmission. We can see this in Fig. 2.

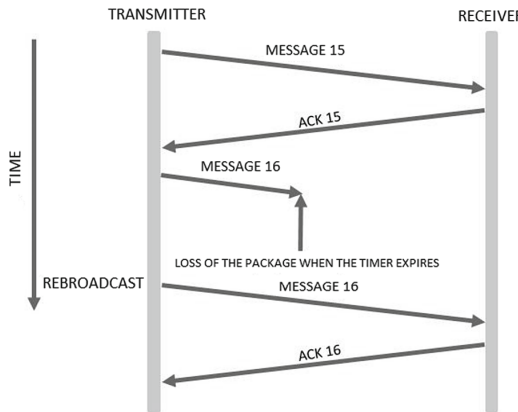


Fig. 2. An example of retransmission.

3.3 Sliding Window

TCP uses a sliding window scheme for data flow control in packet communications. Transmitter may send the number of bytes specified in the windows, without having to wait for an acknowledgement from the other part [2]. The sliding window can be enlarged or reduced depending on the needs. The aim is to carry the complete information from

one side to the other. Figure 3 shows an example of how this function in the TCP protocol is done.

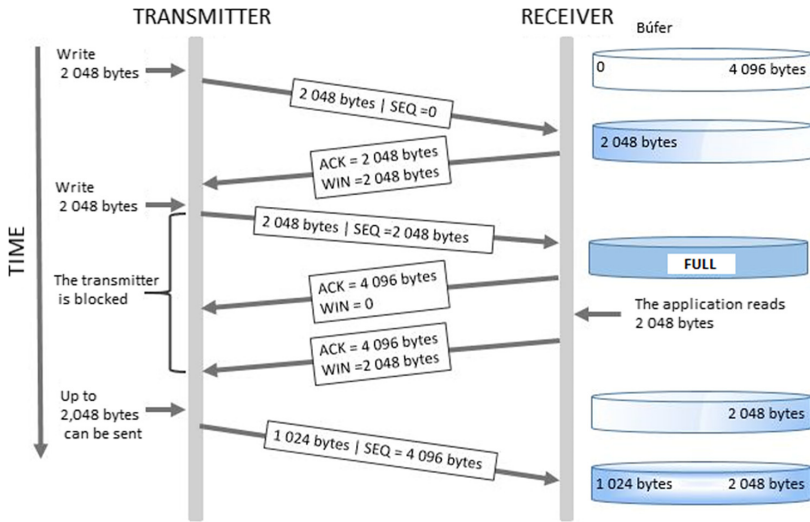


Fig. 3. An example of sliding window (adapted from [5]).

4 Analysis and Design

Designing an interface simple and easy-to-use both students and teachers is one of the objectives of this work. When a web page is created, we need to develop a system which should be easy to access and use by the main actors. In our case main actors are the students and teachers. This section describes the different design stages of our didactic tool.

4.1 System Structure

To develop our system, we have structured the TCP information into a main page, where each TCP function is represented. When the users enter the system, they can access all the information or only information that they could consider of interest. This is done through a main menu where the users can visualize this information.

4.2 Main Page

The main page shows the TCP definition and general information about this protocol. There is also a glossary with some technical terms that are used throughout the system. The main page of the system has a menu with three different options: TCP connection establishment, retransmission, and sliding window.

4.3 TCP Connection Establishment, Retransmission, and Sliding Window Sections

Any section in the main menu (connection establishment, retransmission or sliding window) has two options: definition and example. When the definition option is selected, we find a brief definition about the selected function and an animation. The animation shows a concrete example of how the function is performed between two devices and how the flow is carried out by this TCP function. The example option contains an interactive animation which represents a realistic case of how selected TCP function works (e.g.: connection establishment).

From these sections users can navigate through all the information in the system, they can learn about TCP connection establishment, retransmission or TCP sliding window. Even the users can return to the definitions or the examples.

In general, our design is proposed to use more images and animations avoiding use a lot of text. The animations invite to the user putting more attention in a specific TCP function. Thereby, our tool tries to capture the interest of students during the teaching/learning process. Likewise, teachers have a possibility to reinforce the knowledge given by them in the classroom. An approach based on animations can also be used to give repeated explanations about a topic or as a self-taught tool.

5 Implementation

Our didactic tool has been mainly developed using HTML and CSS language. Power Point and Scratch were used for the animations and examples. We use HTML to develop the system structure, the main menu was developed to display a submenu which shows the different options (definition and example) for each TCP function. We use CSS to define the colors and the fonts. Figure 4 shows the main page of the system where we can see at the top of menu the TCP functions (Our system has been written in Spanish).

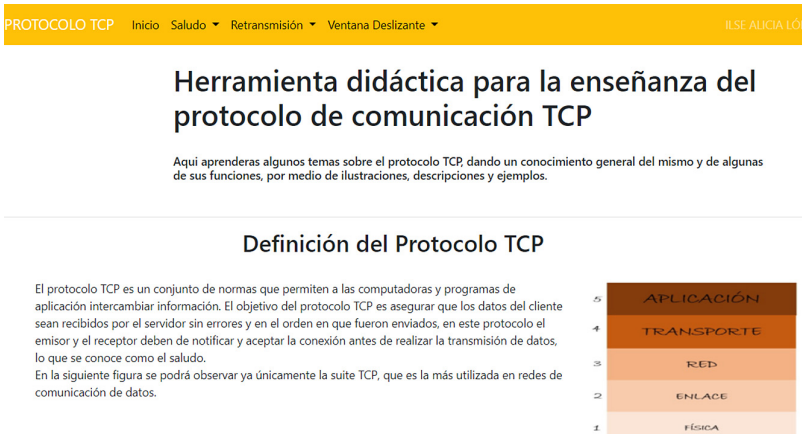


Fig. 4. System main page.

For a better understanding of the TCP, our system uses simple animations instead of images, since these animations explain step by step the processes of the TCP functions. First, the animations were made in Power Point, later they were integrated into the system using the iSpring tool. Using Power Point, we can draw the figures, add movements and more visual effects that help to understand the subject. We can obtain simple, understandable, and easy-to-use animations which can be viewed, stopped, restarted, or advanced. In this way, the user can see our tool as a more interactive method instead of an images-only collection. The Fig. 5 shows this animation development process and its final display in the system.

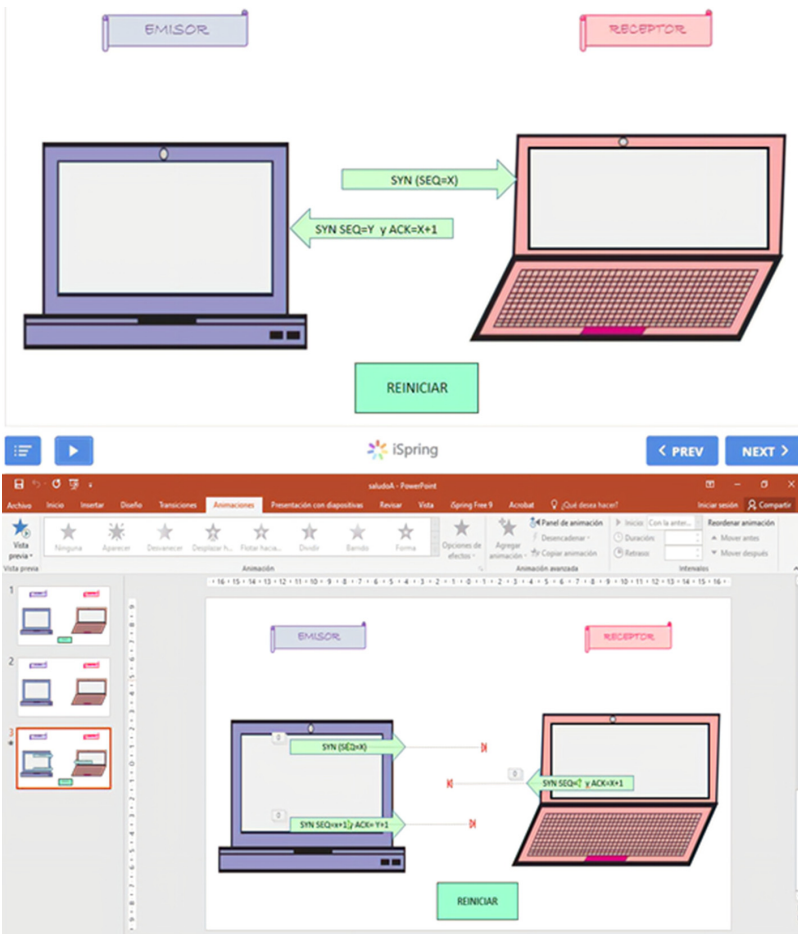


Fig. 5. Animation development for the connection establishment function.

The examples shown in the system are interactive, in the form of a game. So, the user can enter different data and simulate the moment when a failure in the function occurs or whether the function works correctly. Scratch [12] was used to develop the

interactive examples. Scratch is a visual programming language, and it is an alternative to develop the examples in such a way that the general expectations of the system are met. The examples were programmed within the Scratch home page, so these examples can be accessed from this home page. Later the examples were added to the system. The examples can be reproduced as many times as the user wishes. Each time the user clicks on the green flag, the example will be started from the beginning, requesting the login data again. Figure 6 shows an example of how an interactive example for the retransmission function is displayed in the system.

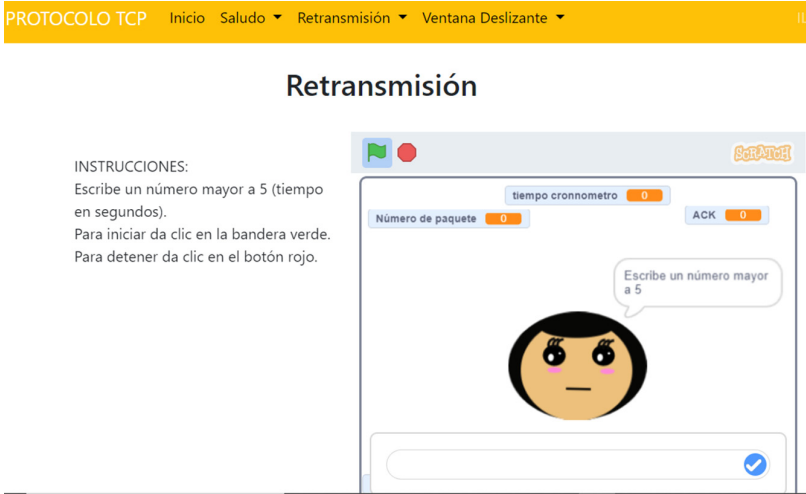


Fig. 6. Example of the retransmission function.

The programming of the animations was developed specifically according to the functions of the protocol. Figure 7 shows a general structure example about how the animations are displayed in Scratch. This application provides instructions, commands, variables, actions, functions, and other options. This instructions series are dragged to the middle, where the logic of programming the animations is carried out (actions in the right part of the Fig. 7). In this example we can see a programming block for the connection establishment function, which includes conditionals, and we can see that more than one block has already been added for a single drawing within the animation.

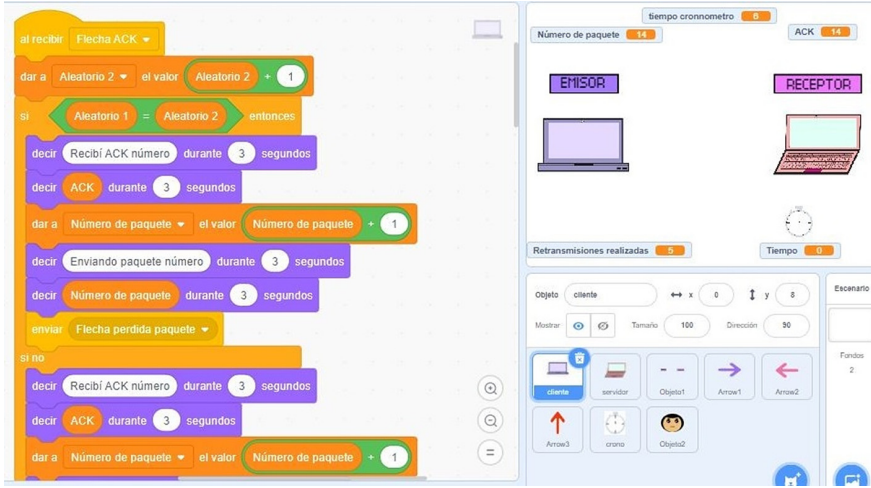


Fig. 7. Scratch development for the retransmission function.

6 Evaluation

Our didactic tool was tested with some users to evaluate its performance. The users have previous knowledge of TCP, some them are students and other graduates. Therefore, not all have recent knowledge over this protocol.

Users had access to the system, later they were given to fill out a form where the main questions were to highlight the design of the system and how this information was useful to them. The responses were very homogeneous since most of them had the same observations and likes for the system. The profile of the users who were tested was users with previous knowledge of TCP. The tests applied to users yielded results that helped to assess the quality of the system as well as see the advantages and improvements that can be made to it in the future work.

In general, the system was liked by the users. However, the design part received the most observations. The users made several observations related to the design and placing a greater emphasis on font size and system colors. Regarding the evaluation of the system, we observe that users showed a greater acceptance for the animations and interactive examples than the text. Some users did not use the text, because they were guided by the examples and animations. In fact, some users did not use the text, because they were guided by the examples and animations. Some results of our evaluation are shown in the Fig. 8.

¿Qué información te fue más útil?

8 respuestas

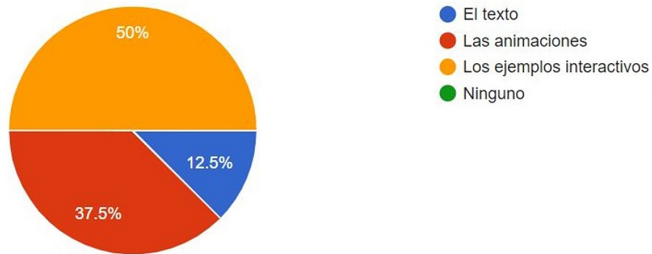


Fig. 8. User responses to the quality tests of our didactic tool.

7 Conclusions

The current world situation has shown us the importance of information technology, especially in education. In this paper, we present a didactic tool for teaching TCP protocol, which shows how information technology can play an important role in the teaching and learning process. The TCP protocol, like other subjects, can be taught through computational means such as cellular phones or electronic tablet. In our proposed didactic tool, we have learned that visual aspects are very important. The inclusion of little texts helps the users to have more taste and ease for learning of a subject. Our tool does not try to substitute to the teacher, but it tries to complement the information about TCP. Likewise, our tool could help to the students to clarify doubts related to TCP. Our didactic tool is easily accessible, it can be consulted within classrooms, at home or anywhere with internet access. A limitation of our tool is its dependence of internet because it cannot be used in location where the internet access is not available.

As future work, our didactic tool can be extended in different aspects. For example, we can add more animations, more extensive examples, new evaluation questionnaires, or aspect related to usability can be considered. Also, an open access version for teachers or students could improve this tool with feedback or collaboration. Finally, others TCP functions can be added such as round-trip time during the retransmission or a comparison with UDP communication protocol.

References

1. Postel, J.: Transmission Control Protocol. RFC 793 (1981)
2. Santifaller, M.: TCP/IP and NFS Internetworking in UNIX environment, Addison-Wesley (1981)
3. Peterson, L.L., Davie, B.S.: Computer Networks: A Systems Approach. 5th edition, Morgan Kaufmann (2011)
4. Kurose, J., Ross, K.: Computer Networking: A Top-Down Approach. 7th edition, Pearson (2016)
5. Tanenbaum, A., Wetherall, D.J.: Computer Networks. 5th edition, Pearson (2011)
6. Comer, D.E.: Computer Networks and Internet. 1st edition, Prentice-Hall (1997)

7. Stalling, W.: Data and computer communications. 8th edition, Pearson Prentice Hall (2007)
8. Comer, D.E., Stevens D.L.: Internetworking TCP/IP Vol. 2, 3th edition, Prentice-Hall (1995)
9. NEO: <http://neo.lcc.uma.es/evirtual/cdd/tutorial/Indice.html>. Accessed 08 Dec 2019
10. Isiunne: <http://www.lsiunne.com.ar/>. Accessed 10 Dec 2019
11. KivaNS: <http://www.disclab.ua.es/kiva/>. Accessed 10 Dec 2019
12. Scratch: <https://scratch.mit.edu/>. Accessed 22 Jul 2020